

# Pro

Mensuel - Mai 2007 - N°97

www.programmez.com

# grammmez!

LE MAGAZINE DU DÉVELOPPEMENT

Créer un composant **Web 2.0** Ma première application

**Google**™  
Web ToolKit

avec  
**Adobe**  
**Apollo**



## Générez automatiquement votre

# C O D E

Les différentes techniques, les outils. Faut-il adopter le MDA ?  
Etes-vous "Code" ou "Modèle" ?

## SOA ? Facile !

- ✓ Construire une Architecture Orientée Services
- ✓ SOA avec .NET 3.0
- ✓ Quelle sécurité ?

## LANGAGE

Découvrez  
le langage D

### JAVA

Générer des documents PDF

### LINUX

- Surveiller vos fichiers avec INotify
- Développer avec Mono
- Mécanisme fork()

# DÉVELOPPEZ 10 FOIS PLUS VITE

Ne gaspillez plus votre temps ni vos budgets. Vous aussi, développez et faites développer vos applications avec WINDEV, le logiciel de développement professionnel qui permet de développer 10 fois plus vite.

N°1 EN FRANCE

IDE - ALM - toutes les bases de données - Merise - UML - RAD - RAD Java - langage de 5<sup>e</sup> génération - Windows - .Net - API - DLL - générateur d'IHM - tests de non régression - gestionnaire de sources - installateur automatique - WebServices - XML - SOA - 3-tiers - composants - règles métier - Lotus Notes natif - PDF - codes barres - support technique gratuit - déploiement gratuit - autoformation livrée - etc...



Windows 98, NT, 2000, Vista, XP...  
Vos applications sont performantes,  
avec un «look» moderne et efficace.

WINDEV XI permet de développer jusqu'à 10 fois plus vite tous les types d'applications, de l'application départementale à l'ERP diffusé mondialement (à découvrir dans les «100 témoignages», gratuit).

Le développeur bénéficie d'un environnement intégré totalement en français, qui gère l'intégralité du cycle de vie des applications, avec les fonctionnalités les plus sophistiquées disponibles en standard. WINDEV 11 est l'outil de développement le plus complet au monde.

Les utilisateurs bénéficient eux de fonctionnalités automatiques dans leurs applications, comme l'export vers Excel, OpenOffice, PDF, ou un générateur d'états, de requêtes et de codes barres libre et gratuit.



Vous aussi découvrez WINDEV XI, et développez 10 fois plus vite...  
Demandez le dossier + le DVD gratuit.



444  
Nouveautés



Demandez le dossier technique gratuit (124 pages en couleurs, en français) accompagné de 100 pages de témoignages et d'un DVD. Version Express Gratuite

Tél Province 04.67.032.032 Tél Paris 01.48.01.48.88 [www.pcsoft.fr](http://www.pcsoft.fr)

Fournisseur Officiel de la  
Préparation Olympique



## SOMMAIRE

N°97

## &gt; Actus

L'actualité en bref	6
Agenda	8

## &gt; Evénements

Sun : la machine virtuelle Java devient multi langage !	10
---	----

## &gt; Test

Adobe Creative Suite 3 Web Premium	12
------------------------------------	----

## &gt; Projets

Les meilleures pratiques du développement des logiciels libres	16
--	----

## &gt; SGBD

Synchronisation avec ADO.net	20
------------------------------	----

## &gt; Gros Plan

Le langage D : votre prochain langage	24
---------------------------------------	----

## &gt; Dossier : Génération de code

Générer automatiquement votre code ?	28
Génération automatique d'applications selon les concepts MDA	30
La génération de code au sein de l'usine logicielle CodeFluent	33
Le code ou le modèle ?	36
Panorama des outils de génération de code	40

## &gt; Dossier : SOA facile

SOA : de la complexité à la facilité	42
La SOA facile selon Microsoft	46
La SOA pragmatique	50
Panorama des outils orientés SOA	54

## &gt; Technique

Le débogage avec Visual Studio	55
--------------------------------	----

## &gt; Développement Web

Ma première application Apollo	56
Construire un composant " glisser - déposer " avec Google Web Toolkit (1re Partie)	58

## &gt; Code

Mécanisme du fork() avec l'EDI Code::Blocks	62
Premiers pas avec Mono sous Linux	67
Mes débuts en WPF 3e partie : la 3D	69
Générez vos documents PDF et RTF avec Java	72
Surveillez vos fichiers sous Linux avec INotify	76

## &gt; Temps libre

Ludique	80
Les livres du mois	82



## CD-Rom 97 PROGRAMMEZ !

**Adobe RoboHelp 6**

RoboHelp 6 permet de créer, gérer et publier des contenus destinés à des systèmes d'aide et à des bases de connaissances. Limité 30 jours

**Sun Web Developer Pack (Windows et Java)**

Package spécialement étudié pour utiliser Ajax et les flux RSS dans Java

**Glassfish v2 bêta 1 (Windows et Linux)**

Dernière version du serveur Java EE 5.0

**D Compiler**

Compilateur pour le langage D

**Mono 1.2.3 (Windows)**

Implémentation open source de .NET

**Iona Celtix 1.1 Enterprise (Windows)**

Solution ouverte complète, comprenant toutes les couches nécessaires à la création d'une SOA

**Croquet 1.0 SDK**

Plate-forme open source collaborative 3D basée sur la machine virtuelle Squeak/Smalltalk

**Objecteering 6.0 SP1 Free Edition**

Version gratuite illimitée en modélisation UML2. Génération automatique de code Java ou C++ limitée à 50 éléments de modèle.

(La version complète **Enterprise Edition** est téléchargeable sur [www.objecteering.com](http://www.objecteering.com), limitée à 30 jours).



**VOUS VOUS EFFORCEZ D'ADAPTER EN PERMANENCE VOS APPLICATIONS AUX ÉVOLUTIONS DES SYSTÈMES D'EXPLOITATION. VOS APPLICATIONS FONCTIONNENT, MAIS QU'EN EST-IL DE VOTRE PRODUCTIVITÉ?**

Les évolutions constantes des OS obligent votre équipe à s'adapter toujours plus vite aux évolutions des plateformes telles que Windows® Vista™, Mac® OS X ou Linux®.

Avec Qt® – l'environnement standard pour le développement d'applications performantes et multiplateformes – suivre le rythme des innovations des systèmes d'exploitation devient facile. Depuis plus de douze ans, Trolltech s'emploie à rechercher et incorporer dans l'API unique et multiplateforme de Qt les technologies spécifiques à chaque plateforme, préservant ainsi nos clients tels que Adobe®, Google® ou Skype™ des changements constants sur chaque plateforme cible. Qt prend en charge l'adaptation de ces évolutions et vous permet ainsi de vous concentrer sur l'aspect production pure de vos applications.

**NOUS VOUS AIDONS À ALLER DE L'AVANT.**

Pour en savoir plus, téléchargez dès aujourd'hui une version de test de Qt, gratuite, valable un mois:  
[www.trolltech.com/futureproof](http://www.trolltech.com/futureproof)

**Qt BY TROLLTECH®**  
**CODE LESS. CREATE MORE.**



# NOUVEAU : l'emploi

# Édito



## Et si on parlait de Web 2.5 ?

Depuis quelques mois, un couple est en train de se former. On devrait plutôt parler d'un couple de frères ennemis. Mon premier est un mastodonte connu pour ses fenêtres et le second est un terme de construction. Et oui vous avez reconnu Microsoft et Adobe.

Ils se poursuivent à coup d'annonces, de présentations. L'un sort sa plate-forme-forme Internet, client riche, pas de souci, l'autre suit. L'un veut imposer un nouveau player vidéo, no problem, l'autre réplique. L'enjeu ? Définir les nouveaux paradigmes pour Internet, le desktop et le développeur.

Cette interrogation permet aujourd'hui de dresser un premier bilan. Si Ajax constitue toujours la colonne vertébrale des applications web dites Web 2.0, de facto, nous passons, sans le savoir, au Web 2.5. On revient à un modèle applicatif connecté, déconnecté, capable de fonctionner sur un navigateur mais aussi, et surtout, à partir d'un runtime desktop. WPF et Apollo (version finale prévue fin 2007) ne proposent pas autre chose. On nous réinvente le client riche, mi-jour. WPF et Apollo autorisent des processus fonctionnelles que l'on ne connaissait pas notamment sur l'interface et surtout, capables de fonctionner en mode connecté et déconnecté. À côté de ces nouveaux clients riches, ces technologies se déclinent naturellement sur le navigateur, d'un côté WPF/E (prévu pour la rentrée 2007), euh pardon, Silverlight, de l'autre, Flash - Flex. Et le mobile redevient une cible centrale. Le but n'est plus aujourd'hui l'application, mais comment afficher et utiliser le contenu. Dans notre Web 2.5, on ne dissoocie plus le contenu de l'application et de son interface.

L'autre tendance lourde est de lier le designer et le développeur. On veut les faire travailler ensemble. On doit pouvoir collaborer, échanger, communiquer, partager les idées, les fichiers dans un environnement unique. Microsoft le propose avec Expression (bêtas prévues cette été) et Adobe mise sur l'intégration entre ses outils. Là se pose un dilemme. Quel environnement choisir ?

Car, une fois de plus, nous sommes confrontés à un choix cornélien. Un environnement neuf, puissant, séduisant avec Expression mais dont les logiciels ne fonctionnent que sur Windows et les langages se limitent à ceux de la plate-forme. De l'autre, on joue la multi-plate-forme, le multi-langage avec la Creative Suite 3 et le futur Apollo. C'est là une question de philosophie. Les deux ont des avantages et des inconvénients mais les deux éditeurs jouent une partie de leur avenir dans cette guerre. Celui qui réussira à s'imposer, aura une prédominance. Ce n'est pas un hasard si Adobe a décidé d'offrir le moteur d'exécution d'ActionScript à la fondation Mozilla. Histoire d'étendre sa présence. Pour Microsoft, le pari est de séduire aussi bien le développeur web que le designer. Et si notre développeur devenait prescripteur auprès du graphiste ?

Car n'oublions pas que sans applications, donc sans développeurs, un langage, un outil, meurt. Il faut donc des *killers app* capables d'époustouffer l'utilisateur, le développeur. Elles restent à inventer.

■ François Tonic

## PROCHAIN NUMERO

N°98 - Juin - Parution : 31 mai 2007

## Dossier SGBD

Choisir sa base de données.  
Quel SGBD, pour quelle application ?  
Les nouvelles fonctions.

## XUL

Développez avec un langage surpuissant facilitant la création d'interface graphique dynamique et interactive.



- Quand les utilisateurs disent que le Directeur Informatique a du "charisme" je me demande s'il ne s'agit pas d'une formule polie pour dire que je suis une "grande gueule"

Souriez avec Jissey  
[www.programmez.com](http://www.programmez.com)  
chaque semaine,  
un gif animé

Jissey

## Programmez!

LE MAGAZINE DU DEVELOPPEMENT

Rédaction : [redaction@programmez.com](mailto:redaction@programmez.com)

Directeur de la Rédaction : Jean Kaminsky

Rédacteur en Chef : François Tonic

Ont collaboré : J.M. Maman, F. Mazué, X. Vanneste, C. Durand, S. Marc, C. Robert, N. Muderwa, J. Neveux, D. Cohen-Zardi, S. Mourier, T. Matusiak, J.P. Schoch, D. Tizon, P. Grosjean, M. Chaize, H. Darmet, X. Leclercq, L. Guillois.

Dessins : Michel Prédoux

Maquette : AJE Conseils

Publicité : Pour la publicité uniquement :

Tél. : 01 41 77 16 03

[publicite@programmez.com](mailto:publicite@programmez.com)

Editeur : Go-02 sarl, 6 rue Bezout - 75014 Paris  
Coordination@programmez.com - Dépôt légal : à parution - Commission paritaire : 0707K78366 - ISSN : 1627-0908 - Imprimeur : ETC - 76198 Yvetot

Directeur de la publication : Jean-Claude Vaudecrane  
Le numéro comporte un CD Rom et un catalogue SOS Developper

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 : 45 € - Etudiant : 39 € - CEE et Suisse : 51,83 € Algérie, Maroc, Tunisie : 55,95 € Canada : 64,33 € Tom : 79,61 € - Dom : 62,84 € Autres pays : nous consulter.  
PDF : 35 € (Monde Entier) souscription en ligne.

## SÉCURITÉ

## Aladdin sécurise vos déploiements

Comment protéger ses applications ou proposer des formules d'abonnement ou de location ? Aladdin propose sa nouvelle clé : HASP HL NetTime. Elle est conçue pour fonctionner en réseau. " Conçu spécialement pour les besoins particuliers des logiciels multi-utilisateurs, HASP NetTime répond aux nombreuses demandes de fournisseurs de logiciels dont les clients utilisent des logiciels dotés de



licences basées sur le temps en environnements réseau ", a déclaré M.

Avi Barir, vice président de la division Software DRM de Aladdin Knowledge Systems. La clé supporte les modèles suivants :

- Licence basée sur le temps : utiliser l'horloge en temps réel de HASP HL NetTime pour accorder des abonnements ou louer les logiciels pendant une durée donnée.
- Usage simultané sur réseau : permettre à 250 utilisateurs maximum d'accéder simultanément à une application en environnement réseau.
- Basée sur les fonctions : activer ou désactiver des fonctionnalités du produit selon les besoins et le budget de l'utilisateur. La mémoire accrue de HASP HL NetTime permet de protéger davantage de fonctions et/ou d'applications avec chaque clé.
- Licences personnalisées : utiliser le nombre de postes réseau, la mémoire sécurisée et l'identifiant unique de HASP HL NetTime pour créer des modèles de licence parfaitement adaptés aux besoins de l'entreprise.

## PUCE

## WebSphere mise sur RFID

IBM implémente les puces sans contact RFID dans WebSphere Premium Server 6. Cette nouvelle version permet d'agréger et d'analyser les informations issues des composants RFID, notamment dans le cadre d'une SOA ou d'une application. Avec le développement des puces RFID, il devient important de pouvoir utiliser cette technologie et lire les informations. L'outil utilise un modèle Eclipse pour les terminaux RFID. Les principaux constructeurs seront compatibles avec WebSphere.

## LANGAGE

## Et si HTML n'était pas mort ?

" Le langage HTML a commencé simplement, avec un balisage structuré, aucun impératif d'octroi de licence et la capacité d'être associé à tout. Plus que tout, c'est à ses qualités de simplicité et d'ouverture qu'il doit son omniprésence et son énorme succès " explique Tim Berners-Lee, directeur du W3C et inventeur du langage HTML. " Il est temps de revisiter ce standard pour l'adapter aux besoins actuels des différentes communautés, de manière visible et ouverte, et pour ce faire, nous comptons sur la collaboration des fabricants de navigateurs ". HTML n'est peut être pas mort comme on aurait pu le penser. Alors que le W3C et de nombreux éditeurs misaient sur XHTML, depuis quelques semaines, le W3C relance l'activité HTML. Mozilla ou encore Apple veulent une nouvelle version de HTML. À voir si cette annonce aboutira à quelque chose de concret.

## WEB

## Intorel compile toujours plus

L'éditeur Intorel vient de dévoiler ActiveSite Compiler 5. Il est compatible avec Vista, .Net 2 et OpenSSL / HTTPS. Il prend en charge de nombreux langages pour concevoir des applications connectées et déconnectées, que ce soit pour une clé USB, Internet, etc. " ActiveSite 5.0 s'appuie sur ce qui a fait le succès des versions précédentes de la plate-forme pour offrir une suite compilateur de tout premier plan, parfaitement adaptée aux besoins individuels des professionnels du développement Web ", comme l'explique Dragana Lazic-Mijanovic, Responsable Ventes et Marketing chez Intorel. " La toute dernière version de cette technologie permet de compiler tout type d'application Web, personnelle, professionnelle, ASP, ASP.NET ou PHP, dans une application autonome qui peut être utilisée sur n'importe quelle plateforme PC, y compris désormais Vista. Cette fonctionnalité peut servir pour toute une gamme d'opérations différentes de développement Web, depuis la compilation d'applications d'entreprise, de sites Internet internes ou d'applications Web, jusqu'au développement de matériaux de promotion commerciale pour la distribution hors ligne ".

Site : [www.intorel.com](http://www.intorel.com)

## SUPERVISION

## Surveiller ses systèmes et applications

NetIQ vient de lancer AppManager v7. Il s'agit de pouvoir superviser l'ensemble de ses applications et systèmes sous Linux, Windows et Unix. " Avec AppManager v7, nous lançons la solution de supervision des applications et des systèmes la plus complète de l'industrie pour permettre aux entreprises de contrôler leurs environnements existants tout en se préparant à adopter des technologies de prochaine génération ", dit Chris Pick, Vice-Président Produits et marketing de NetIQ. " Les responsables opérationnels ne peuvent pas s'appuyer sur des technologies obsolètes pour faire face à leurs challenges métier de demain. " Cette version propose des agents distants pour supprimer les installations manuelles, une détection automatique des plates-formes et des applications en environnements virtuels, hétérogènes et dynamiques. Les tarifs démarrent à 500 euros par serveur windows.

## WEB

## WFP/E devient Silverlight

Microsoft a annoncé mi-avril le nom définitif de WFP / E : Silverlight. Pour mémoire, il s'agit de pouvoir faire fonctionner des applications riches de type WPF sur différents navigateurs sur MacOS X et Windows. Une version mobile sera disponible plus tard dans l'année. La première version finale pour desktop sera disponible courant de l'été. Il sera possible, par exemple, de diffuser via Silverlight des vidéos au format Windows Media (incluant une implémentation de VC-1). On peut aussi mettre en œuvre un système de DRM (basé sur Microsoft PlayReady). À noter que Microsoft propose aussi un kit de développement pour utiliser l'encodage VC-1.



Microsoft  
Silverlight



## Pour vos projets de développement, choisissez la qualité

Faites l'acquisition du MissionKit d'Altova, et découvrez  
comment économiser sur certains des logiciels les plus prisés.

### Contenu du MissionKit Altova 2007:

- Les outils de développement XML de référence dans le monde entier (XMLSpy, MapForce, StyleVision, etc.)
- Design d'applications, gestion de données et options de modélisation pour les créateurs de logiciels

Le MissionKit 2007 d'Altova regroupe des outils intelligents de développement d'applications, de gestion de données et de modélisation à 50% de leur prix habituel. Disponible dans différentes configurations adaptées aux besoins des créateurs de logiciels et des développeurs XML, le MissionKit d'Altova vous offre les fonctionnalités les plus performantes et le meilleur rapport qualité-prix. C'est un véritable billet première classe pour atteindre la puissance, la rapidité et la simplicité d'utilisation de cette ligne de produits Altova maintes fois primée. Faites un tir groupé pour économiser!

Téléchargez le MissionKit 2007 d'Altova dès aujourd'hui: [www.altova.com](http://www.altova.com)



## MOBILITÉ

## RIM étend la programmation



Pour attirer les utilisateurs, il faut des applications. RIM, constructeur du BlackBerry, propose de nouvelles API utilisables dans le BlackBerry Java Development Environment. Aujourd'hui, plus de 100 000 développeurs utilisent cet outil. Parmi les nouvelles API, on retrouve : le mobile multimedia API (JSR 135), Camera API,

un support étendu des formats audio, la disponibilité des web services et du XML, les API pour le GPS ou encore de nouvelles fonctions pour la géolocalisation et la 3D. Cela devrait faciliter la création d'applications riches. Site : <http://www.blackberry.com>

## RISQUES

## Compuware aime le risque

L'éditeur vous aide à gérer les risques opérationnels avec une solution d'audit applicatif : Hiperstation. Cela permet de mesurer l'impact des failles de sécurité interne et de les réduire. " Nos recherches ont démontré que les professionnels de la sécurité et de la confidentialité des données rencontraient de grandes difficultés dans ce domaine. Mais le défi le plus difficile à relever reste la menace interne. Pour bien détecter, freiner et gérer ces menaces au sein d'une entreprise, il est indispensable d'utiliser des solutions performantes telles que la solution de Compuware, " affirme Larry Ponemon, Président et Fondateur de l'Institut Ponemon. Hiperstation propose la capture des transactions, l'analyse des failles de sécurité, le traitement des données.

Site : [www.compuware.com](http://www.compuware.com)

## SOCIÉTÉ

## OpenWide se diversifie

OpenWide, intégration open source, poursuit son développement dans les solutions industrielles. Pour ce faire, une nouvelle entité est créée : OS4i (open source for industries). Il s'agira de proposer des solutions packagées. Des formations seront aussi disponibles. Un portail sera ouvert courant septembre sur l'open source dans l'industrie. " Le marché de Linux embarqué dans les produits industriels connaît une très forte croissance. Les enjeux de nos clients sont doubles. D'une part garder la maîtrise de leur environnement technique et le bénéfice de leurs applications existantes, de l'autre, respecter un " time to market " toujours plus court imposé par les services commerciaux et marketing. Avec l'internationalisation des marchés ils attendent également de nous une réactivité forte pour leur différentes implantations à l'étranger" commente François Hilaire (directeur d'OS4i).

## PLATE-FORME

## OSLO gère l'imprévu

L'environnement OSLO d'OSLO Software se propose de prendre en charge la gestion des événements. L'outil s'appuie sur la modélisation

et l'exécution par objectifs. Il utilise le multi-agent permettant une gestion fine des événements. Les agents sont des composants logiciels distribués capables de communiquer entre eux et pouvant résoudre des problèmes. Chaque agent possède ses propriétés. OSLO doit faciliter la modélisation métier et l'alignement des applications sur le métier. Le but est d'être plus réactif en permettant des changements rapides.

Site : [www.oslo-software.com](http://www.oslo-software.com)

## IDE

## Oracle met à jour Application Express

L'éditeur propose la 3e version majeure de son outil de développement Application Express. Il permet de développer des applications web en utilisant un simple navigateur ! Cette version doit améliorer le support et la conception de services web. Il supporte Flash Chart, PDF, Access Migration, le glisser-déposer... Il s'intègre avec la base Oracle 9 et 10. Rien de tel pour développer rapidement des applications utilisant une base Oracle ! Sur l'amélioration des web services, il s'agit de mieux prendre en compte les services venant de JDeveloper ou d'utiliser des processus BPEL.

## AJAX

## TIBCO aime toujours Ajax

L'éditeur TIBCO vient d'annoncer une suite logicielle open source pour développer et déployer des applications d'entreprise Ajax ! Il s'agit de General Interface Test Automation Kit. La suite comprend des fonctions de tests. Tibco avait déjà poussé en avant son outil de développement Ajax : General Interface. Il s'agit pour Tibco de rapprocher les applications Ajax de la SOA. La suite fonctionne directement dans un navigateur pour les applications conçues avec General Interface. L'éditeur met en avant les capacités de tests et de qualité pour Ajax, qui demeure difficile à déboguer. On pourra ainsi automatiser les tests !

Site : <http://developer.tibco.com>.

## Agenda

## MAI

## XP Day France 2007

Paris, les 2 et 3 mai 2007

## La conférence sur les méthodes agiles !

<http://www.xpday.fr/>

## Du 15 Mai 2007 au 18 Mai 2007

Paris

Novotel Paris Tour Eiffel

## XTech 2007

## Le Web Omniprésent

<http://2007.xtech.org/public/news>

## 22 mai, INTEL Paris FSO 92196

Meudon, **Présentation des solutions logicielles Rogue Wave eblattes@roguewave.com**

## Du 23 Mai 2007 au 25 Mai 2007

Centre international d'Affaires et de congrès de Paris (CAP 15) **Eurosec 2007**

<http://www.devoteam.fr/eurosec/2007/home.php?lang=fr>

Porte Maillot - Paris Palais des

Congrès / **jeudi 24 mai 2007**

**Salon Les Jeudis** - Emploi

Informatique & Ingénierie

Spécial Recrutement IT !

[www.lesjeudis.com](http://www.lesjeudis.com)

**Le 23 mai 2007, Paris 16e, La**

Maison des Arts et Métiers,

**BEA Dev2Dev TechDays 2007**

<http://eu.bea.com/event/tech-days/venue-paris.jsp>

**31 mai 2007, Google**

**Developer Day**, Conférence sur

les API Google et les outils de

développement, [www.google.com/events/developerday](http://www.google.com/events/developerday)

## JUN

**Du 05 Juin au 06 Juin 2007**

Paris - La Défense - CNIT

**SOA 2007**

**Solutions BPM 2007**

<http://www.infopromotions.fr/>

## SOA

### Software AG rachète WebMethod

Le marché de la SOA connaît aujourd'hui un mouvement important avec le rachat de webMethod par Software AG qui cherche à prendre une part significative du marché grâce à sa gamme CrossVision. Ce rachat va permettre à SoftwareAG de renforcer son offre, notamment dans le BPM. L'éditeur cherche à avoir une stratégie globale autour de la SOA et doit, pour ce faire, compléter ses outils, dans le BPM et sur un marché très prometteur, le portfolio et surtout la gouvernance. On doit donc s'attendre dans les prochains mois à une intégration entre les solutions des deux éditeurs. Pour Software AG, il s'agit aussi de croître et de saisir les opportunités de croissance offertes par la SOA. De facto, Software AG se place en concurrent des solutions IBM, Tibco ou encore Microsoft et BEA. Le BPM est devenu un secteur stratégique pour les éditeurs, qui est désormais arimé à la SOA. Fin mars dernier, Software AG et Mega International annonçaient l'arrivée de Mega dans la communauté CentraSite. Cette communauté doit aider les entreprises à comprendre, mettre en œuvre, créer des SOA. Mega apporte sa suite logicielle et son expérience.

## SYSTEME

### Symbian dévoile son nouveau système

L'éditeur Symbian a dévoilé la v9.5 de SymbianOS. L'éditeur a travaillé avec les constructeurs utilisant le système. Cette version se veut compatible avec les applications tournant sur SymbianOS v9.x. La 9.5 doit augmenter les performances et la montée en charge tout en améliorant la gestion de l'énergie. La gestion de la mémoire a subi un sérieux lifting, notamment dans la défragmentation de la RAM. Il s'agit aussi de mettre sur le marché le plus rapidement possible cette nouvelle version. La télévision numérique est intégrée par défaut. Pour les développeurs, on dispose d'une base SQL.

## GESTION



### Utiliser Fissum...

Fissum est un outil proposé sur CodePlex. Il s'agit d'un client permettant d'accéder aux "work items" de Team System sans avoir besoin d'ouvrir le Team Explorer ! On peut ainsi afficher les requêtes selon les projets, changer la personne assignée à une tâche, on peut même fil-

trer les requêtes. Pratique et efficace !

Site : <http://www.codeplex.com/fissum>

## PORTAIL

### Evaluant soutient CodePlex

CodePlex, le portail communautaire pour les projets .Net / Windows a reçu un soutien de poids avec l'éditeur Evaluant. L'éditeur français va mettre en ligne l'ensemble de ses travaux de recherche ! Pour la société, il s'agit de stimuler les projets. " Notre soutien et contribution active au sein du projet CodePlex est un prolongement naturel de notre partenariat historique avec Microsoft et les spécialistes de ses environnements " commente Nicolas Chabrier, Président d'Evaluant. D'ici fin 2007, ce seront 5 projets qui seront disponibles !

# Perforce

Gestion rapide de configuration logicielle (GCL)



## La fonctionnalité Folder Diff,

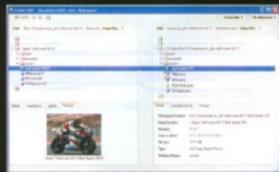
un atout de productivité du système de GCL Perforce.

Folder Diff est un outil interactif d'affichage en juxtaposition permettant de comparer l'état de deux groupes de fichiers.

À l'aide de Folder Diff, on peut rapidement déterminer les différences entre les fichiers situés dans des dossiers, des branches, des étiquettes ou sur votre disque local. Cette fonction est particulièrement utile lorsque vous devez réaliser des fusions de codes complexes.

De plus, si vous travaillez deconnecter, Folder Diff facilite la synchronisation des données avec celles du serveur Perforce lorsque vous vous connectez de nouveau au réseau.

Folder Diff n'est qu'un des nombreux atouts de productivité offerts par le système de GCL Perforce.



Folder Diff de Perforce

# PERFORCE

SOFTWARE

Téléchargez sans conditions une copie gratuite de Perforce sur [www.perforce.com](http://www.perforce.com). Vous bénéficiez d'une assistance technique gratuite pendant votre période d'évaluation.

## Sun : la machine virtuelle Java devient multi langage !

*Fin mars s'est tenu la grand' messe de Sun en France : les Sun TechDays. Véritable tour du monde de 14 grandes villes, l'étape française s'arrêta à Paris pour 3 jours. Plus grand événement européen pour l'éditeur, il a eu la chance d'avoir de nombreux speakers français et d'un contenu adapté. Comme l'an dernier, James Gosling profita de l'événement pour parler de Java et de son futur.*

Aucune annonce, aucune nouveauté ne fut au menu de l'événement. Il s'agissait tout d'abord de montrer la disponibilité des produits et des technologies promis lors de la JavaOne 2006.

### Java et son destin

James Gosling, l'inventeur de Java, a été très clair durant la session inaugurale : le futur de Java passe par la communauté et c'est à elle de proposer, de développer les nouvelles fonctions, les nouvelles évolutions ! " C'est la communauté qui sait, qui utilise Java " assure James. Quelle sera la place de Sun et de James Gosling dans cette nouvelle organisation ? Aucune réponse ne fut fournie. "Sun demeure dans le processus, dans les tests, dans l'ingénierie. Je discute avec la communauté sur ce que veulent les développeurs " poursuit James. OpenJDK officialise cette ouverture vers la communauté même si Sun conserve la marque Java et une forte influence dans la JCP. Ce fut aussi l'occasion pour l'éditeur de pousser en avant Java Temps Réel (Java RT), un chantier ambitieux et stratégique. " Ce fut un défi pour moi. Une early access (une pré-version, ndr) sera disponible courant du mois de mai. Un des problèmes est de déterminer le temps (dans les processus et la librairie) " commente James Gos-

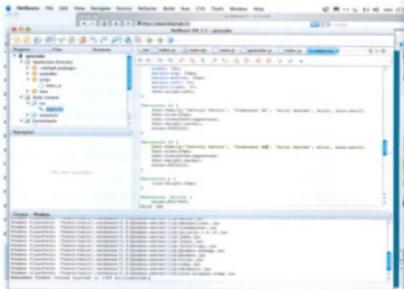
ling. Mais à côté de ce chantier, Sun travaille à intégrer, à interopérer Java et les possibilités du Web 2 : les langages de script, Ajax. Pour ce faire, le développeur Java dispose pour la partie Java de JMaki et Dynamic Faces. Pour le scripting dans Java, on dispose de Phobos, pour le REST on utilisera Restful Web Services API, WADL et ROMÉ API.

### JMaki

JMaki constitue un projet important pour Sun. Il s'agit d'un framework client - serveur léger pour créer des applications Web 2.0 à base de JavaScript, CSS. Sur le runtime client, reposent trois sections : le modèle widget, les services client JMaki et les couches layout JMaki. Sur la partie serveur, on dispose d'un runtime serveur et du XmlHttpProxy. C'est de là que l'on peut accéder aux ressources de l'entreprise et surtout aux services extérieurs.

### Project Dynamic Faces

L'objectif de ce projet est de pouvoir concilier la puissance des Java Server Faces et d'Ajax. Il s'agit donc d'obtenir de faire des applications Ajax plus " friendly " c'est-à-dire sans la lourdeur d'Ajax et de JavaScript. JSF 1.2 est la première version à supporter cela. L'implémentation se limite à un fichier Jar et une ligne de configuration dans le web.xml.



### Projet Phobos

Phobos est un environnement applicatif web léger, orienté scripting, fonctionnant sur une plate-forme Java. Pour le développement, on dispose d'un plug-in NetBeans, pour la partie Ajax, on dispose du framework JMaki, inclus par défaut. JavaScript est le premier langage script supporté par Phobos mais d'autres devraient s'ajouter. L'intégration NetBeans est plutôt réussie, notamment pour la partie design d'interface. La programmation JavaScript est réellement transparente dans l'IDE NetBeans.

### RESTful Web Services, WADL et ROMÉ API

RESTful a pour but d'accélérer le développement de services web en Java. Pour ce faire, Sun utilise l'architecture REST. WADL est une des nouveautés majeures dans le développement de web services en Java. WADL signifie Web Application Description Language. Pour faire simple, il s'agit d'un WSDL non pas pour SOAP mais pour REST. Cela devrait largement améliorer l'utilisation de REST comme fondation des web services. Enfin ROMÉ est une API capable de lire, de sauver et de convertir des flux RSS et Atom, le tout en Java. Cela devrait simplifier le travail des développeurs

pour implémenter RSS dans son application Java. On dispose par défaut d'un parser. Comme vous l'aurez compris, Sun espère prendre le virage Web 2 et surtout des langages de scripts en les intégrant dans Java. Il fallait simplifier et rendre plus aisé l'utilisation des scripting dans Java, tout en apportant un meilleur support XML dans la syndication des flux RSS. À cette panoplie déjà bien fournie, on peut rajouter deux autres projets Jruby et Groovy. "La JVM est multi langage, un peu comme la machine virtuelle .Net. Cela a surpris beaucoup de monde ", poursuit James. Pour ce faire, on dispose de Sun Web Developer Pack qui propose un package complet contenant l'ensemble des projets ci-dessus. Il doit permettre aux développeurs de se simplifier le développement dans NetBeans. Une release 2 est disponible depuis peu. Site : <http://developers.sun.com/web/swdp/>

### Et sur le langage Java

Si la communauté doit s'occuper du futur du langage, James Gosling nous a tout de même indiqué quelques pistes " les API de concurrence, la compilation side by side, simplification de la syntaxe, un style de codage plus

Fournisseur d'outils de développement depuis plus de 15 ans.

## TOUT CE QU'IL FAUT POUR DÉVELOPPER :

### DEVELOPPEMENT.NET

#### Visual Studio 2005 Professional

Des outils conçus pour relever les défis de tous vos développements



La gamme Visual Studio 2005 Professional vous permet de travailler dans un environnement de développement performant avec des outils de modification et de débogage ainsi que des concepteurs graphiques intuitifs qui vous permettent de bâtir rapidement des applications Windows, web et mobiles.

#### Avec abonnement MSDN Pro

Comprend Visual Studio Pro avec en plus SQL Server, Tous les principaux Systèmes dont Vista, et la gestion 2 incidents.

à partir de, par an, **549€ HT**

LICENCE OPEN 2 ans

#### Avec abonnement MSDN Premium

Comprend Visual Studio Pro avec en plus tous les produits Microsoft Server (Exchange, Windows, SQL), Tous les produits Office, dont Office 2007, Visual Source Safe, MSN, et la gestion 4 incidents.

LICENCE OPEN 2 ans

à partir de, par an, **1239€ HT**

#### Visual Studio 2005 Team System

- Visual Team Tester, Développeur, ou Architect
- Visual Team Foundation Server
- Visual Team Suite
- Visual Team Database

à partir de **3899€ HT**

Nous consulter pour devis sur mesure



### COMPOSANTS NET ASP JAVA

#### NetAdvantage for .NET 2006 Volume 3

à partir de **725€ HT**

#### NetAdvantage for JSF 2006 Volume 2

à partir de **575€ HT**

### OUTIL DE CREATION D'AIDE

#### ComponentOne Doc-To-Help 2007

La solution la plus connue du marché pour créer l'aide en ligne et la documentation des applications que vous avez développées. Nouvelle version 2007

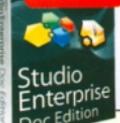
à partir de **820€ HT**

#### Doc-To-Help for Word 2007

\*La CD Studio Enterprise Doc Edition est un pack comprenant Doc-To-Help et la suite de composants Studio Enterprise

à partir de **615€ HT**

à partir de **1150€ HT**



### CONTACTEZ NOUS AU

**01 56 20 24 20**  
www.sienerinformatique.com

simple ". Avec l'arrivée des applications riches, comme avec Adobe Appollo ou WPF de Microsoft, où en est Java ? " Java est présent sur le desktop, notamment avec Java Web Start " avance James. Cependant, Java demeure en retard par rapport à Adobe et à Microsoft.

#### Glassfish mis en avant !

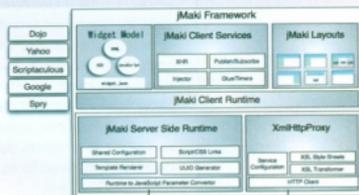
Glassfish est l'implémentation de référence de Java EE 5, utilisable en production. Sa version commerciale est Sun Java System Application Server 9.x. Si aujourd'hui, le projet est sous la licence open source Sun, il est envisagé de passer à la GPL v2 avec l'exception classpath. La version 2 arrivera en finale sans doute vers août prochain. L'environnement a été largement mis en avant durant la rencontre.

La version 2 apporte de nouvelles briques dans les web services, la montée en charge, la réplication, le scripting via Phobos, le clustering. Sur la partie service web, on dispose de JaxB, Jax-WS et stAX. Sur JaxWS, il s'agit de la version

2.1 apportant le clustering et une meilleure haute disponibilité, avec support du JBI ainsi qu'une bonne interopérabilité avec .Net. Sur l'interopérabilité, on dispose du projet Tango. Dans la v1, il manquait des fonctions orientées entreprises dans le déploiement, le clustering dynamique, la réplication en mémoire. Le projet Shoal, basé sur JXTA (un protocole P2P), doit faciliter les échanges. La réplication en mémoire se fera sans ajout dans Glassfish (on passera par le profil cluster). Côté performance, cela entraînera une dégradation. L'autre solution est de passer par HADB pour assurer la disponibilité. Cela remplace la réplication en mémoire même si la mise en place demeure plus complexe. Sur la v3 qui arrivera peut être courant 2008, l'architecture de Glassfish sera entièrement remodelée. Le scripting sera amélioré, tout comme la réplication. Un des axes fondamentaux sera la modularité. Côté développeur, il faut s'attendre à aussi à de profondes modifications.

■ François Tonic

## Installer SWDP dans NetBeans



Sun Web Developer Pack s'utilise avec NetBeans (pas de plug-in Eclipse pour le moment). Le plus pratique, et le plus rapide, est de passer directement par le centre de mise à jour de l'outil. Vous pouvez soit télécharger le Web Developer Pack dans son entier, soit installer uniquement les principaux éléments (Phobos / J2Maki). L'installation nécessite un redémarrage. On aura alors accès à deux nouveaux projets. On accédera ensuite au code JavaScript et aux CSS.

# Adobe Creative Suite 3 Web Premium

Attendue avec impatience par les créatifs et développeurs web, Creative Suite d'Adobe se décline désormais en plusieurs éditions spécialisées. La suite Web Premium se veut la Rolls de l'environnement complet pour le designer et développeur web. Voyons cela de plus près.

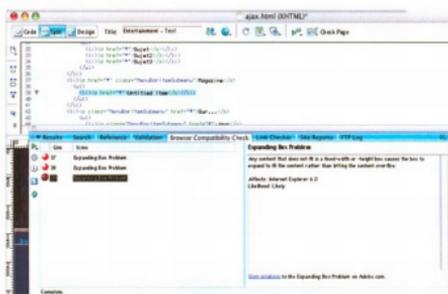
Nous avons testé une version bêta de la Web Premium sur un MacBook Dual Core et un PowerMac G5 (bi processeur). L'installation complète nécessite plus de 5 Go sur le disque dur ! Elle fonctionne sur PowerPC, Mac Intel et est recommandée pour Windows Vista. Une configuration musclée est indispensable pour exploiter au mieux les outils.

## Dreamweaver revu et corrigé

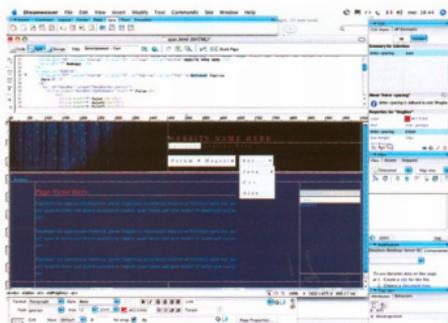
Premier changement important de la Creative Suite, la présence de Dreamweaver (suite au rachat de Macromedia) en lieu et place de GoLive. Bien que GoLive n'appartienne plus à la Creative Suite, l'outil devrait continuer à être disponible (même si à terme, sa pérennité n'est pas garantie). Si par le passé nous avons souvent pointé les faiblesses de l'interface utilisateur dues à un manque d'ergonomie, reconnaissons que le travail d'Adobe fournit un outil de développement web doté d'une interface pratique et relativement intuitive. De plus, on dispose d'une interface unifiée entre les différents outils ce qui facilite la prise en main même si les habitués de Dreamweaver et même de la CS 2 devront modifier leurs habitudes mais en contre partie, cela facilitera le travail entre les différents outils. D'ailleurs, l'intégration de l'interface n'est pas la seule nouveauté. Les échanges entre les différents outils se font de manière plus rapide. Ainsi les images provenant d'Illustrator ou Photoshop peuvent être collées dans une page Dreamweaver, ce dernier gérant ensuite le "coller".



L'éditeur de Device Central



Vérification de compatibilité



L'éditeur de Dreamweaver

Si votre image comporte plusieurs calques et de la transparence, il faut faire une copie mergée (Copy Merged) pour prendre en compte l'ensemble des éléments.

Au niveau des CSS, cette version améliore considérablement la chose pour simplifier leur usage et conception avec éditeur graphique.

En tant qu'utilisateur, nous avons été plutôt séduit, il simplifie réellement le travail ! Ajax est l'autre gros morceau, avec l'ajout du framework maison, Spry. Malheureusement, Dreamweaver n'accepte que ce framework, il n'est pas possible de le changer ou d'intégrer des objets Ajax non Spry. Du moins pour le moment.

La communauté proposera rapidement des composants Spry pour compléter les quelques objets disponibles. Leur mise en œuvre est simple. Il suffit d'aller dans l'onglet Spry (barre d'outils du haut), de sélectionner la zone prête à accueillir le composant et de cliquer sur son icône.

Le composant s'affichera dans sa page. Sa configuration se fera par les onglets de propriétés du bas. On a bien entendu accès au code JavaScript.

L'accès à ces propriétés et options se révèle un peu laborieux au début, mais après un peu de pratique, on arrive à être plus productif, plus rapide.

Surtout, on peut afficher les paramètres selon la balise que l'on sélectionne. Le mode split (code-design) est plus souple qu'auparavant et les changements peuvent se faire alternativement sur les deux vues.

### Une meilleure vérification...

Pour vérifier la conformité des codes de ces pages avec les différents navigateurs (Safari, IE, Firefox), on accède à la fonction Validate markup. Très pratique pour voir d'un coup d'œil les problèmes... À chaque fois, il indique le problème, la localisation, avec une courte explication. Un double clic affiche le code en cause. Pour aider le développeur, on a accès aux dictionnaires O'Reilly quand on souhaite plus d'information sur une balise, une fonction. On dispose aussi de l'auto complétion dans l'éditeur.

S'il n'existe pas d'outil de migration des projets GoLive, en théorie, leur reprise dans le nouvel environnement ne doit pas poser trop de problèmes, même si nous avons quelques soucis avec du code JavaScript. D'autre part, il

n'est pas possible non plus de migrer ses comptes WebDav et FTP, il faut tout ressaisir.

### Quid du mobile ?

Le mobile est devenu pour Adobe un axe stratégique. Cependant, il ne cible pas spécifiquement Dreamweaver. Il n'y a pas de liens forts entre Device Central et Dreamweaver. Device Central est surtout orienté contenu. Cependant, on peut tout de même créer des pages pour mobile, notamment avec WML. Device Central permet de valider son contenu (Illustrator, Flash ou Photoshop) suivant les profils des terminaux. Espérons qu'une intégration forte existera avec Dreamweaver.

Pour faciliter la navigation pour les développeurs et designers, on dispose de Bridge. On accède rapidement aux fichiers, docu-

ments, projets. Pratique à l'usage, il devient vite indispensable. Son interface est plutôt réussie même si elle diffère des autres produits. Pour la gestion collaborative, on dispose de CUE CS3 pour travailler à plusieurs, partager les documents et fichiers, centraliser la gestion. Contribute fait lui aussi peau neuve, avec de nombreuses nouvelles fonctions très attendues comme la compatibilité avec Vista et Mac Intel, les dernières versions des navigateurs, la prise en compte du PDF, intégration avec Office, le support des fichiers FLV, prise en charge du blog. Pour ces derniers, on peut rencontrer des problèmes de connexions sur des serveurs non listés. Cette pré-version laisse une excellente impression, même si nous avons eu quelques petits soucis d'affichage. L'éditeur offre

une belle ergonomie, un accès rapide aux fonctions, on retrouve une influence de GoLive... On dispose là d'un environnement intégré même si le côté mobile reste en retrait. La CS3 apporte un argument de poids face à la gamme Expression de Microsoft. Adobe peut miser sur l'aspect multi langage et multi-plate-forme.

■ François Tonic

### Creative Suite 3 Web Premium (bêta)

Éditeur : Adobe

Configuration : un PC ou Mac récent (G5 ou Mac Intel), minimum 1 Go de Ram (2 - 3 Go conseillés), MacOS X ou Vista.

⊕ nouvelle interface, intégration des outils, fonctions CSS, Ajax, multi-plate-forme

⊖ Ajax maison, mobilité

Utilisation optimale des  
processeurs multi-core



Vos logiciels n'utilisent pas tout le potentiel  
des processeurs multi-core ?

Intel et Rogue Wave ont des solutions logicielles  
à vous proposer.

**Vous êtes conviés à une présentation  
suivie d'un atelier.**

Inscription et renseignements

Contactez Emmanuel BLATTES, Ingénieur d'Applications  
Par mail : eblattes@roguewave.com  
Par téléphone : 01 30 09 78 78

Prochaine session :

Mardi 22 mai  
de 9h à 12h30

Adresse :

INTEL Paris FSO  
Les Montalets  
2, rue de Paris  
92196 Meudon

## GESTION

## Microsoft planifie Dynamics Nav



L'éditeur vient de présenter la 5e version de son ERP : Dynamics Nav. Cette version introduit un grand nombre de nouveautés dans tous les domaines fonctionnels (administration, achat, vente, acomptes, etc.). L'outil collabore bien mieux avec Office et notamment Outlook et MapPoint. On dispose aussi d'une recherche locale via Windows Live Local Search. Cette version devrait permettre à Microsoft de continuer à croître rapidement sur ce marché. L'éditeur prépare déjà la relève avec la v5.1 qui devrait arriver fin 2007. Cette version apportera de nombreuses nouveautés : support de .Net 2, meilleure intégration avec Office 2007, synchronisation avec Outlook améliorée. Les puces RFID feront leur apparition avec cette mise à jour, ce qui devrait renforcer l'attrait du produit dans la logistique. Une version 6.0 est déjà planifiée pour 2008 - 2009.

## ADMINISTRATION

## Simplifiez-vous PowerShell

Avec PowerShell, Microsoft dote enfin Windows d'un shell digne d'Unix. Mais encore faut-il savoir l'utiliser et surtout le maîtriser. Pour aider les utilisateurs, Quest Software propose une interface graphique : PowerGUI. Elle permet d'accéder à l'ensemble des fonctions sans avoir besoin de les connaître. Parallèlement, l'éditeur ouvre un site communautaire pour partager son expérience ([www.powergui.org](http://www.powergui.org)). D'autre part, l'éditeur a conçu des commandes PowerShell spécifiques pour ActiveDirectory Lightweight Directory Services. À noter que PowerGUI est gratuit. Site : [www.quest.com](http://www.quest.com)

## SERVEUR

## Home Server se dévoile enfin !

À quelques semaines des premiers serveurs personnels, Microsoft a dévoilé les éléments de Home Server. Pour le développeur, on dispose d'un SDK spécifique contenant les API, la documentation et des exemples. Il faut disposer de Visual Studio 2005 ou d'un C# Express et de .Net 2. Les API permettent d'accéder aux clients matériels, aux dossiers partagés sur le serveur (et aux dossiers applications), à la gestion des disques durs, à la notification ou encore au backup. Il est aussi possible d'étendre la console Windows Home Server. À vous de jouer !

## SGBD

## L'actualité base de données en bref :

- **McObject** dévoile la version 3.1 d'extremeDB, une base de données embarquée temps réel. Cette version inclut une fonction d'index, des performances et une montée en charge améliorée.
- **EMC** améliore le stockage pour SQL Server 2005 et Exchange Server 2007. On peut maintenant gérer, optimiser, planifier la sauvegarde de ses données, tout en mettant en place leur recouvrement.
- Depuis peu, une nouvelle version d'**Apollo VCL** est disponible. Il s'agit d'un moteur de base de données supportant Delphi 2005, 2006 et 2007.
- L'éditeur **EMS** rend disponible une nouvelle version de SQL Manager pour Oracle. Cette nouvelle édition supporte les bases statiques, les objets Tables, apporte aussi des corrections de bugs. L'éditeur annonce aussi une mise à jour de DB Extract, outil pour créer des sauvegardes des bases de données.
- **SQL Maestro** Groupe lance la version 7.3 d'Oracle Maestro, un puissant outil graphique pour gérer, administrer le serveur SGBD d'Oracle et la partie développeur. Cette version améliore le mode debug pour le PL/SQL et complète les outils de conception de bases de données.

## DÉPLOIEMENT

## Améliorer la diffusion des applications

F5 Networks se propose de vous aider à gérer la diffusion des applications sur le réseau, avec Application Ready Network. Il s'agit d'une infrastructure de réseau applicatif adapté à Office 2007. L'éditeur prévoit de supporter Office Communication Server 2007 prochainement. L'outil cherche à supprimer les goulets d'étranglements et tout ce qui pourrait nuire à un bon déploiement des applications. " Nous avons été impressionnés par la vision globale qu'a F5 de l'application et du réseau ainsi que par son souci de mettre le réseau au service des déploiements Microsoft ", commente Rob Bernard, Directeur Général ISV (Indépendant Software Vendor) Monde de Microsoft. " L'architecture complète conçue par F5 va optimiser les performances des applications pour les clients de Microsoft. " L'outil a été testé pour les applications Microsoft. Côté développement, le développeur peut utiliser les fonctions de F5 comme des services.

## DÉPLOIEMENT

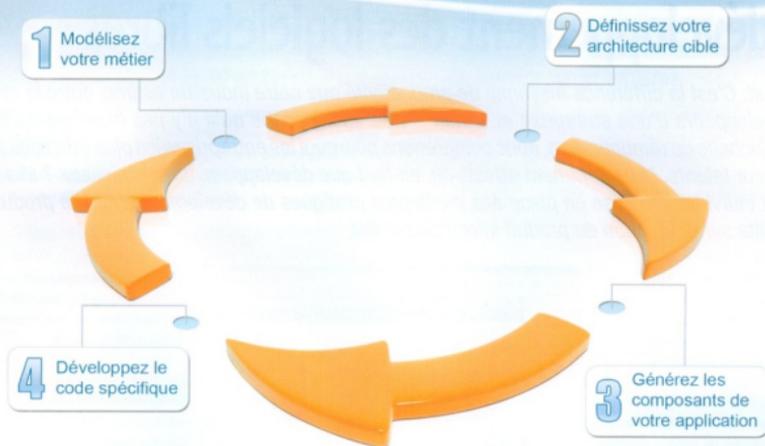
## Altiris optimise l'installation

L'éditeur lance la version 7 de Wise Installation Studio. Il supporte Vista, Windows Mobile 5 et les éditeurs 64-bit de Windows. L'éditeur propose dans le studio sa technologie Virtualization Solution. Il s'agit de pouvoir installer une application (et ses données) dans une image virtuelle. Cela facilite le déploiement et la désinstallation ! En cas de corruption, il suffit de restaurer l'image virtuelle. L'installation sur les postes se fait par simple copie de l'image. Cela apporte une souplesse d'utilisation sans précédent ! Donc on virtualise uniquement l'application. D'autre part, cette version 7 intègre aussi Naiperion Pro-Testor Standard qui permet de créer des versions d'essai. Une version Express est disponible.

# CodeFluent

Accélérez vos développements .NET

Réalisez votre application métier à l'aide de la première usine de composants à la demande.



## CodeFluent.com

La fabrique en ligne est totalement gratuite pour les modèles métier comprenant jusque 30 entités.



nouveau !

Inscrivez-vous dès aujourd'hui sur :  
<http://www.codefluent.com>

0€

## CodeFluent

Version complète Développeur donnant accès aux codes sources des composants générés.

Version d'évaluation sur le CD-Rom encarté dans ce magazine



Prix public : ~~2490€~~ttc

Prix spécial lecteurs de "Programmez" : **1990€**ttc

CodeFluent est une usine logicielle qui automatise la création des composants de votre application métier .NET selon les meilleures pratiques des experts de la plate-forme Microsoft.



SoftFluent est une société spécialisée dans le développement sous plate-forme Microsoft. Elle compte déjà de nombreux clients prestigieux tels qu'ILOG, VCS Timeless, TF1 ou Microsoft France et Europe. Contactez-nous sur [info@softfluent.com](mailto:info@softfluent.com)

SoftFluent 46, rue Auguste Blanqui 94250 GENTILLY - 01 46 16 05 02

Microsoft | PROGRAMME IDÉES  
L'INNOVATION COMBINEE ET

## Les meilleures pratiques de développement des logiciels libres

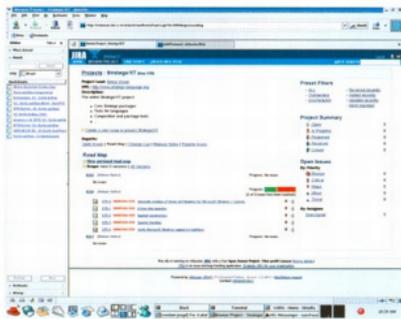
*Vingt-huit. C'est la différence moyenne de productivité que notre industrie estime, entre le meilleur des développeurs d'une entreprise et le moins bon. Etant donné qu'il n'y pas le même facteur en terme d'échelle de rémunération, nous comprenons pourquoi les entreprises les plus efficaces savent retenir leur talents. Mais comment atteint-on, en tant que développeur, de tels niveaux ? Mis à part le talent individuel, la mise en place des meilleures pratiques de développement offre productivité et visibilité sur la création du produit informatique fini.*

Dans cet article, nous montrons, autour de 8 points clés, comment l'ensemble des bonnes pratiques du développement peut contribuer significativement à améliorer les chances de succès des logiciels demandés. Gardez cependant à l'esprit que leur seule mise en place n'est pas garantie de succès. Comme toujours dans le développement, il n'y a pas de "silver bullet" (<http://www-inst.eecs.berkeley.edu/~mararb/readings/NoSilverBullet.html>). Cela est vrai pour la technologie mais aussi les meilleures pratiques. C'est la combinaison de l'ensemble des acteurs du développement qui mènera le projet à bien, mais sans la bonne culture d'entreprise et les processus de développement s'intégrant autour d'une équipe compétente, les chances du succès s'en trouvent sérieusement diminuées.

### Culture du management

Le rôle du management d'une entreprise informatique est d'abord de créer les conditions de travail permettant aux meilleurs des développeurs de s'exprimer. Souvent, la technologie n'est pas mise en valeur, voire dépréciée. Pourtant, si votre vie devait dépendre du fonctionnement d'un logiciel, que voudriez vous savoir sur ce logiciel ? Probablement qu'il a écrit...

Nous devons remettre l'activité du développement au cœur du projet informatique, et non pas comme une tâche considérée annexe et souvent "outsourcée". La culture d'entreprise doit réintégrer le facteur humain et de créativité autour de ses équipes que les méthodes de gestion ont écarté. Le code doit être valorisé car il est le seul garant de la bonne fonction



du logiciel, et celui qui l'écrit doit en recevoir les honneurs. Les entreprises qui ont cette culture attirent le top, ceux qui sont 28 fois plus performants et il est de notre responsabilité, en tant qu'ingénieurs, de contribuer à établir cette culture dans nos entreprises.

Une fois que la culture de l'entreprise est en phase avec l'activité de création informatique, nous devons mettre en place ces fameuses meilleures pratiques qui font si souvent la différence.

### Méthodologie de développement

Entre les adeptes du cycle en V, et ceux des méthodologies agiles, le débat est vaste ([http://en.wikipedia.org/wiki/Software\\_development\\_methodology](http://en.wikipedia.org/wiki/Software_development_methodology)). Il n'est pas dans le cadre de cet article de peser avantages et inconvénients de chacune, mais sans méthodologie, il est difficile d'atteindre un bon niveau de productivité. Une équipe doit travailler ensemble, autour d'un processus qui découle de la méthodologie de développement choisie, ce processus devant au minimum

coordonner la communication entre les acteurs impliqués, des spécifications fonctionnelles, au développement, jusqu'au contrôle qualité.

### Gestion des sources

Le logiciel est le code source. Il est au cœur de tout : il implémente les spécifications fonctionnelles et techniques, il est la qualité, les performances finales etc.... Son rôle central justifie des procédures de sauvegarde et des outils nécessaires à sa propre gestion.

Il est pourtant étonnant de constater que beaucoup d'entreprises, y compris certaines de grande renommée, n'ont

pas les processus en place pour gérer le code source qu'elles produisent.

Bien géré, le code source doit être sauvegardé, et stocké dans un endroit physiquement séparé, à l'abri des événements naturels (eau et feu en particulier).

L'autre pilier d'une bonne gestion des sources est la capacité de créer plusieurs versions, ou branches, du code. Il est important d'être en capacité de reconstruire une version déjà dérivée et aussi de la patcher, ou d'en faire une autre version, tout en continuant de construire les futures évolutions du logiciel. Beaucoup d'outils aident à faire cela, mais ils doivent être associés à un processus connu de chaque développeur. Dans le monde open source, on peut citer CVS (Concurrent Versions System - <http://www.nongnu.org/cvs>) ou Subversion (<http://subversion.tigris.org>), plus moderne.

### Environnement de développement

Ces dernières années, en particulier avec l'avènement de Java et C#, les environne-

# ihm

EN TOUTE SIMPLICITÉ !



Une application multilingue avec plus de 100 vues de 20 types différents, en DHTML/Ajax, en Swing ou en plugin Eclipse, connectée à un SGBD et un bus JMS.

“ il vous faut  
combien de temps  
pour la réaliser ? ”

Si votre réponse est moins d'une semaine, inutile de vous rendre sur notre site, ni de télécharger la version gratuite de LEONARDI, sinon...



LEONARDI

NOUVELLE  
VERSION 3.3  
DISPONIBLE !

ments de développement ont beaucoup progressé. Dans le monde des IDE (Environnement de Développement Intégré) Open Source, Eclipse s'est imposé comme une référence aujourd'hui incontournable (<http://www.eclipse.org/>). NetBeans (<http://www.netbeans.org/>), suivi de près par SUN, est une alternative open source également de grande qualité. Bien développés pour Java, ces outils supportent un grand nombre de langages et sont multi-plates-formes, ce qui permet souvent de développer sur la plate-forme qui fera tourner ultimement le logiciel, permettant d'anticiper des problèmes de compatibilité éventuels.

Un bon environnement de développement intégré est important car il permet au développeur de découvrir ses erreurs plus tôt (en temps réel dans le cas de Java) et offre des outils de refactoring (<http://www.refactoring.com/>) qui améliorent sa productivité.

## Intégration continue

Les outils d'intégration continue prouvent leur efficacité. Combinés avec un bon outil de gestion des sources, ils permettent une grande traçabilité de qui a fait quoi sur quelles parties du code source, à quel moment, et si la modification a mis un ou plusieurs tests unitaires en erreur (<http://www.martinfowler.com/articles/continuousIntegration.html>)

Ces outils permettent de savoir en temps réel les effets des modifications de chaque membre d'une équipe de développement, et sont maintenant indispensables sur des équipes importantes.

Dans le monde libre, citons CruiseControl (<http://cruisecontrol.sourceforge.net/>) qui est aujourd'hui le plus utilisé.

Il est intéressant de voir l'émergence d'outils de gestion de projets qui coordonnent l'intégration continue et gestion des sources et de sa documentation dans un ensemble cohérent.

Maven (<http://maven.apache.org>), logiciel open source, fait une entrée remarquée dans le monde de l'entreprise car il est ouvert, via un système de plug-in, à la plupart des outils aujourd'hui utilisés.

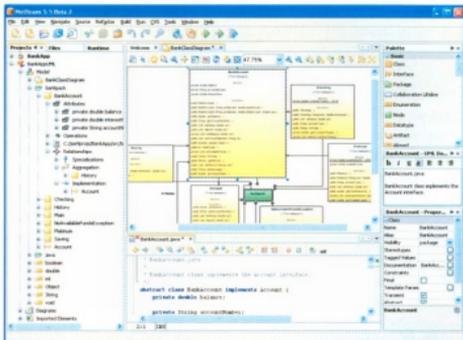
Notons que Maven est aussi intégré à Eclipse et NetBeans, ce qui en fait une suite particulièrement cohérente.

## Gestion des bugs

Les bugs sont la plaie de notre industrie. A défaut de pouvoir faire du logiciel sans bug, nous nous devons de les gérer le mieux possible. Le constat peut être dur à admettre pour un informaticien, mais les logiciels que nous fabriquons sont criblés de bugs.

Ils apparaissent à l'utilisateur final dans les mauvais développements, ils n'apparaissent pas dans les bons. Dans les deux cas, les équipes de développement doivent garder trace des bugs découverts, dans quelles versions ils ont été découverts, dans quelles versions ils ont été fixés, et, s'ils sont toujours ouverts, dans quelles versions ils existent.

Utilisé depuis quelques années, Bugzilla (<http://www.bugzilla.org/>) offre une solution open source. JIRA (<http://www.atlassian.com/>)



software/jira/) s'est aussi imposé et est disponible gratuitement à condition que le logiciel développé soit lui-même open source.

## Tests, contrôle de la qualité et de la régression

Nous, développeurs, ne testons pas assez. Nous n'aimons pas tester ce que nous faisons. Pourtant, c'est ce qui fait la différence. Pour pallier notre tendance naturelle, des pratiques ont émergé. Les tests unitaires sont maintenant intégrés aux environnements de développement. Junit (<http://www.junit.org>), librairie open source, est le standard du monde Java, mais Junit ne suffit pas.

Les tests de non régression doivent être intégrés au processus de développement. Les tests de non régression doivent apporter un degré de confiance significatif sur le fait que la version 3 fait au moins aussi bien que la ver-

sion 2, c'est-à-dire que tout ce qui fonctionne dans la version 2 fonctionne dans la version 3. Sans cet investissement, la confiance autour du logiciel ne peut pas s'établir.

Un troisième série de tests doit ensuite être mise en place : Les tests d'intégration. Les systèmes ne fonctionnent pas en isolation les uns des autres mais s'intègrent de plus en plus dans une architecture d'entreprise cohérente. Il faut donc s'assurer que le dérivable s'intègre bien avec les autres systèmes existants mis en place.

## Culture du développeur

Que valent ces pratiques si l'équipe n'est pas compétente ? Rien. Rien ne remplace le talent individuel. Que valent-elles si l'équipe n'est pas soudée autour du produit qu'elle délire ?

Pas grand-chose, car c'est l'élément le plus faible qui détermine une bonne partie de la qualité de l'ensemble.

Au delà des pratiques, le développeur doit être intégré et intégré, aider, parrainer, prendre sa responsabilité, être engagé envers le dérivable, être passionné, rechercher les améliorations, ne rien laisser passer, être paranoïaque sur la qualité, et être, parfois, raisonnablement pénible lorsqu'il s'agit de s'assurer que les meilleurs pratiques sont respectées.

Il est intéressant de constater l'importance de la contribution du monde libre des dernières années, Eclipse et Netbeans ont émergé, Junit, CruiseControl, Maven. Subversion ont tous émergé dans un contexte open source, notamment pour pallier des problématiques d'équipes de développement distribuées.

Il faut aussi noter que ces outils s'intègrent les uns les autres et donc combinent leurs fonctionnalités entre eux. Nous exécutons les tests unitaires depuis Eclipse ou Netbeans ou CruiseControl. CruiseControl recopie les sources depuis Subversion ou CVS. Nous contrôlons le projet dans son ensemble depuis Maven. Cet ensemble intégré et de haute qualité libère les gains de productivité nécessaires qui permettent aux plus talentueux d'entre nous d'exprimer pleinement leurs talents.

■ Sébastien Marc

## WEB

### Adobe s'attaque à Windows Media Video et Quicktime

L'éditeur décide de s'attaquer frontalement à la diffusion de la vidéo sur Internet que Microsoft et Apple tentent d'accaparer. Adobe a dévoilé son propre lecteur : Adobe Media Player. Il s'agit d'une application de bureau, complémentaire aux fonctions disponibles en ligne. Media Player se veut léger en taille mais offrant une qualité de la vidéo supérieure. Il utilisera le format vidéo Flash et inclut RSS et SMIL. Des mécanismes de protections seront disponibles. Il s'appuie sur la plate-forme Apollo. La disponibilité n'est pas attendue avant fin 2007.

## PROCESSEUR

### Intel optimise le code

Le fondeur vient de sortir les nouvelles versions de ses outils Thread Checker, Thread Profiler, Vtune, Threading Building Blocks. Ils doivent permettre aux développeurs de coder des applications plus stables et performantes avec les processus multi cœur (processeurs Intel Core 2) et les systèmes les plus récents (Windows et Linux). Intel Thread Checker 3.1 détecte les erreurs de parallélisation au niveau du codage et permet ainsi aux développeurs de paralléliser plus rapidement leurs logiciels tout en prenant moins de risques. Il est capable de détecter les erreurs de parallélisation les plus difficiles à déceler dans les logiciels 64 bits. Ses performances ont été optimisées dans une optique de

réduction de la taille de l'image instrumentée et de la durée d'exécution, ce qui accélère encore l'analyse. Thread Profiler 3.1 examine le comportement des threads et repère les problèmes de performances de parallélisme, dont les goulets d'étranglement. Il permet aux développeurs de trouver le nombre de coeurs auxquels une application fait effectivement appel et indique la proportion de code qui n'a pas été parallélisé de manière optimale pour la plate-forme matérielle concernée. Threading Building Blocks 1.1 est une bibliothèque de modèles de programmation C++ pour la parallélisation d'un logiciel et favorise des gains de performances tout en épargnant aux développeurs l'opération fastidieuse qui consiste à répartir les threads entre les coeurs d'un processeur. VTune Performance Analyzer 9.0 prend désormais totalement en charge la programmation pour Windows Vista, notamment au travers du nouveau pilote signé d'Intel pour ce système d'exploitation, qui s'installe sans blocage de sécurité par ce dernier. Dans sa version Linux, l'outil établit un nouveau lien entre l'analyse du compilateur Intel et celle qu'il effectue lui-même sur la durée d'exécution pour les " points chauds " (" hotspots ").

## QUALITE

### DataFlux aide vos données

L'éditeur DataFlux propose DataFlux Accelerator pour aider à la qualité des données et leur intégration. Il s'agit d'une suite d'outils dont le premier élément est Accelerator for Customer Data Analysis, dédié plutôt à l'analyse en entreprise. D'autres modules suivront courant 2007.

Site : [www.dataflux.com](http://www.dataflux.com)

Accordez-vous plus de temps.

[www.bluage.com](http://www.bluage.com)

La solution de génération d'applications par NETFFECTIVE TECHNOLOGY.

Implémentation pragmatique du standard MDA (Modern Driven Architecture), BLU AGE™ adopte une démarche basée sur le modèle et non sur la technologie, permettant une nouvelle approche de la réalisation de projets, et révolutionnant la conception des applications métiers.

# BLU AGE™

J2EE .NET APPLICATION GENERATOR  
2 0 0 7 E D I T I O N

BLU AGE™ est une marque déposée NETFFECTIVE TECHNOLOGY Toutes les marques citées sont la propriété de leurs propriétaires respectifs

Le logiciel BLU AGE™ est destiné à être utilisé en priorité par les consultants métier/BPM, les architectes techniques et les analystes concepteurs. BLU AGE™ permet d'interpréter les spécifications détaillées au format UML/OCL et XHTML pour générer automatiquement des

applications sur les environnements J2EE & .NET. En interprétant une modélisation de haut niveau des processus métier indépendants des technologies, BLU AGE™ valorise les investissements BPM des entreprises.

Venez assister à une démonstration en ligne de génération automatique d'application réalisée avec BLU AGE™ Edition 2007 sur [www.bluage.com](http://www.bluage.com).

Partenaires :  



# Synchronisation avec ADO.net

Les domaines dans lesquels on souhaite être déconnecté du serveur de base de données sont nombreux. De la simple mise en cache pour améliorer les performances (et ainsi économiser les ressources du serveur de base de données) à des situations où l'on ne peut fonctionner qu'en mode déconnecté (par exemple un commercial itinérant qui ne synchronisera que périodiquement ses données avec la base centrale).

On supporte partiellement le principe de données déconnectées avec ADO.net, mais essentiellement pour des problématiques de mise en cache. Les systèmes de synchronisation plus poussés (tels que la réplication de fusion de SQL Server) sont en plus capables de gérer les données si l'un des systèmes est totalement hors ligne, d'effectuer des mises à jours incrémentielles et de résoudre les conflits se produisant. Les modules ayant ces fonctionnalités actuellement dans la plate-forme Microsoft sont RDA (Remote Data Access) et la réplication de fusion avec SQL Server. Le premier étant davantage orienté développeur que le second.

La couche Synchronisation Services d'ADO.net (aussi labellisée Occasionally Connected Systems / OCS) est une brique pour développer à mi-chemin entre RDA et la réplication de fusion. Elle offre tous les avantages de RDA (sauf le support de périphériques mobiles pour le moment), n'est pas spécifique à un moteur de base de données et sait gérer les données de manière incrémentielle. De plus, le système est extensible, permettant par exemple d'ajouter la communication entre client et serveur par Web Service.

## SQL Server CE, Mobile, Everywhere, Compact

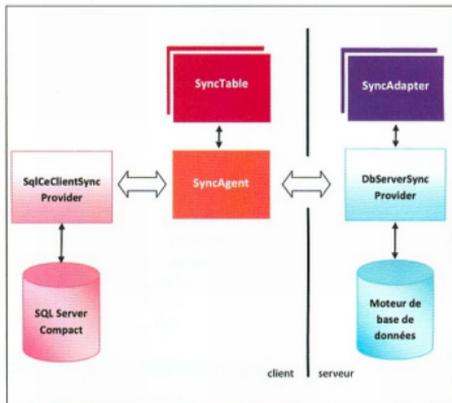
La partie cliente de la couche de synchronisation d'ADO.net repose sur une édition particulière de SQL Server 2005, appelée Compact. Historiquement, cette édition de SQL Server n'est autre que l'édition mobile / CE de SQL Server, qui n'était destinée au départ qu'à l'utilisation sur appareils mobiles (Pocket PC, Smartphone, etc.).

Au gré des versions et changements de nom, il est devenu évident que cette édition avait aussi sa place sur les postes de travail. L'intérêt étant d'avoir une plate-forme de données unique, que l'on soit sur appareils mobiles ou sur une station de travail. L'autre avantage non négligeable de cette édition est d'avoir une base de données très proche de SQL Server, sans être un serveur de base de données, mais une base de données à base de fichier. La syntaxe est en effet très proche ainsi que les types de données.

Pour la synchronisation, il s'agit de la version 3.5 (bêta) de SQL Server Compact que nous utiliserons. Ses bases de données peuvent aller jusqu'à 4Go (l'extension par défaut étant SDF), le code de gestion de la base de données ne pesant que 1,4Mo. Il est aussi parfaitement envisageable de développer un autre système de mise en cache côté client, autre que celui livré basé sur SQL Server Compact.

## Principe et fonctionnement

Le système de synchronisation se décompose en 2 parties. La partie cliente utilisant SQL Server Compact Edition si vous vous servez du provider livré d'origine. La seconde partie est, elle, côté serveur, le provider par défaut n'étant pas basé sur un moteur particulier, toute source de



données .net ou OleDb est utilisable. L'espace de nom utilisé est **Microsoft.Synchronization**. Celui-ci est aussi décomposé entre client et serveur. La classe **DbServerSyncProvider** située côté serveur (dérivée de **ServerSyncProvider**) détient la connexion vers le serveur de base de données. C'est elle aussi qui fournit les informations servant à la mise à jour incrémentielle, les informations du schéma de base de données et les **SyncAdapter** décrits plus loin. Le code des événements (**ApplyChangeFailed**) pour la résolution de conflits est aussi à écrire dans cette classe. **SqlCeClientSyncProvider** est le pendant de **DbServerSyncProvider** mais côté client. Elle est dérivée de **ClientSyncProvider** et référence la base de cache SQL Server Compact.

Les **SyncAdapter** sont définis sous forme de collection dans **DbServerSyncProvider**, ils contiennent les requêtes SQL appelées en fonction de l'opération demandée (suppression, insertion, mise à jour de données incrémentielles ou non). Vous aurez à en créer un par table exposée du côté du client. La classe **SqlSyncAdapterBuilder** permet de générer les **SyncAdapter** pour une base de données SQL Server en réduisant le code (SQL et .net) au strict nécessaire. Du côté client, on trouve une collection de **SyncTable** dans **SyncAgent** qui détermine les tables auxquelles est abonné le client et le mode de synchronisation. Les **SyncGroup** permettent la mise à jour transactionnelle d'un ensemble de tables.

Le **SyncAgent** est le mécanisme central, il lie les 2 providers ensemble (client et serveur), et implémente les **SyncTable**. La synchronisation est déclenchée par la méthode **Synchronize()** de cette classe.

" Infragistics takes care of the UI,  
so I can **focus on the business logic.**"

-Application Developer

" We saved weeks of work using Infragistics components, in terms of the cost and time coding. Since their toolset contains both Windows Forms and ASP.NET controls, I'll already have the tools needed for our next project. "



Office 2007 UI with  
Application Styling



Windows Vista™ Aero™ Glass StyleSet

## Office 2007 UI for Windows Forms

# NetAdvantage® for Windows Forms 2007 Vol. 1

**Support complet d'Office 2007** - avec le RibbonMDI Merging, support pour Windows Vista™ Aero (Glass) et le nouveau Mini ToolBar

**Application Styling** - Design once, style everywhere. Le nouveau contrôle Express Styling™ permet de donner du style aux applications à partir d'une seule couleur

**Productivité** - Permettez rapidement le formatage conditionnel des données de la grille

**Performance** - Support complet de l'architecture .NET 2.0 en 64 bit incluant Windows XP 64-bit, Vista 64-bit et Windows Server 2003 64-bit

**Accessibilité du Web** - Tous les contrôles Windows Forms sont maintenant conformes aux normes de la Section 508 sur l'accessibilité du Web

Pour de plus amples informations:

[infragistics.com/Office2007](http://infragistics.com/Office2007)

[sales-europe@infragistics.com](mailto:sales-europe@infragistics.com)

 0800 667 307

**Infragistics**  
Powering The Presentation Layer

## Préparer le schéma de base de données

Au niveau du serveur de base de données il est indispensable de réaliser certaines modifications de schéma pour pouvoir synchroniser parfaitement les données. Pour les mises à jours de données incrémentielles, il faudra se baser sur une date et heure de création et modification ou sur un champ de type rowversion (timestamp sur SQL Server) permettant le suivi des enregistrements. Si vous souhaitez suivre les suppressions de données il sera alors nécessaire d'ajouter une table " tombstone " dont le rôle sera l'historisation des suppressions d'enregistrements, celle-ci alimentée par trigger. La résolution de conflit nécessitera toutes ces informations ainsi que la source des données et toute information que vous jugerez nécessaire.

## Exemple de code de synchronisation

Voici un exemple de chargement complet de table.

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Data.SqlServerCe;
using Microsoft.Synchronization.Data;
using Microsoft.Synchronization.Data.Server;
using Microsoft.Synchronization.Data.Client;
using System.IO;

namespace mesApp
{
    class TestSync
    {
        static void Main(string[] args)
        {
            SqlConnection srvCnx;
            SqlCommand cmdProduit;

            // Variable pour le service de synchronisation
            SyncAgent agentBdd;
            SyncTable tableProduit;
            SyncAdapter adapterProduit;
            SqlCeClientSyncProvider clientBddProvider;
            DbServerSyncProvider serverBddProvider;

            // Chaîne de connexion pour le client (SQL Compact) et le serveur
            (SqlConnection)
            string cnxServer;
            string cnxCliant;
            cnxServer = "...à renseigner...";
            cnxCliant = "...à renseigner...";

            // Crée notre agent de synchronisation
            agentBdd = new SyncAgent();

            // Renseigne notre SyncTable pour la table Produit
            // On signale l'utilisation de la table en chargement complet
            (snapshot)
            tableProduit = new SyncTable("Product");
            tableProduit.SyncDirection = SyncDirection.Snapshot;
        }
    }
}
```

```
agentBdd.SyncTables.Add(tableProduit);

// Création du provider côté serveur
// On le lie au SyncAgent préalablement créé
srvCnx = new SqlConnection(cnxServer);
serverBddProvider = new DbServerSyncProvider();
serverBddProvider.Connection = srvCnx;
agentBdd.ServerSyncProvider = serverBddProvider;

// Ajout du SyncAdapter Produit dans lequel on trouve
// la requête de récupération des données produit
adapterProduit = new SyncAdapter("Product");
cmdProduit = new SqlCommand();
cmdProduit.CommandType = CommandType.Text;
cmdProduit.CommandText = "SELECT ProductID, Name, ListPrice
FROM Production.Product";
cmdProduit.Connection = srvCnx;
adapterProduit.SelectIncrementalInsertsCommand = cmdProduit;
serverBddProvider.SyncAdapters.Add(adapterProduit);

// On ajoute le Provider client et sa connexion
// Puis on le lie à l'agent
clientBddProvider = new SqlCeClientSyncProvider(cnxClient, true);
agentBdd.ClientSyncProvider = clientBddProvider;

// Lance la synchronisation
agentBdd.Synchronize();
}
```

Pour changer le type de synchronisation il est possible de passer SynDi-  
rection à Bidirectionnel et de générer le **SyncAdapter** par un **SqlSyncA-  
dapterBuilder** qu'il faudra paramétrer.

## Finalement

Si vous vous lancez dans ce genre de développement, sachez qu'il doit bien être pensé au départ. Particulièrement les formes de synchronisation bidirectionnelles sont difficiles à gérer, et dans OCS toute la logique de gestion des conflits est à écrire. En tout cas, ce système de synchronisation est assez générique et permet de facilement créer des bases de données déconnectées sur les postes clients. On les utilisera en complément des systèmes de mises en cache classiques ou pour le développement d'applications nécessitant ce genre de technique. Il est à espérer que ce service sera disponible assez rapidement sur les plates-formes mobiles et que l'offre de providers devienne plus étoffée qu'actuellement pour profiter au plus grand nombre de développeurs. N'hésitez pas à tester ses fonctionnalités dès aujourd'hui en téléchargeant sa CTP.

■ **Christian ROBERT** - Winwise - Consultant - Formateur  
MVP System Server - SQL Server  
Email : christian.robert@winwise.fr  
Blog : <http://blogs.codes-sources.com/christian/>  
Site : <http://www.winwise.fr>

# INDISPENSABLE!

Hors-Série

**Pro**  
Mai / Juin 2007 - Hors-Série

www.programmez.com

**grammez!**  
LE MAGAZINE DU DÉVELOPPEMENT

## 100% Pratique ! Spécial .Net 3.0

Windows Presentation Foundation, Windows Workflow Foundation, Windows Communication Foundation, Visual Studio Orcas, Ajax

### Web 2.0

- Réalisez votre player MP3 avec WPF / E
- Maîtrisez la 3D dans WPF
- ASP.Net Ajax : Ajax selon Microsoft !

### Workflow Foundation

Intégrez des Workflows dans SharePoint

### Office 2007

Générer des documents OpenXML

### Vista

- ✓ ORCAS : découvrez le prochain Visual Studio !

## LA BOÎTE à OUTILS .NET 3.0

Visual Basic 2005 Express Edition Version gratuite

Microsoft .NET Framework .NET3.0  
Dernière version du package redistribuable, édition 32 bits

Windows Workflow Foundation  
Extensions Visual Studio 2005 pour le framework .net 3.0

Infragistics  
NetAdvantage for WPF 2007 beta 1  
Composants pour vos développements WPF

DEVELOPPEMENT WEB  
WPF / E CTP et son SDK (Février)  
Dernière version de Windows Presentation Foundation / Everywhere

Windows VISTA  
Microsoft Application Compatibility Toolkit 5.0  
Pour vérifier la compatibilité des applications

ASP.NET AJAX  
ASP.Net Ajax Extensions 1.0  
Découvrez l'Ajax de Microsoft.

Windows Upgrade Advisor  
Votre PC est-il compatible avec Vista ?

Code Fluent  
Générer votre code avec une usine à l'intention.

En vente chez  
votre marchand  
de journaux

## Le langage D : votre prochain langage

Sorti dans sa version 1.0 depuis le début du mois de janvier 2007, le langage D est un langage de programmation système orienté objet dont la première mouture date de 1999 et qui, jusqu'à présent, est resté relativement méconnu de la plupart des développeurs. Découverte de ce langage que ses adeptes qualifient de futur successeur du célèbre langage C++.

Le langage D a été créé en 1999 par Walter Bright et sa société Digital Mars. Il est réputé allier la rapidité et la puissance du C/C++ à la simplicité d'écriture du C# et du Java. Walter Bright dit à son propos " Si un langage peut récupérer 90% de la puissance du C++ mais seulement 10% de sa complexité, cela vaut largement le coup". Le D est un langage compilé directement en code natif et qui ne nécessite pas de machine virtuelle pour s'exécuter. Cela en fait un langage parfaitement adapté pour la programmation système et bas niveau. La syntaxe du D étant très proche de celles des langages tels C++, C# et Java, cela facilite d'autant son apprentissage par les développeurs déjà rompus à ces langages. De plus, le D se base sur des bonnes pratiques de programmation (programmation par contrat, tests unitaires...) ce qui permet l'écriture de programmes robustes, fiables et dont la maintenance est facilitée.

### Spécificités particulières

- Le Ramasse-miettes (Garbage collector)

Le ramasse-miettes permet au développeur de ne pas s'occuper de la tâche plutôt ardue de la gestion de la mémoire. Il est activé par défaut, mais peut tout à fait être désactivé si le développeur le souhaite.

- La programmation par contrat (Design-By-Contract)

La programmation par contrat est directement intégrée au D. Cette méthode de programmation a été inventée par Bertrand Meyer, qui l'a introduite dans son langage Eiffel en 1985. Elle se base sur une notion de passage de contrats entre les différents objets d'un programme, permettant ainsi de faire collaborer ces derniers entre eux de façon claire. Les contrats se basent sur les assertions (conditions qui doivent être vraies) qui sont de trois types :

- Les pré-conditions : doivent être vraies avant l'appel de la méthode. Signifie que la méthode va pouvoir être exécutée sans erreur.
- Les post-conditions : doivent être vraies après l'appel de la méthode. Signifie que la méthode s'est exécutée sans erreur.
- Les invariants : doivent être vraies tout au long de la vie de l'objet.

La programmation par contrat est généralement couplée aux tests unitaires qui, eux aussi, sont directement intégrés au D.

- Les tests unitaires

Concept généralisé par l'Extreme programming (méthode agile de gestion de projet informatique), les tests unitaires permettent de tester un petit bout de programme (généralement une classe) de façon automatique. Le test est écrit pendant ou avant la partie de programme qu'il teste. Il permet au cours du développement de cette dernière, d'avoir tout le temps un code sans bugs car celui-ci doit toujours passer le test avec succès.

- Les templates

Le D supporte les fonctions et les classes templates. Les templates permettent l'écriture de code générique réutilisable et adaptable à

tout type de variable. Ils favorisent donc la réutilisation du code qui permet de réduire le nombre de lignes de code avec tous les avantages que cela entraîne : diminution des temps de développement, réduction des bugs et code moins complexe et donc plus facile de maintenance.

- Les délégués

Les délégués du langage D agissent comme les pointeurs sur fonctions du C et du C++ mais ils sont plus simples à mettre en œuvre que ces derniers. Ils sont composés d'une référence à un objet et d'un pointeur sur une fonction. De plus, l'utilisation des pointeurs, bien que possible en D, n'est pas recommandée.

- Les tableaux dynamiques

La taille d'un tableau dynamique n'est pas fixe et peut changer en fonction des besoins au cours de l'exécution du programme.

- Les tableaux associatifs

Ce sont des tableaux basés sur une combinaison clé/valeur. La clé est codée en type.

- Les modules

Un module correspond à un fichier source. Plusieurs modules peuvent être regroupés dans un package et utilisés via un import dans un fichier source.

Exemple tiré de la section du D du site de Digital Mars :

```
import std.stdio;
import foo, bar;

void main()
{
    writeln("hello\n");
}
```

1ère ligne : import du module stdio contenu dans le package std.

2ème ligne : import des modules foo et bar.

Dans la fonction main : appel de la fonction writeln contenue dans stdio, le véritable appel est std.stdio.writeln.

- Interfaçage avec des bibliothèques écrites en C

On peut très facilement, au sein d'un programme écrit en D, accéder à des bibliothèques écrites en langage C. Cela permet l'utilisation de bibliothèques existantes sans avoir nécessairement à les traduire en D. Vous trouverez une liste plus complète de ses particularités à cette adresse : <http://www.digitalmars.com/d/overview.html>.

### Comparaison avec d'autres langages

Il existe de nombreuses différences entre D et les autres langages. Nous pouvons retenir :

- C++
  - La syntaxe du D est moins complexe
  - Le D ne supporte pas l'héritage multiple
- C#
  - Le D est compilé directement en code binaire et non en bytecode
  - Il ne nécessite pas de machine virtuelle pour s'exécuter
  - Il n'est pas strictement orienté objet et on peut tout à fait développer de façon procédurale comme en C
- Java
  - Mêmes différences qu'avec le C# mais avec en plus :
    - En D, l'utilisation des pointeurs est possible (mais déconseillée)

Tableaux de comparaison plus complets entre le D et un plus grand nombre de langages : <http://www.prowiki.org/wiki4d/wiki.cgi?LangagesVersusD> et <http://www.digitalmars.com/d/comparison.html>

### Les différents compilateurs disponibles

Il existe à l'heure actuelle deux compilateurs pour le langage D :

- DigitalMars D (DMD) : compilateur de Digital Mars pour Windows et Linux. <http://www.digitalmars.com/d/dcompiler.html>
- GDC (disponible sur [sourceforge.net](http://sourceforge.net)) : compilateur libre basé sur GCC. Il se décline également en une version pour Windows gdcwin et une pour Mac gdcmac. <http://dgcc.sourceforge.net/>

### Installation et utilisation du compilateur DMD

Ici, nous allons installer la version pour Windows. Il va donc falloir télécharger, dans la partie "Win 32 D Compiler", dmd.zip (D compiler) et dmc.zip (linker and utilities) en les plaçant à la racine (C:\) et les décompresser. dmd.zip va créer un répertoire dmd et dmc.zip va créer un répertoire dm.

Ensuite, il va falloir rajouter le chemin vers l'installation de DMD dans notre variable d'environnement PATH. Pour cela, il faut faire un clic droit sur "Poste de travail" (soit sur le bureau, soit dans le menu démarrer) et sélectionner Propriétés. La fenêtre "Propriétés système" s'affiche. Dans celle-ci, il faut aller sur l'onglet *Avancé* puis cliquer sur le bouton *Variables d'environnement* en bas de la fenêtre. La fenêtre suivante apparaît : Fig A.

Il faut y sélectionner la ligne Path dans la deuxième liste, puis cliquer sur Modifier. La fenêtre de modification de la variable d'environnement s'affiche : Fig B.

On rajoute, à la suite des autres chemins, les chemins vers les répertoires bin du compilateur et du linker (ici respectivement C:\dmd\bin et C:\dm\bin) et enfin on clique trois fois sur OK.

Tout est prêt et vous pouvez désormais commencer à développer et exécuter des programmes en langage D.

### Quelques exemples

Il existe quelques IDE permettant de développer en D :

- Elephant : <http://www.dsource.org/projects/elephant/>
- Poseidon : <http://www.dsource.org/projects/poseidon>
- LEDS : <http://www.dsource.org/projects/leds/>

Toutefois, nous allons réaliser tous nos exemples à l'aide du bloc note (ou tout autre éditeur de texte) et en ligne de commande.

Le traditionnel Hello World!

Nous allons créer un fichier HelloWorld.d contenant le code suivant :

```
import std.stdio; //librairie contenant writeln
```



Fig A.

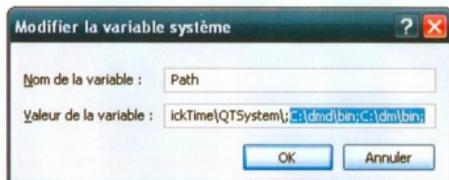


Fig B.

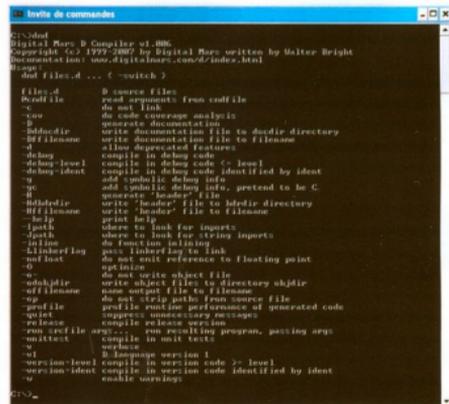


Fig C.

# Gros Plan

```
int main(char[][] args)
{
    writeln("Hello World!"); //affiche Hello World! sur la console

    return 0;
}
```

Afin de le compiler et de l'exécuter, on va ouvrir une Invite de commande Windows.

La commande `dmd` va nous permettre de compiler notre programme. On peut afficher toutes les options disponibles pour cette commande en tapant juste `dmd` sur la ligne de commandes : Fig C.

On va, tout d'abord, compiler notre programme `HelloWorld` en tapant `dmd HelloWorld.d` sur la ligne de commandes.

On peut remarquer que la compilation a créé les fichiers suivants :

- `HelloWorld.exe` : exécutable du programme
  - `HelloWorld.map` : fichier auxiliaire
  - `HelloWorld.obj` : version compilée mais non "liée" du programme
- Ensuite, on peut lancer l'exécutable généré en tapant son nom : `HelloWorld`.



```

C:\> dmd HelloWorld.d
C:\> HelloWorld.exe
Hello World!
C:\>
```

## • Tableau associatif

```
import std.stdio;

int main(char[][] args)
{
    //déclaration d'un tableau de type entier et dont le type de la cle est
    un tableau de caracteres
    int[char[]] tab;

    //initialisation du tableau
    tab["test"] = 2;

    //affichage de la valeur correspondant a la cle test
    writeln(tab["test"]);

    return 0;
}
```

## • Délégué

```
import std.stdio;

/* déclaration du delegue qui prend en parametre un tableau de caracteres
 * et retourne un tableau de caracteres
 */
char[] delegate(char[]) monDelegue;

class Test {
    char[] maMethode(char[] maChaine) {
        return maChaine;
    }
}

int main(char[][] args)
{
    Test test = new Test();

    //monDelegue est un delegue sur l'objet test et la methode maMethode
    monDelegue = &test.maMethode;

    //appel de la methode par le biais du delegue
    writeln(monDelegue("Hello"));

    return 0;
}
```

## Conclusion

Le langage D semble présenter beaucoup d'avantages pour le développement d'applications système fiables et performantes. Alors, est-il le futur langage "star" de la programmation système ou deviendra-t-il une énième "curiosité" présentée sur des sites tel "99 bottles of beer"? Seul l'avenir nous le dira...

## Liens utiles

- Les spécifications du langage D sur le site de Digital Mars : <http://www.digitalmars.com/d/index.html>
- Des sites et tutoriels pour bien commencer à développer en D : <http://www.quit-clan.de/docwiki/view.php?pageid=1>  
[http://spottedtiger.tripod.com/D\\_Language/D\\_Main\\_XP.html](http://spottedtiger.tripod.com/D_Language/D_Main_XP.html)  
<http://www.dsource.org/projects/tutorials/wiki/>  
<http://www.prowiki.org/wiki4d/wiki.cgi?FrontPage>

■ Nathalie Muderhwa - [juwell@supinfo.com](mailto:juwell@supinfo.com)

Tous les jours : l'actu et le téléchargement

[www.programmez.com](http://www.programmez.com)

## La qualité à votre portée



### Serveurs Dédiés **IBM** x Séries

IDC avec toutes les garanties de sécurité

Support technique gratuit par mail et téléphone

Garantie 30 jours Satisfait ou Remboursé

Adresses Ips illimitées

Devis en ligne et sur mesure

...

Découvrez tous les avantages de travailler avec la meilleure marque. Arsys, hébergeur professionnel depuis plus de 10 ans, vous offre votre propre serveur IBM ou Supermicro avec toute la technologie, sécurité et qualité de service à partir de **99 €/mois**.

Découvrez avec [arsys.fr](http://arsys.fr) l'internet de qualité.

**arsys.fr**  
internet de qualité

Noms de Domaine

Hébergement

Serveurs Dédiés

Applications

Dédié Générique  
Dédié Administré  
Dédié de Courrier

[www.arsys.fr](http://www.arsys.fr) / 0800 940 865

Appel Gratuit

## Générer automatiquement votre code !

La génération automatique de code fait partie du quotidien de nombreux développeurs. Et cela concerne quasiment toute l'application qu'elle soit en Java, .Net et même C++. Nous verrons dans ce dossier que la définition, l'approche de la génération automatique varient selon les éditeurs, même si les techniques se ressemblent peu ou prou. Il faut savoir bien définir son modèle et être conscient des limites de ses outils.

De nombreuses techniques existent. La plus commune, tout du moins, la plus utilisée est la génération à partir de modèles. Cela concerne les outils UML, MDA ou encore MDD. Cependant, il existe de nombreux outils capables de générer du code notamment pour la couche d'accès aux données.

### Entre problèmes et marché

La génération automatique, un choix pertinent ? " Je le pense. Avec le modèle, on a une bonne abstraction ", commente Bruno de Combiens (Borland France). Cela fournit de la souplesse et une agilité certaine. Cependant, il ne faut pas non plus attendre de miracle. Des problèmes existent avec cette technique de codage. " Le problème est la transformation, ainsi que le changement de formalisme ", poursuit Bruno. La synchronisation entre le modèle et le code cause des soucis en cas de modifications ou dans les projets lourds (plus le modèle ou le code devient complexe, plus il y a des risques d'un mauvais fonctionnement du bi-directionnel). L'autre souci des outils "généralistes", notamment en approche MDA/MDD, est la prise en compte des spécificités des langages, voire des plates-formes. Mais alors, peut-on générer une application dans son entier ? Non, cela n'est pas possible avec les modèles actuels de générateurs. Mais on peut atteindre 50 à 80 % de code, selon l'abstraction des modèles et surtout la capacité des générateurs. " On peut faire peu ou beaucoup. Jusqu'où peut-on aller ? Il n'y a pas de réelles limites ". Dans tous les cas, le développeur reste incontournable pour pouvoir vérifier, certifier et surtout intégrer le code et réaliser la glue entre les différents éléments. " Il y a les développeurs encore très collés au code et ceux qui tendent à s'en détacher. C'est une pratique qui va s'étendre. Mais on risque de voir apparaître des outils de modélisation, spécialisés " conclut Bruno de Combiens.



### Plusieurs types de générateurs ?

Christophe Ingelaere (Effect Software, éditeur d'ObjectAlive), distingue plusieurs types d'outils et techniques, pour le code source :

- Modélisation : Généralement à partir d'un langage de modélisation tel qu'UML pour le plus connu. On génère ici, par exemple, du code C# à partir des modèles statiques (diagramme de classes). La partie dynamique n'est pas à ce jour correctement prise en charge pour des raisons de complexité, et est sujet à polémique due notamment à la difficulté de modélisation du code à partir d'un diagramme de séquence. Les meilleurs outils permettent une mise à jour transparente du code et du modèle dans les deux sens (round trip engineering = génération + reverse). On

peut citer les produits Rational Rose, Rational XDE, Borland Together, ..., et dans une (bien) moindre mesure le Class Designer de VS Studio 2005.

- Génération de code standard : ici, IntelliSense est un bon exemple de génération de code "simple" permettant une meilleure productivité (implémentation d'une interface, d'une classe de base, ...).
- Refactoring : un autre exemple de génération de code très appréciable dans l'évolution de l'application (création des accesseurs, extraction d'interface, ...).

La situation change avec la génération de code IL (langage intermédiaire). Cette technique est utilisée par les développeurs afin de répondre à une problématique de performance ou de génération de code venant compléter les fonctionnalités de

base d'un assemblage. L'utilisation d'Emit semble ici indiquée. On retrouve aussi :

- ajout de code personnalisé : il est parfois nécessaire d'intercepter un traitement à l'exécution et d'y ajouter des actions répondant par exemple à une problématique de trace ou de gestion des erreurs. Le trace de l'exécution d'une application impose souvent un développement intrusif dans le code métier. La génération du code de trace, qui reste une problématique purement technique, libère le développeur et permet d'obtenir un code plus lisible. Le tissage (AOP) est un principe employant cette technique. On peut noter également qu'il est tout à fait possible d'ajouter du code au moment de la compilation. Dans ce cas, on pourra très précisément ajouter, modifier ou soustraire des instructions dans un assemblage à l'aide d'ILASM et ILDASM. Cette technique, très pragmatique, permet de s'affranchir d'une part d'une certaine complexité, d'autre part des limites de l'utilisation d'Emit, et également dans une certaine mesure des difficultés de débogage et de maintenance du code généré à l'exécution. L'outil Dotfuscator est un bon exemple d'utilisation de cette technique.

- Améliorer les performances : les performances de certains traitements génériques peuvent être améliorées en générant du code spécialisé, par exemple sur des calculs mathématiques coûteux en temps processeur. On pourra alors générer un code très optimisé tenant compte du contexte (valeurs des paramètres, choix de la formule la plus adaptée, ...).

- Générer du code redondant : là aussi pour gagner en temps de développement.

- Autres cas spécifiques: créer son propre langage, un compilateur, un interpréteur, ...

Bref, selon les besoins, on n'emploiera pas les mêmes outils, les mêmes approches.

■ François Tonic



## Java et la génération automatique

" Il existe de nombreuses façons de générer du code, certaines peu recommandables car elles s'apparentent trop à du copier/coller, source de trop nombreux bugs, d'autres beaucoup plus maîtrisées. L'usage le plus courant est probablement celui consistant à utiliser des assistants pour créer des squelettes de code. Ces squelettes sont habituellement configurables mais le problème principal réside dans le fait que toute nouvelle génération de code est à sens unique et qu'elle écrase les modifications d'une génération précédente. Cela reste dans le monde Java l'approche la plus pragmatique et probablement la plus utilisée. A titre d'exemple, les fonctionnalités "Entity Classes from Database" et "JSF Pages from Entity Classes" de NetBeans 5.5 qui proposent en deux étapes de créer une application complète de manipulation (création, modification, énumération, suppression) d'enregistrements d'une base relationnelle. Ce qui est généré est standard (EJB 3 et pages JSF) et modifiable. L'autre exemple classique est celui de la génération de code à partir de diagrammes UML, surtout de diagrammes de classes. Les notions de retro-ingénierie (reverse-engineering) restent dans ce cas explicites et ne se préoccupent que de la structure des classes, pas de leur comportement. Une autre approche dans la génération de code consiste à passer par un format pivot manipulé à la fois par l'éditeur de code et des outils visuel. C'est l'approche retenue dans NetBeans Visual Web

Pack (ex-Java Studio Creator) qui propose des éditeurs visuels pour les requêtes SQL, la conception WYSIWYG de pages JSF ou encore les règles de navigation entre pages web. Ce format existe en mémoire et permet à tout moment de répercuter le changement effectué par l'éditeur graphique dans le code source (Java, SQL ou XML) et vice versa. Si on pousse le raisonnement un peu plus loin, on se dit que l'on peut utiliser un format qui existe déjà: l'AST (Abstract Syntax Tree) tel qu'il est manipulé par le compilateur (Javac en l'occurrence). Il existe un standard (JSR 199, "Java Compiler API") qui expose ce format et Jackpot qui se base dessus pour scripter des opérations de refactoring. Cette technologie qui est à la base de NetBeans 6.0 et permet d'obtenir du code généré qui compile toujours, puisqu'il est issu du compilateur lui-même. Jackpot peut-être utilisé à l'échelle d'un projet complet pour appliquer des transformations de code qui visent à modifier l'usage d'API: passer d'une API de log à une autre, rajouter de la généricité sur des collections, passer d'EJB 2 à EJB 3, etc... Le JSR 199 et Jackpot sont des technologies récentes dont on commence à peine à utiliser les capacités. Elles se positionnent comme des outils de manipulation/génération de code sans pour autant être des boîtes noires, ce qui est le reproche le plus courant fait aux outils de génération de code à partir de règles, quel que soit le formalisme retenu. "

■ Alexis Moussine-Pouchkine - Java Middleware Evangelism Team - Sun

## Questions à Grégory Dufour (développeur et créateur de Denki-SQL)

**Programmez ! : vous êtes l'auteur de Denki-SQL, quelle a été la problématique de départ ?**

**Grégory Dufour :** La plus grosse problématique de Denki-SQL fut sans aucun doute la possibilité aux classes générées d'être mises à jour automatiquement pour suivre l'évolution du schéma de la base (ajout/suppression/modifications des tables/

colonnes/rerelations...). Le processus de Mise à jour se charge de comparer la base actuelle avec le scénario XML, trouve les différences et fait le nécessaire. J'aime dire que Denki-SQL est un " manager de code " plutôt qu'un " générateur de code ".

**PI ! : sur quelles techniques vous appuyez-vous pour faire cette génération automatique ?**

**Grégory Dufour :** Denki-SQL n'utilise aucune technique particulière, ne serait-ce que la prise en charge du multithread, indispensable. Les objets métiers et les procédures stockées générées sont les résultats de classes analysées en T-SQL. Il n'y a ni SMO, ni REFLEXION.

■ FT

## Génération automatique d'applications selon les concepts MDA.

*A partir de quelle source génère-t-on ? Quel gain réel obtient-on ? Est-ce plus compliqué ? Est-ce une solution viable ? Faut-il encore coder ? Voici quelques unes des questions que l'on se pose couramment face à la génération automatique d'applications. (1)*

Aujourd'hui, on peut affirmer que la génération d'applications passe par le MDA (Model Driven Architecture). Il est basé sur la transformation de modèle en code et orbite autour du langage UML. La raison principale du choix de l'UML par rapport à d'autres langages de modélisation est certainement la myriade d'outils disponibles tels que Together, Enterprise Architect de SPARX-SYSTEM, Magic Draw de NO MAGIC, etc., pour un investissement financier relativement faible. Cependant, dans le cas qui nous préoccupe, le point clé est le standard de stockage. A vrai dire, il en existe deux, l'EMF (Eclipse Modeling Framework) et l'XMI (XML Meta Interchange). Pour avoir exploré les deux, celui qui ouvre les plus belles perspectives pour la génération automatique d'applications est l'XMI. Standard de l'OMG, il a été créé pour échanger des informations méta-data, via l'XML (eXtensible Markup Language). Il peut être utilisé pour n'importe quelle méta data dont les méta modèles sont exprimés en Meta Object Facility (MOF). C'est grâce à l'XMI que la génération automatique d'applications devient possible. Il est nécessaire de comprendre les notions de MOF, méta classe, méta modèle, et le principe de transformation basé sur le PIM : Plateforme Indépendant Models en PSM : Platform Specific Models.

### Le PIM.

Le PIM représente des modèles indépendants de la plate-forme et de la technologie. En résumé, les modèles UML représentent le fonctionnel à générer. A la génération et selon les choix du concepteur, le PIM est transformé en PSM pour finalement produire le code déployable. La transformation peut se faire en plusieurs étapes et suivant diverses tech-

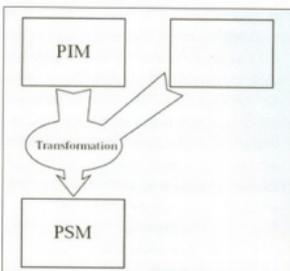
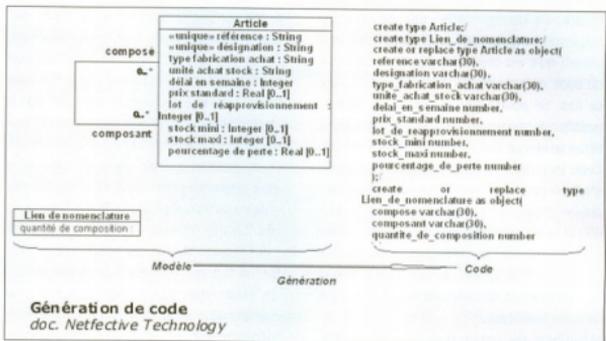


Figure 1 : (Source OMG) Principe de transformation PIM → PSM

niques. Ci-dessus le pattern issu de l'OMG, préconisant un enrichissement automatique ou manuel du PIM afin de le transformer en PSM. (Figure 1)

### Le PSM

Le PSM est un modèle fortement typé en fonction de la plate-forme cible (code exécutable +

architecture + plate-forme d'exécution). Dans les implémentations des "moteurs" dits MDA on ne voit jamais le PSM ; il n'est présent que temporairement au moment de la génération. L'archétype de transformation le plus classique de PIM vers PSM est le "mapping" des données présentes dans le PIM vers des données spécifiques à la technologie générée. Par exemple, je modélise l'attribut "nom" d'une classe "Client" avec un type "string", il se transforme en "java.lang.String" si je génère en JAVA ou bien en "System.String" pour une génération .NET.

### Le méta-modèle et le MOF

Les méta-classes issues du MOF, représentent les objets structurant l'UML. C'est à partir de ces méta-classes que l'on peut définir un méta-modèle. Ce dernier, non seulement est capable d'interpréter un PIM, mais il le valide aussi. Les éditeurs de logiciels de modélisation commencent à intégrer cette notion de méta-modèle comme un composant de leur DSL (Domain Specific Language). Dans la der-

(1) par génération automatique d'applications, on entend génération automatique d'un EAR (Java EE) prêt à être déployé sans intervention de ressources de type développement, par opposition à génération automatique de code qui généralement nécessite un effort d'intégration et de tests conséquent avant constitution d'un EAR prêt à être déployé.

nière version 12.1 de Magic Draw, on peut coder un méta-modèle en s'appuyant sur des contraintes OCL et du code JAVA, ceci à pour avantage de pouvoir créer son propre type de modélisation, et aussi, de pouvoir le valider directement dans l'outil de conception et non pas pendant la génération. Il est évident que le générateur doit interpréter le méta modèle de la même façon. Dans Blu Age, le méta modèle pré codé, définit un sous ensemble de l'UML 2.1, et permet la transformation du PIM vers le PSM. D'un point de vue plus pragmatique cette action s'effectue au travers un principe de méta-façade : " système de wrapper(2) pour des méta-classes ", qui une fois les modèles chargés, expose les données contenues dans les PIMs comme un contexte de génération pour les " Template Velocity ".

(2) dans ce contexte : *conteneurs intelligents pouvant se contenir eux-mêmes*

### Codes des " templates " et les quatre niveaux de transformations de M0 à M3.

Prenons par exemple l'action click sur un bouton dans une application, c'est le niveau M0 : " le code exécutable ". Dans le modèle, niveau M1 : " modèle ", cette action est modélisée par une " transition " dans un diagramme d'activité entre deux activités. Au niveau M2 : "méta-modèle" on retrouve l'encapsulation de la méta-classe " transition " dans un "wrapper" appelé méta-façade. Et finalement le niveau M3 : " méta-méta-modèle, le MOF ", la définition de la " transition " dans le MOF. Le passage du niveau M1 au niveau M0, s'effectue en mettant en relation la méta-façade avec le bon " template " de génération. Par exemple, si l'on veut générer des classes contrôleur à partir de l'élément UML " transition ", on mettra en relation la " template " avec la méta-façade " ActivityFacade " contenant des " TransitionFacade ". De la même manière que l'on navigue aux travers des requêtes dans une base de données, on exploite les relations entre méta-objets pour requêter sur le modèle directement à l'intérieur des " templates Velocity ". (Fig. 2 - 3)

Les exemples au-dessus représentent la relation entre une méta-façade et son " template " de génération. Voici le code que cela générerait : (Fig. 4)

Dans l'Open API de Blu Age on retrouve plus de 160 méta-façades contenant environ 1200 opérations. Pour vous aider à obtenir l'informa-

```
<template location="Controleur.vsl" outFile="(0)/Controleur(1).java"
outdir="controleurDir">
<metafacades>
<metafacade
entryPoints="com.bluage.meta.model.uml2.behavioural.ActivityFacade"
variable="activity"/>
</metafacade>
</template>
```

Figure 2 : Définition d'une « template » et mapping avec une méta façade  
*Doc.Netfective Technology*

```
public class Controleur{activity.name}
{
foreach(faction in {activity}.transition)
public String click_{f.action.name}()
{
return invokeService("{faction.callOperation.serviceCall"});
}
}
#end
}
```

Figure 3 : Codage d'une « template Velocity » contrôleur.  
*Doc.Netfective Technology*

```
public class ControleurAScreen
{
public String click_btn_Valider()
{
return invokeService("valideFacture");
}
public String click_btn_RetourClient()
{
return invokeService("ReturnClient");
}
}
}
```

Figure 4 : code généré : ControleurAScreen.java  
*Doc.Netfective Technology*

- o M3 méta-méta-modèle - Le MOF
- o M2 méta-modèle - le méta-modèle UML
- o M1 modèle - des modèles UML et d'autres
- o M0 monde réel - le code exécutable

Figure 5 : (figure OMC) Transformation à quatre niveaux  
*Doc.Netfective Technology*

tion exacte et pour faciliter le " mapping " d'un " template " à une méta-façade, il existe une série de plug-in Eclipse aidant le concepteur à choisir la bonne méta-façade.

### L'approche génération par cartouche

Les cartouches de génération sont le cœur de la transformation des modèles PIM en PSM, elles fonctionnent comme des plug-in et sont interchangeables. C'est au niveau des cartouches que l'on retrouve les liaisons entre le méta-modèle et les " templates " de générations.

Tout se passe dans la configuration des cartouches. Dans un premier temps, il est nécessaire de connaître l'environnement cible. Tout d'abord le choix de la plate-forme, Java EE ou .NET. Puis viennent les choix d'architectures JSF / Struts / Spring / EJB / Hibernate. Une fois l'architecture choisie, il faut configurer l'environnement d'exécution : WebSphere 6.0,

WebLogic, IIS, etc. Chaque cartouche contient sa propre configuration. Par exemple, dans la cartouche Hibernate on configure le type de base de données SQL Server, Oracle, DB2, etc. Puis l'URL de connexion à la base de données, les librairies gérant la connexion, le mode transactionnel, la gestion du cache... Pour aider l'architecte en charge du paramétrage, un système de " template " de configuration permet de configurer une seule fois le moteur pour un type d'application.

Une fois le paramétrage effectué, on lance une itération de génération, comme flux d'entrée on passe le ou les modèles UML, une maquette IHM XHTML enrichie de tags Blu Age, le code spécifique s'il y a lieu. Puis le moteur instancie les cartouches, charge les modèles XML dans son " repository " (assimilable à une base de données pour modèles UML au format XML). Les " templates velocity " contenues dans les cartouches s'exécutent. Une fois le code généré, il est repris par un packageur (Ant ou Nant en l'occurrence), qui compile et produit soit un EAR ou soit un MSI. L'archive peut être ensuite récupérée puis déployée.

### L'UML est-il assez complet pour modéliser une application ?

Dés lors une question se pose : " L'UML est-il assez complet pour modéliser une application

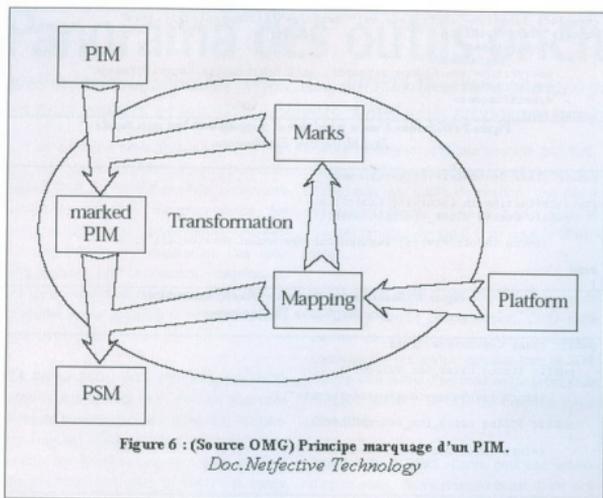


Figure 6 : (Source OMG) Principe marquage d'un PIM.  
Doc.Netfective Technology

de A à Z en vue d'une génération sans intervention de développeurs ? " Ma réponse est normative oui et non. Oui dans la mesure où d'un point de vue théorique il permet de représenter à peu près n'importe quel système complexe, mais non si on se place dans un contexte de génération automatisée d'applications et dans la mesure où chacun des modèles doit être enrichi et où la modélisation des IHM reste quasiment impossible. L'UML s'attaque à la modélisation des systèmes ; pratiquement on modélise pour simplifier un système ou la génération automatisée nécessite un référentiel de modèle à générer complet et non implicite. Partant de cette contrainte il est nécessaire d'enrichir les modèles UML et XHTML, selon les concepts préconisés par l'OMG : " le marquage d'un PIM ".

Par exemple dans Blu Age cet enrichissement se concrétise au niveau de plusieurs actions :

- Le marquage par stéréotypes et valeurs balisées,

- L'utilisation d'OCL pour raffiner des processus métier,
- L'utilisation directe d'HQL,
- L'utilisation de " binary code " (dans notre cas code JAVA ou .NET),
- L'utilisation d'une maquette enrichie, représentant les interfaces utilisateurs.

(Fig. 6)

### Une autre méthode d'enrichissement : la factorisation de processus métiers

Dans le cas où certains processus sont complexes à modéliser, ou bien répétitifs, La factorisation des processus métiers au travers de profils de modélisations s'avère être une technique productive. Par création de profils UML on entend création de nouveaux stéréotypes, création de nouvelles valeurs balisées et création d'interfaces / classes abstraites pouvant être réutilisés tout au long de la modélisation. Cette action implique le codage de nouvelles cartouches.

### Réaliser une application à partir de modèle UML : une approche 100% conception qui change les métiers du développement.

Les modèles UML utilisés par Blu Age, sont considérés comme des modèles exécutables. Partant de ce principe, le travail des architectes est plus précis, celui des concepteurs plus focalisé sur des parties complexes de l'application. La façon de concevoir en équipe change, on passe d'un mode usant des modèles UML comme des figures graphiques dans des documents techniques à des modèles aboutis, source pour la génération d'une application. La gestion du travail de conception en mode collaboratif est primordiale, et il est nécessaire de s'équiper d'outils pouvant supporter effectivement ce type de fonction. Le niveau des responsabilités et des compétences peut être aussi réparti. Domaine métier, services, processus utilisateurs font appel à des niveaux de compétences différents. Avec le MDA et une fois maîtrisé un outil comme Blu Age, c'est aussi la façon de concevoir et de développer qui change. Tout se passe dans la phase de conception et non dans la réalisation. Ceci apporte un gain au niveau de la rapidité du développement d'application, une plus grande implication des utilisateurs, car ils ont la possibilité de tester l'application tout au long de sa construction. Une grande souplesse au niveau du changement et évolutions des environnements techniques, exemple pour passer d'un WebSphere 5.1.2 à un WebLogic 9.2 ou à .Net, il suffit de changer les paramètres de génération. Au total, on estime entre 40% et 50% de gain de temps par rapport au développement classique. Mais surtout, ceci permet de garder la maîtrise du métier de l'entreprise en interne, de recenser les ressources sur le métier de la conception et minimise les recours aux ressources techniques, pour proposer des structures de réalisation alliant uniquement concepteur, architectes techniques et chefs de projet.

■ Jean Neveux

A.S. Computer Science - Directeur R&D Blu Age  
www.bluage.com

Prochain numéro : Juin 2007 N°98 - Juin - Parution : 31 mai 2007

## Dossier SGBD

Choisir sa base de données.  
Quel SGBD, pour quelle application.  
Les nouvelles fonctions.

## XUL

Développez avec un langage surpuissant facilitant la création d'interface graphique dynamique et interactive.



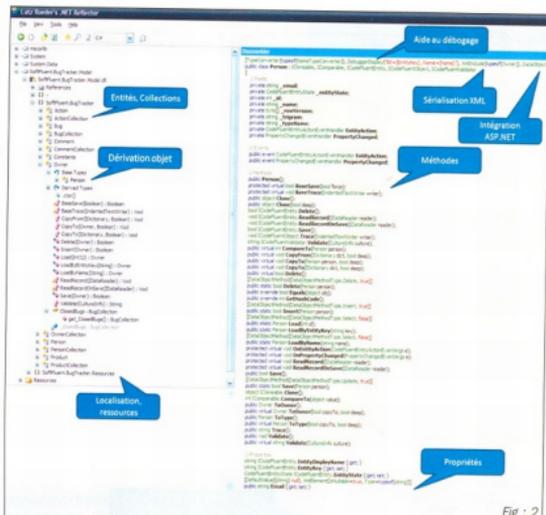


Fig : 2

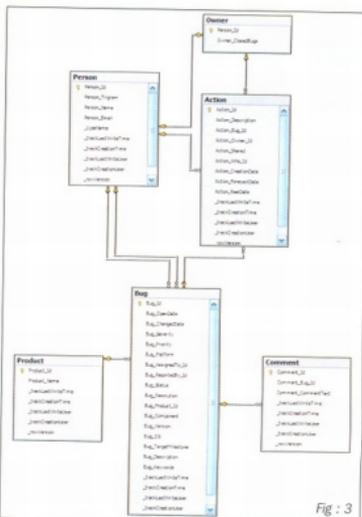


Fig : 3

et nous introduisons donc les entités : Bug, Action, Product, Comment, Person... Au niveau de ces entités, nous décrivons toutes les propriétés dont nous aurons besoin avec leur type, sachant que le produit déduit automatiquement certains types à l'aide de règles par défaut (identifiant en première position, nom se terminant par date, etc.). Certaines propriétés expriment également les relations entre les entités ainsi que leur cardinalité. Par exemple, on associe ici une action à un bogue, mais un bogue peut avoir une liste d'actions. De nombreuses méthodes de navigation dans le modèle seront déduites automatiquement en fonction de ces relations, mais on peut décrire des méthodes spécifiques au moyen de différentes techniques : langage simplifié de requête CFQL, inclusion directe possible de Snippets de code...

```

set ANSI_NULLS ON
set QUOTED_IDENTIFIER OFF
GO
ALTER PROCEDURE (dbo).[Bug_LoadOwnerClosedBugs]
(
    @ownerId [int],
    @orderBy0 [nvarchar] (64) = NULL,
    @orderByDirection0 [bit] = 0
)
AS
SET NOCOUNT ON
SELECT [Bug].[Bug_Id], [Bug].[Bug_OpenDate], [Bug].[Bug_ChangedDate], [Bug].[Bug_Severity], [Bug].[Bug_
FROM [Bug]
LEFT OUTER JOIN [Action] ON ([Bug].[Bug_Id] = [Action].[Action_Bug_Id]
WHERE ([Action].[Action_Owner_Id] = @ownerId AND ([Bug].[Bug_Status] LIKE 'Closed'))
ORDER BY CASE
WHEN @orderBy0 = '[Bug].[OpenDate]' AND @orderByDirection0 = 0 THEN [Bug].[Bug_OpenDate]
END ASC,
CASE
WHEN @orderBy0 = '[Bug].[Status]' AND @orderByDirection0 = 0 THEN [Bug].[Bug_Status]
END ASC,
CASE
WHEN @orderBy0 = '[Bug].[OpenDate]' AND @orderByDirection0 = 1 THEN [Bug].[Bug_OpenDate]
END DESC,
CASE
WHEN @orderBy0 = '[Bug].[Status]' AND @orderByDirection0 = 1 THEN [Bug].[Bug_Status]
END DESC
RETURN
    
```

Fig : 4

La puissance d'une approche de génération de code se traduit par le fait que ces simples éléments suffisent à construire de très gros morceaux de l'application réelle : base de données incluant les procédures stockées de manipulation, objets métiers, services web et composants d'interface utilisateur.

Pour cela, on utilise une notion de producteur, qui indique les éléments que l'on veut produire ainsi que les paramètres de génération (répertoire où ranger les fichiers, compilation automatique, etc.). En utilisant la configuration de

base des producteurs, vous obtenez une génération complète de ces éléments. La figure 2 montre les différentes classes fabriquées au niveau du tiers "métier", ainsi que la structure de la classe Person générée avec ses propriétés et méthodes.

Sur la figure 3, on retrouve le schéma de base de données déduit par l'outil à partir du fichier modèle.

La Figure 4 montre la procédure stockée qui est générée (ici sous SQL Server) à partir de la méthode spécifique LoadOwnerClosedBugs décrite dans le modèle.

l'environnement comprend des producteurs très spécifiques optimisés pour .NET dans la mise en œuvre de différentes couches de l'application. Mais il comprend aussi un producteur particulier qui traite de manière générique des "Templates".

Cette notion de Templates se retrouve dans beaucoup d'autres générateurs de code (CodeSmith, MIA, Olymars...), et parfois constitue d'ailleurs la seule technique de génération. Elle était également la technique des pages Web en ASP de Microsoft avant ASP.NET.

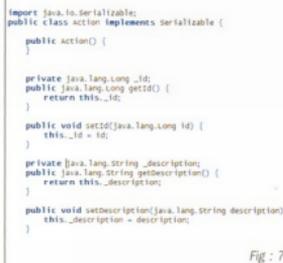


Fig : 7

de passer à une génération complètement automatique de composants réellement opérationnels. Ceci étant, si on utilise UML, il est par contre toujours possible et relativement simple d'effectuer des transformations en amont pour passer d'UML à un modèle Code-Fluent.

Parmi les approches alternatives, certains produits proposent également des échanges bidirectionnels entre modèle et code. De notre expérience, cela s'avère en pratique assez complexe à mettre en œuvre et à maîtriser dans le temps. C'est pourquoi, en ce qui nous concerne, nous avons fait le choix de bien séparer le code généré du code spécifique, de manière à pouvoir travailler uniquement dans le sens du modèle vers le code. Cette stratégie reste très maîtrisable et permet de déployer une modification de modèle de manière extrêmement simple et sûre.

## Et après ?

Nous invitons le lecteur intéressé par le sujet à poursuivre en :

- regardant l'offre pléthorique qui existe en matière de purs outils de génération de code : <http://www.codegeneration.net>
- écrivant ses propres modèles pour les générer à l'aide de la version d'évaluation de CodeFluent de l'usine gratuite en ligne sur <http://www.codefluent.com>
- posant des questions sur <http://forums.codefluent.com> ou en nous écrivant directement.



**Daniel COHEN-ZARDI**  
(SoftFluent)



**Simon MOURIER**  
(SoftFluent)

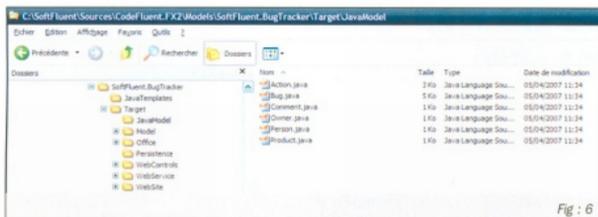


Fig : 6

Cette technique est présente dans l'outil, car elle présente l'avantage de permettre de générer n'importe quel type de fichier de format textuel : code dans un langage quelconque, documentation, scripts d'exploitation.

Le principe du Template est de mélanger au sein d'un même flux :

- Une partie code qui sera exécutée par le générateur et qui est repérée par des balises particulières (ici [% ... %])
- Une partie textuelle qui sera restituée à l'exécution du Template

L'exemple en figure 5 est un Template pour fabriquer des classes Java très simples reprenant les propriétés du modèle. On voit par exemple qu'on effectue une boucle sur les entités du modèle et que l'on restitue le nom de l'entité juste après la déclaration du mot-clé Class.

Notons que si le code produit par ce Template est en Java, le code du Template lui-même est écrit en C#. En fait, l'environnement de génération ne présage en rien de l'environnement cible. Rappelons que nous générons ici des fichiers texte et que ces fichiers pourraient très bien ne pas être du code.

Ce Template produit un fichier classe par entité du modèle (voir Figure 6), chaque classe comprenant les propriétés décrites dans le modèle (exemple en Figure 7)

## Est-ce la seule approche ?

Bien sûr que non, mais cette approche a le mérite d'être très pragmatique. Elle se positionne différemment d'autres produits qui imposent UML pour la modélisation, ce qui ne nous semble pas indispensable, dans la mesure où UML ne permet de toute façon pas

## Le code ou le modèle ?

**MDD : Model Driven Development. Rêve ou réalité ? L'équipe IBM nous donne son point de vue et son expérience dans ce domaine en présentant 5 scénarios d'intégration entre UML et le code, puis les briques techniques à la base de MDD dans Eclipse et enfin son industrialisation dans Rational Software Architect, le vaisseau amiral des outils de développement IBM Rational.**

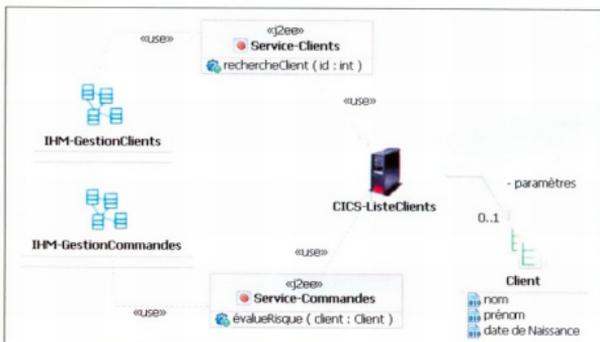
Les équipes de développement du monde entier cherchent sans cesse à améliorer leur processus de développement pour produire mieux et plus vite. Dans ce contexte, la modélisation joue un rôle clef. L'évolution de la technologie et les retours d'expériences mettent en avant 5 scénarios d'intégration possibles entre la modélisation et l'implémentation.

### 1 "Code is King" : Le code est roi

Le premier scénario est typique de la méthode XP : l'équipe ne modélise pas en UML. Le code Java est visualisé graphiquement dans des diagrammes UML. Ils s'insèrent directement dans l'arborescence de packages Java et introspectent les sources au moment de leur affichage. En plus de leur fonction documentaire, ils permettent d'ajouter un attribut, modifier un héritage ... directement dans le diagramme. Le développeur peut "attaquer le code" à partir du source ou de l'éditeur graphique : tous les avantages des diagrammes de classe UML sans surcharge de travail pour gérer un modèle. Cependant, ce mode de développement n'est pas applicable sur des projets à grande échelle car il ne permet pas de gérer efficacement des problèmes métier complexes.

### 2 Génération initiale du code par un modèle UML

UML est largement reconnu aujourd'hui par l'industrie logicielle. De nombreuses organisations ont introduit la modélisation dans leur processus de développement. Elles utilisent UML en amont sans liens directs avec la phase d'implémentation. Le modèle est alors une spécification initiale à la base du travail des développeurs à laquelle on associe souvent une génération initiale de squelettes de code. Cette démarche est assez rigide (modèle puis code) et n'optimise pas la phase de modélisation qui s'arrête rapidement dans le déroulement du projet. Elle n'a donc plus le vent en poupe, même si l'organisation en silo des équipes projet peut parfois la justifier.



Exemple d'utilisation des profils UML. On obtient rapidement des diagrammes clairs et synthétiques.

### 3 Modélisation "mixte" : partage du pouvoir entre modèle et code

Le modèle de développement précédent montre rapidement ses limites : manque de communication entre les différents acteurs, peu de souplesse... Un moyen d'y remédier : le concepteur et le programmeur doivent pouvoir partager la même information sans avoir à la dupliquer. Elle existe alors dans le modèle ou dans le code mais pas dans les 2. Comment est-ce possible ? Les modèles généraux UML (cas d'utilisation ...) n'ont pas d'équivalents directs dans le code. Le problème ne se pose donc pas pour eux. Le concepteur définit ensuite des classes UML. Quand il décide de générer le code Java équivalent, l'outil de développement remplace la classe UML dans le modèle par une visualisation de la classe Java générée. Le fichier java devient la référence. Code et modèle ne pourront pas diverger. Cette démarche a des avantages : pas de réconciliation, modèles UML toujours à jour..., mais aussi des inconvénients : le développeur peut modifier la définition d'une classe, il a un rôle important dans le respect de l'architecture cible. Techniquement, ce scénario novateur n'est supporté que par les dernières généra-

tions d'outils (Rational Software Architect 7.0 par exemple). On a donc assez peu de recul, mais ce scénario est l'évolution logique de la méthode XP pour introduire en amont une phase de modélisation utile pour résoudre les problèmes complexes

### 4 Génération de code et rétroconception

Les générations précédentes d'outils de modélisation avaient introduit la rétroconception. Souvent remplacée aujourd'hui par une visualisation directe du code source beaucoup moins coûteuse, elle reste principalement utilisée quand l'implémentation est sous-traitée et que l'on veut contrôler l'architecture cible. Les 2 images Java et UML évoluent alors séparément et divergent. Leur réconciliation est une tâche lourde et difficile, gérée par le concepteur à la fin d'une itération. Et si on réconciliait automatiquement ? Cette charge disparaîtrait ? Sur le papier, oui. Mais en pratique, la réconciliation automatique tient plus du rêve (ou du cauchemar ;) que de la réalité. L'outil identifie les différences mais il ne peut pas décider à notre place. De plus, les modèles doivent masquer une partie des contraintes d'implémentation pour donner une vision synthétique de

	Qui fait foi ?	Avantages	Limites
Code is King	Code	Mise en œuvre facile et rapide	Convient pour de petits projets Logique métier gérée par le développeur Approche purement technologique. Ne facilite pas la maintenance
Génération initiale par UML	Code	Démarche structurée	Faible intégration entre concepteurs et programmeurs Le code diverge au fil des itérations
Modélisation mixte	Modèle puis code	Diagrammes à jour, Pas de réconciliation	Frontière floue et conflits potentiels entre concepteurs et développeurs
Rétroconception	Modèle et code	Démarche structurée Modèle et code peuvent évoluer librement	Réconciliation difficile, voire impossible Risque élevé de divergence entre modèle et code
MDD	Modèle	Démarche très structurée, Maintenance facilitée Productivité accrue, Cohérence du code	Développement de générateurs Il faut qualifier le retour sur investissement pour chaque générateur

l'application. Sinon, autant regarder directement le code ! Par exemple, il est possible de générer 3 classes Java à partir d'une seule classe UML. Et ces 3 classes Java n'ont pas leur place dans un modèle.

## 5 MDD (Model Driven Development): "Model is King"

MDD rassemble les approches de développement pilotées par les modèles. MDD est souvent présenté comme le prochain Graal de l'industrie informatique. Bien sûr, ce ne sera pas la solution miracle à tous nos problèmes mais les gains sont nombreux : productivité, qualité, maintenance et évolutions facilitées, gouvernance d'applications composites... autant de domaines où les scénarios précédents sont moins performants.

La démarche MDD consiste à mettre en place plusieurs niveaux de modèles (2, 3, 4 ... en fonction du besoin) pour ne pas les polluer avec des considérations trop techniques. Chacun se base sur le précédent et ajoute de nouvelles informations. Il faut automatiser la transition entre les différents niveaux pour maîtriser les coûts de développement. On parle de transformations et on en identifie 2 types : Model2Model, qui génère un modèle à partir d'un autre modèle et Model2Text, qui génère des fichiers (Java, XML, JSP ...) en fin de chaîne à partir d'un modèle.

L'Object Management Group (OMG) a standardisé les principes de MDD. MDA (Model Driven Architecture) introduit différents niveaux de modèles (Computation Independent Model, Platform Independent Model et Platform Specific Model). Mais il n'est pas forcément nécessaire de déployer cette démarche pour tirer parti de MDD. Maintenant que les concepts ont été définis, nous allons rentrer dans le détail de la démarche dirigée par les modèles et présenter les briques techniques nécessaires pour construire la maison MDD. Eclipse, UML, EMF et JET sont les fondations qui permettent de mettre en œuvre MDD sur un projet. Ou comment passer d'un concept prometteur à une réalité.

## Les fondations Eclipse

Est-il nécessaire de présenter Eclipse ? Tous les développeurs Java ont déjà travaillé avec une distribution Eclipse, qu'elle soit gratuite ou commerciale. Eclipse est le socle technique des outils IBM et de la plupart des grands acteurs de l'industrie informatique. Il faut donc étendre ce socle pour ajouter le support de MDD.

### UML et profils

UML est le standard pour modéliser une application informatique. Il trouve tout naturellement sa place dans une démarche MDD. Si la sémantique UML ne vous suffit pas, une solution consiste à affiner cette sémantique en créant un profil UML. Ce profil contient des stéréotypes associés à un type d'entité UML (Classe, Propriété, Opération, ...). Ils peuvent aussi inclure des attributs spécifiques et des règles de validation. Les profils sont faciles à mettre en œuvre et gérés efficacement dans les outils de modélisation actuels, en particulier pour le rendu graphique.

### Domain Specific Language (DSL)

Pour étendre UML, on peut aussi créer un nouveau méta-modèle : un DSL. Il vient compléter UML, qu'on peut d'ailleurs définir comme le "DSL pour la définition générale d'une application". Cette démarche est beaucoup plus ouverte mais nécessite plus de travail de mise en place. Attention à ne pas concurrencer UML : vous n'êtes pas là pour redéfinir les standards existants, mais bien pour les utiliser et les étendre si nécessaire. Enfin, on pourrait

être tenté de définir des DSL pour spécifier totalement une application mais de nombreux travaux de recherche ont clairement établi que le meilleur moyen de définir un algorithme est d'utiliser un langage de programmation. Le code a donc de beaux jours devant lui.

### MOF, ecore et EMF

Un zoo contient des animaux. C'est un modèle. Il s'appuie sur le méta-modèle UML qui définit le concept de classe. Que faire si on souhaite définir un nouveau méta-modèle ? Utiliser un méta-méta-modèle bien sûr ... Ce standard de méta-méta-modélisation est le MOF. Avec lequel on peut définir tous les méta-modèles. UML en est l'exemple le plus évident mais on peut aussi créer d'autres DSLs. L'implémentation de MOF dans Eclipse s'appelle EMF. Issu d'un projet IBM dans la plate-forme WSAD (WebSphere Studio Application Developer), le projet EMF se sépare entre le méta-langage lui-même (ecore) et les outils de génération (API Java et éditeurs). Je suis sûr que certains d'entre vous se demandent fébrilement : et le méta-méta-méta-modèle dans tout ça ? Celui qui permet de spécifier MOF ? Et bien c'est MOF lui-même. Ouf ! L'API Java UML2 est le premier exemple d'utilisation d'EMF dans Eclipse.

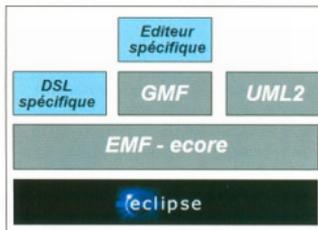
### GMF

Avec EMF, vous pouvez définir de nouveaux méta-modèles. Vous pouvez les modifier et les éditer avec un éditeur arborescent basique. Si

## Comment choisir entre profils UML et DSL ?

Tout d'abord, il faut garder à l'esprit que l'utilisation de profils n'est guère plus standard que la création d'un DSL. Elle introduit une nouvelle sémantique et risque de polluer le modèle UML. Mais elle est simple et bien adaptée quand on veut seulement raffiner le méta-modèle UML. La mise en place d'un nouveau DSL est plus technique et repose sur des frameworks comme EMF dans Eclipse (voir ci-dessous). Au final, il faut souvent mixer les 2 approches en fonction du besoin : profils pour raffiner UML et DSL pour élargir le domaine de la modélisation.

vous avez besoin d'éditeurs graphiques évolués, GMF entre en jeu. Cependant, le développement GMF reste aujourd'hui assez coûteux et nécessite donc de qualifier précisément le besoin pour évaluer le retour sur investissement. On ne peut pas toujours faire joli et on doit parfois se contenter d'un éditeur arborescent "de base". Au final, on obtient donc une fusée à 4 étages pour la gestion des modèles dans Eclipse.



Couches d'implémentation MDD dans Eclipse

## JET

Il nous manque encore une brique de base : la génération de code et plus généralement la génération de fichiers, de répertoires et même de projets Eclipse en entier... C'est la technologie JET. Un template JET mixe contenu statique et dynamique – on peut faire une analogie avec la technologie JSP.

JET permet de spécifier les paramètres des templates dans une classe Java, un fichier XML ou un modèle ecore. Le fichier XML est le plus simple a priori. La classe Java permet d'introduire des algorithmes et d'insérer la génération dans un processus automatique. Enfin, ecore élargit encore le champ des possibilités. Nous souhaitons mettre en place une démarche dirigée par les modèles. Quoi de plus naturel dans ce cas que d'extraire les paramètres dans un modèle ? Les méta-modèles (DSL et UML) s'appuient sur ecore dans Eclipse. Les templates JET peuvent être alimentés à partir d'un modèle ecore. La boucle est bouclée.

## Evolutions futures

La communauté Eclipse est très active dans le domaine MDD/MDA. Des projets intitulés Model-to-Model et Model-to-Text ont été créés récemment. Ils doivent permettre de fournir une infrastructure pour les transitions de méta-modèles. Mais ces travaux sont encore au stade embryonnaire.

## Et QVT dans tout ça ?

MOF QVT (Query View Transformation) est le standard de l'OMG pour spécifier les transitions entre méta-modèles. QVT introduit en particulier un langage pour définir ces transitions. C'est à la fois une force et une faiblesse. On doit apprendre un n-ième langage, on ne dispose pas d'outils performants ni de compétences dans les équipes. En revanche, si les transformateurs sont écrits en Java, nos développeurs peuvent aisément utiliser Eclipse pour les créer. L'avenir de la démarche QVT est donc encore incertain.

## Cas pratique : implémenter MDD avec Rational Software Architect

RSA est l'atelier de développement IBM issu de la fusion de WSAD et des outils de modélisation Rational (Rose et XDE). Sa version 7.0 est basée sur Eclipse 3.2. On peut le résumer avec 1 équation simple : RSA = Java EE + UML + MDD. Les briques techniques MDD sont présentes dans Eclipse et RSA, à l'exception d'une technologie de transition de méta-modèle. Mais on dispose d'une API Java EMF pour naviguer dans les méta-modèles source et cible. On peut donc implémenter ce transformateur en Java. Mais ce développement a un coût qu'il faut mettre en parallèle des développements "à la main". La rentabilité de cette démarche peut être quantifiée précisément et c'est un atout pour MDD : son déploiement en entreprise va s'intégrer dans une logique économique et pas seulement technologique. Il faut y inclure le coût de maintenance qui peut grimper rapidement si les générateurs sont trop touffus ou manquent de structuration. Et c'est précisément sur ce point qu'Eclipse n'apporte pas de réponse satisfaisante. Toutes les briques techniques sont là, nous l'avons vu, mais on construit la maison MDD à la main. A quoi va donc servir un outil comme RSA ? Modéliser les composants, industrialiser le processus de génération, terminer l'implémentation et tester l'application. Construire cette maison mieux et plus vite... Comment ? Le plus simple est de regarder un exemple. RSA propose des outillages graphiques et une démarche de mise en œuvre divisée en 4 phases :

**1** – Une équipe technique crée un projet de référence qui respecte les normes d'architecture de l'équipe. Bien souvent, cette application existe déjà pour démontrer les principales fonctions des frameworks mis en place.

**Livable** : un espace de travail Eclipse (composé de plusieurs projets)

**2** – A partir de cette application de référence, RSA gère une arborescence de templates JET

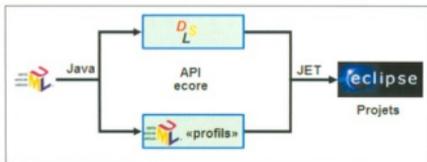
et peut extraire les paramètres attendus par les templates (démarche Model2Text). Ces paramètres sont directement liés à la phase d'implémentation. On pourrait être tenté de les spécifier dans le modèle UML de l'application mais ce serait un très mauvais exemple : le modèle serait pollué par ces informations techniques et prendrait une forme arbitraire dictée par le code. On va donc plutôt capturer l'ensemble de ces paramètres dans un méta-modèle dédié (ecore). RSA gère graphiquement l'application de référence, les templates JET et le méta-modèle ecure puis génère automatiquement le plug-in associé.

**Sachez vous brider** : le socle de l'application peut être généré facilement à partir d'un nombre limité de paramètres. Mais il devient progressivement plus difficile de compléter cette génération. Il faut donc trouver un équilibre entre le volume de code généré et la complexité du modèle associé.

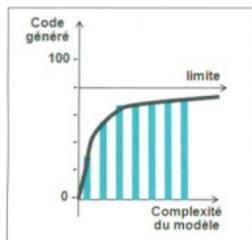
**Livable** : un plug-in Eclipse capable de générer le canevas de l'application de référence : un ensemble de templates JET et le méta-modèle ecure associé.

**3** – Il faut ensuite extraire les paramètres à partir du modèle UML de l'application en utilisant une transition Model2Model. RSA fournit un éditeur graphique de mapping pour lier le méta-modèle source et le méta-modèle cible. Cet éditeur graphique facilite énormément la maintenance du transformateur et génère automatiquement le plug-in associé. On peut y ajouter des extensions sous la forme de classes Java pour compléter la transformation et filtrer les éléments à transformer et il est appelé à s'enrichir au fil des versions pour supporter graphiquement des scénarios de plus en plus évolués.

**Livable** : un plug-in Eclipse capable d'extraire les paramètres de génération à partir d'un modèle UML.



MDD peut se résumer sous la forme de transitions :  
UML → ecore → Projet Eclipse



Il faut trouver un compromis entre complexité du modèle UML et pourcentage de code généré

4 – En enchaînant ces 2 étapes, on obtient une chaîne industrielle de génération de composants. A partir d'un modèle UML, on extrait un modèle ecore dédié qui alimente les générateurs. Cette chaîne MDD peut être livrée à l'équipe de développement métier et utilisée à grande échelle. Dans Rational Software Architect, elle vient s'insérer entre les outils de modélisation (diagrammes, rapports, analyse d'impact...) et les outils d'implémentation (éditeurs visuels, déploiement et test automatique...) pour former une plate-forme industrielle de développement.

Cette démarche d'industrialisation est une tendance majeure dans l'industrie informatique. La pression budgétaire est importante. Les projets sont de plus en plus complexes, mixant par exemple un portail Java EE attaquant des services métiers développés en COBOL. MDD permet de gagner en productivité, en qualité et de mieux gouverner les développements en partageant l'information entre les équipes. Les technologies sont enfin mûres, il ne reste donc plus qu'à les déployer en entreprise. Certaines ont déjà commencé.

#### Pour en savoir plus :

MOF : [www.omg.org/mof](http://www.omg.org/mof) - EMF : [www.eclipse.org/modeling/emf](http://www.eclipse.org/modeling/emf)  
API UML2 : [www.eclipse.org/uml2](http://www.eclipse.org/uml2) - GMF : [www.eclipse.org/gmf](http://www.eclipse.org/gmf)  
JET : [www.eclipse.org/emft/projects/jet](http://www.eclipse.org/emft/projects/jet) - M2M : [www.eclipse.org/m2m](http://www.eclipse.org/m2m) - M2T : [www.eclipse.org/m2t](http://www.eclipse.org/m2t)  
RSA : [www.ibm.com/software/awdtools/architect/swarchitect](http://www.ibm.com/software/awdtools/architect/swarchitect)



**Thierry Matusiak** travaille depuis 10 ans chez IBM dans le domaine du développement logiciel et de la génération

de composants. Il anime aujourd'hui la communauté technique IBM SWG sur les thèmes J2EE et MDD.



**Jean-Pierre Schoch** a commencé avec Rational aux Etats-Unis il y a 12 ans dans le monde de la modélisation. Avec IBM depuis 4

ans, il reste aujourd'hui passionné par les technologies orientées-objet, MDD/MDA, et par la plate-forme Eclipse.

# Génération automatique de code : les atouts de l'approche MDA

L'approche MDA consiste à maximiser l'emploi des modèles UML, pour, par des mécanismes de transformation automatique de modèles, décliner des modèles d'analyse en modèles de conception dédiés à une plate-forme, puis générer automatiquement le code. Elle a aussi été déclinée sous les termes MDD et MDE (Model Driven Engineering), qui accentuent plus spécifiquement l'aspect développement ou les phases amont d'étude.

Au-delà de la production automatisée d'un grand nombre de lignes de code à partir d'un modèle, la plus value d'une approche MDA maîtrisée apparaît surtout aux étapes cruciales d'intégration, de validation et de maintenance, où des facteurs 10 de productivité ont été attestés. Partageant des vues dédiées à chaque type de participant – du client au développeur – le modèle se décline en une vue finale détaillée : le code, tracé et maintenu en cohérence avec les autres vues. Avec MDA, l'expertise du développement est dans les règles ouvertes de transformation modèle/code : la manière optimale de traduire les formes de modélisation en code, de bien exploiter l'infrastructure ciblée, est automatisée, systématisée, constamment optimisée et garantie correcte sur toute l'application.

#### Les techniques de génération : Model Driven ou Roundtrip

L'approche Roundtrip consiste à maintenir la cohérence code/modèle en analysant le code et répercutant ses changements sur le modèle. Cette approche a l'inconvénient de remonter un modèle "physique" : on perd la modélisation la plus abstraite (précisément la plus productive et importante dans les phases antérieures), et on "pollue" le modèle avec des classes ou autres éléments de code très attachés au langage ou à l'infrastructure. L'approche Model Driven consiste à garder le modèle pour référence avec des compléments de code attachés au modèle, saisis au niveau du modèle ou au niveau du code encadré par des balises. L'avantage de cette approche est que le modèle est garanti cohérent au code, et que le code respecte de manière certaine la modélisation, en particulier issue des phases amont. Elle permet des générations plus complexes et riches que l'approche Roundtrip.

#### Au-delà de la génération de "squelettes"

L'approche MDA doit être bien intégrée aux IDE et frameworks, afin de combiner les avantages : associer les meilleurs outils, ne pas doubler les moyens d'automatiser le code afin d'utiliser chaque technique à son niveau. Elle se révèle par exemple très puissante pour assembler des Frameworks tels que Spring, Hibernate ou Struts, en guidant leur usage à travers des modèles. Cette combinaison peut permettre de produire plus de 80% du code : celui-ci prend en charge les parties déclaratives, la persistance, des mécanismes liés aux frameworks, la liaison MVC avec l'IHM, etc. La chaîne de production de code peut aussi être produite à partir du modèle de déploiement UML2.

■ Philippe Desfray – Directeur R&D SOFTEAM - [www.softeam.fr](http://www.softeam.fr)

# Panorama des outils de génération de code

*Difficile de dresser un tableau complet de l'offre de génération automatique de code. Entre la simple modélisation, le MDD, le MDA, la génération par schéma, la génération de la couche de données, le choix est immense que ce soit dans l'open source ou les outils commerciaux.*

Comme vous l'avez lu dans le dossier, les techniques sont nombreuses et le choix de l'outil dépend beaucoup du choix technique et des fonctionnalités que l'on souhaite avoir dans son environnement. Vous trouverez des générations pour tout et n'importe quoi ou presque : Java, PHP, Cobol, Fortran, etc.

**Compuware OptimaJ** : une des premiers outils MDA du marché dont il implémente en profondeur les spécifications de l'OMG. Il vise la plate-forme Java / Java EE. L'édition architecte permet de modélisation et de générer l'ensemble de l'architecture.

**Objectteering** : environnement complet capable de faire aussi bien du MDA que du MDD. Il s'agit d'un environnement complet de modélisation UML 2. Il prend en compte les profils UML. Il propose aussi bien la modélisation que le design, que les tests. Il peut se synchroniser avec Eclipse et des outils de cycle de vie tiers. Il peut générer du code Java, C++, C# n SQL, Corba ou Fortran.

**Mia Mia-Generation** : un des outils les plus réputés du marché. Il utilise de règles de générations. Il supporte UML (dont les profils UML), possède un moteur de génération basé sur les templates, propose un cycle itératif. Il est multi langage (Java, C#, XML, etc.) et supporte de nombreuses technologies et bases de données ce qui permet de l'utiliser sur des projets variés.

**Eclipse** : le projet Eclipse propose une gamme assez étendue d'outils et de librairies pour MDA et MDD. L'un des plus connus est EMF qui est l'implémentation de MOF, servant à créer un méta-modèle UML. À partir de ce méta-modèle, on peut créer d'autres modèles, des DSL, etc. Pour la partie graphique, on passera par GMF.

**Rational Software Architect** : outil phare de la gamme Rational, il s'agit d'un atelier complet de conception, de développement basé sur Java EE, UML et l'approche MDD. Il utilise une API EMF et s'appuie aussi sur les templates JET. Un outil très complet !

**Netfective Blu Age** : outil de génération de code pour Java EE et .Net. Il utilise les diagrammes UML et pour l'interface s'appuie sur XHTML. À partir des modèles et méta-modèles, il génère le code pour la cible choisie.

**AndroMDA** : outil open source de référence pour MDA. Il s'intègre à Eclipse. Il peut générer pour Spring, Java et .Net. Il s'interface avec les principaux outils de modélisation du marché et supporte la transformation des modèles.

**MagicDraw** : outil de modélisation UML 2, il est possible de faire du MDA à partir de ces modèles. Entièrement visuel, il facilite le travail de conception. Il est multi langage et autorise le round-trip.

**Softfluent CodeFluent** : outil français de fabrique à logiciel, on part d'un modèle XML pour générer le code .Net (Java en cours de finalisation). Tout est écrit en XML. Il permet de faire du mapping objet relationnel et supporte de nombreux outils tiers.

**Borland Together** : un des outils de modélisation UML les plus connus. Il génère le code et est capable de prendre en charge le bi-directionnel et la rétro conception. Utilisable avec Eclipse. Il permet aussi de modéliser des processus métiers. Et supporte le C++, Java et .Net pour la version Visual Studio.

**Lyria Leonardi** : outil de génération d'interface graphique, Leonardi s'appuie sur un méta-modèle qui fournit une structure du projet (il

peut être généré par rétro ingéniering). Ensuite, on paramètre. Le développeur peut lier son modèle à un modèle de données, puis l'outil génère l'interface. Idéal pour créer rapidement des squelettes, des maquettes.

**Effect Software ObjectAlive** : outil permettant de générer la couche d'accès aux données pour la plate-forme .Net. Il permet les persistences aux objets métiers. Un Studio est disponible pour faciliter la conception et la génération des schémas de données.

**Denki-SQL** : Denki-SQL est un petit logiciel simple et gratuit vous permettant, en partant d'un catalogue SQL Server, de générer des procédures stockées d'une part (sous forme de scripts SQL), et des classes de données d'autre part (un fichier par table).

**DataBase2Code** : Le logiciel Data Base To Code est un générateur de Code unique et innovant. Il dispose de fonctionnalités que vous ne retrouverez dans aucun générateur de code. Il permet non seulement de générer un code de mapping afin de vous faciliter l'accès aux données, mais il vous génère également le code de mise à jour de la structure de la base de données.

**OlyMars** : voici un petit outil français réalisé par Pascal Belaud, un spécialiste reconnu de SQL Server. OlyMars est un générateur de code basé sur la modélisation de données, capable de générer pour T-SQL et .net.

**Wygwam Hackhours** : le générateur de code Hackhours est une version allégée et condensée des outils de génération de code de Wygwam. Il se compose de deux parties :  
- FormFeed'Hours : permettant de lier automatiquement vos objets à vos formulaires  
- Pop'Hours : permettant de générer vos applications à partir d'une base de données

■ François Tonic

# L'expertise C++ pilotée par le modèle UML

*Objecteering 6 optimise MDA et UML2 pour produire un code C++ d'un haut niveau d'expertise : il maximise la productivité et la qualité des développements C++*

**C**omment tirer parti au mieux de la modélisation UML à des fins de production automatisée d'un code de qualité, maintenu en cohérence avec le modèle ?

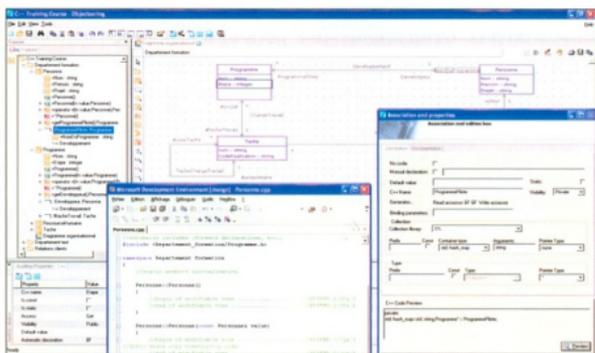
L'approche MDA qui consiste à exploiter le modèle par des mécanismes de transformation répond précisément à cette problématique en assurant également la traçabilité entre le code généré et les modèles. Avec Objecteering 6, Objecteering Software met à disposition des développeurs C++ une nouvelle génération d'outils de développement guidé par le modèle, en s'appuyant sur les dernières avancées de MDA et de UML2.1 tout en prenant en compte la définition complète du langage C++, ses librairies, ses extensions possibles et ses différences selon les plates-formes.

## La maîtrise d'un langage complexe

Le langage C++ est reconnu comme étant particulièrement complexe et nécessitant un haut niveau d'expertise : Le développeur a constamment la responsabilité de définir les modalités de gestion mémoire, à travers des choix de construction/destruction d'objet, de passage de paramètres (const, &, \*, valeur, \*\*, etc.). Il doit définir sa gestion de l'encapsulation des informations, de l'import des sources, du choix approprié des librairies, etc. Il est de ce fait difficile de garantir une bonne qualité de programmation sur une équipe de développement, où chaque ligne de code peut contenir des chausse-trappes significatives. Objecteering 6 résout ces difficultés par une approche systématique fournissant les solutions optimales et assurant une grande homogénéité dans la qualité du code. Toute la flexibilité nécessaire est également apportée pour permettre aux développeurs de contrôler ou de piloter leurs choix dans des cas spécifiques. Le code C++ est produit à 100%, dans le respect du modèle et des règles (paramétrables) de production de code. Il est complet, directement compilable, et conforme aux critères les plus exigeants de lisibilité, d'efficacité, de robustesse et de maintenabilité.

## La richesse de UML2 traduite en C++

Objecteering 6 exploite au mieux les nouvelles capacités du standard UML 2.1. Les templates UML2 permettent ainsi efficacement de modéliser et générer les templates C++. La gestion des liens d'import, couplée à un service de



Objecteering 6 : Diagramme de classes, assistant C++ et code généré.

contrôle de cohérence sur leur bonne utilisation, permet de gérer finement les includes avec leurs diverses options d'emploi. Les associations sont pleinement exploitées, jusqu'à leur qualifier et contraintes associées pour déduire les librairies de container appropriées. Les distinctions entre types de données et classes permettent d'optimiser la stratégie de gestion mémoire. L'organisation en packages et classes du modèle permet de structurer les codes sources en répertoires, interfaces (.hxx) et corps (.cxx) des classes et de gérer les namespaces C++.

La prise en compte de UML2 se prolonge jusqu'au déploiement de l'application où les diagrammes de déploiement prennent tout leur sens en permettant la génération de la chaîne de production (makefiles).

## Une gestion de cohérence modèle/code permanente

Avec Objecteering 6, le modèle est le code : le développeur complète le modèle UML par l'algorithme des méthodes, tout en étant guidé dans l'outil. La modélisation est intégralement exploitée, l'utilisateur n'ayant jamais à reformuler le modèle en C++. Les compléments de code, attachés aux éléments de modèle, sont stockés dans le référentiel. A chaque instant la cohérence du modèle UML et du code C++ est garantie. Le développeur peut ainsi programmer le corps des méthodes avec son éditeur de code ou son IDE favori : quand le modèle est modifié, Objecteering re-génère le

code de façon incrémentale et quand le code est modifié, il assure la mise à jour du modèle via son référentiel.

## Une ergonomie dédiée au programmeur C++

Lors de la modélisation, Objecteering 6 complète dynamiquement le modèle pour y ajouter les parties nécessaires ou recommandées à la programmation C++ (pattern de Coplien, patterns pour les accesseurs, etc.). Les parties ajoutées sont maintenues en cohérence avec le reste du modèle. La prévisualisation permet lors de la saisie d'un modèle, d'en connaître la forme C++ correspondante avant la génération. Des assistants aident le programmeur sur les parties les plus complexes, comme la création de patterns, ou la définition de la cible de génération.

## Objecteering SOFTWARE

Objecteering Software, éditeur de l'atelier Objecteering 6, est le spécialiste français UML/MDA pour le développement d'applications guidé par le modèle.

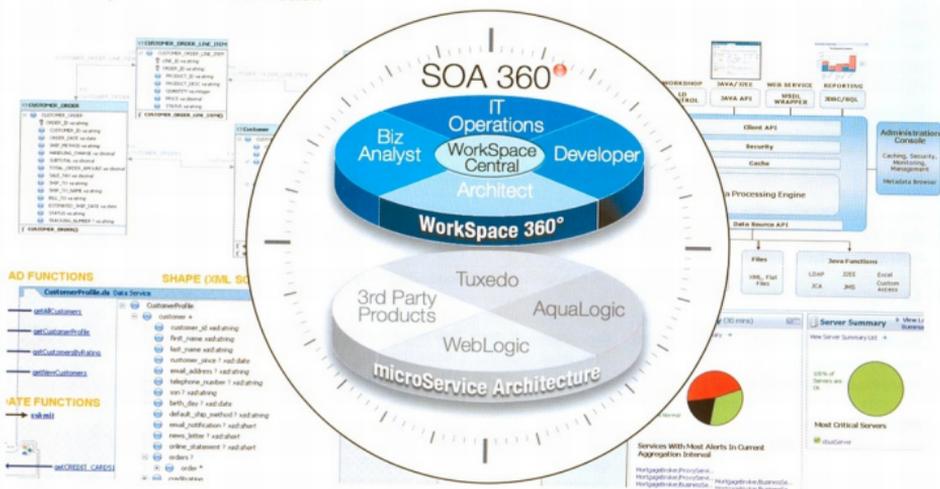
Sa suite d'outils couvre le cycle de vie de la gestion des exigences jusqu'au déploiement de l'application pour les cibles Java/J2ee, C#.Net, C++, SQL, Corba et Fortran.

Pour plus d'informations :

[www.objecteering.com](http://www.objecteering.com)  
info@objecteering.com  
Tél. : 01 30 12 16 60

# SOA : de la complexité à la facilité

L'architecture orientée service ou SOA devient un élément incontournable du système d'information et des projets. Cette architecture permet d'aligner les applications, l'informatique sur le métier, le business de l'entreprise, et surtout, en autorisant une adaptation rapide des services selon les demandes, les besoins du métier.



Nous verrons dans ce dossier comment, aujourd'hui, on peut réaliser une SOA pragmatique et efficace ! Nous verrons aussi que de nombreuses notions interviennent, parfois avec confusion. Et les éditeurs ne sont pas forcément sur les mêmes définitions d'une SOA et des composants liés. La SOA répond à des demandes, à un besoin réel. Aujourd'hui, il est possible de mettre en place une telle architecture sans forcément employer des environnements coûteux et lourds.

"Le premier problème d'une stratégie SOA est sa justification. Une fois cette étape passée, il y a toute la gestion du changement à gérer, par exemple convaincre les équipes techniques de créer des services réutilisables, comment écrire des services réellement réutilisables, réutiliser des services créés par d'autres équipes. La réutilisation de services est le cœur d'une stratégie SOA. Tout doit être mis en œuvre pour la faciliter sinon toutes les promesses en termes de réduction de coût et de flexibilité ne

seront jamais atteintes. Pour qu'un service soit réutilisable, plusieurs aspects doivent être considérés : il doit être disponible dans un catalogue centralisé de services ; il doit être documenté ; il doit être facilement adaptable à un nouveau contexte d'utilisation ; il doit pouvoir fournir des rapports d'exécution => le contraire d'une "boîte noire", nous précise Alain Gendre (BPM SOA Program Manager, ILOG).

## La SOA à l'épreuve du terrain... et des éditeurs

Rappelons que la SOA n'est pas un produit mais une architecture informatique destinée à son SI (Système d'Information). On devrait parler d'outils, d'environnements, de solutions orientées SOA. Depuis 2 ans, tous les éditeurs, ou presque, proposent une telle offre et depuis quelques mois, on conjugue la SOA avec les processeurs métiers ou BPM (Business Process Management). Les deux sont en effet complémentaires, surtout dans un cadre purement d'entreprise, comme c'est le cas la

plupart du temps pour une SOA.

Autor de l'architecture orientée services, un certain nombre de techniques, de technologies gravitent. On peut citer pour les principales : - la BPM - la gouvernance - le cycle de vie - la qualité et les bonnes pratiques - la sécurité - l'orchestration (à différencier de la gouvernance)

La conjugaison SOA et BPM n'est pas due à un hasard. Cela permet de créer des processus métiers et d'y greffer les services au-dessus. Côté flexibilité, le BPM apporte une adaptation aux aspects réglementaires qu'une entreprise doit pouvoir respecter et surtout adapter à chaque modification. Un éditeur comme Oracle travaille beaucoup sur le BPM, ce n'est pas un hasard si l'éditeur a racheté un spécialiste des règles métiers : Collapsa.

## Avoir une vision SOA de son SI, c'est quoi ?

On associe souvent SOA à l'urbanisation de son SI, c'est-à-dire que l'on restructure son

patrimoine applicatif pour en avoir une meilleure vision, une meilleure approche. L'urbanisation est un des grands termes à la mode depuis quelque temps. Alliée à l'urbanisation, on trouve la cartographie qui est une étape incontournable. Si vous ne faites pas d'urbanisation pure et dure, la cartographie constitue une étape essentielle, voire stratégique pour pouvoir répertorier son contenu applicatif et avoir une bonne idée des composants et services existants et cela permet au bout du compte de se constituer une carte alliant dans son futur annuaire et référentiel de services qui va être la colonne vertébrale de sa future SOA.

" SOA est une vision de l'architecture des SI qui implique la collaboration de technologies très diverses. Ce qui explique l'importance des standards pour sa réalisation. Les standards aboutiront, parce que les parties prenantes savent toutes que dans un cadre SOA, personne n'aura le monopole des solutions utilisées, même sur une petite partie de l'architecture. Le travail commun est donc incontournable. Pourtant, il y aura toujours des acteurs qui apporteront des fonctionnalités au-delà des standards, ou qui dériveront à partir des standards. Pour la simple raison que le choix n'est pas purement technique, mais économique. Nous sommes dans un univers concurrentiel, où chacun des acteurs cherchera à se différencier et/ou à créer une solution de "lock-in" pour le client. La contradiction entre le modèle open source et le modèle économique dominant ainsi que la bataille autour du brevetage du logiciel illustrent bien la situation. Seule l'histoire dira quelle en est l'issue... " explique Daniel Calchi (Product Marketing Manager chez Software AG).

La question que l'on doit se poser quand on souhaite une démarche SOA, est de savoir ce que l'on veut mettre dedans, ce que l'on met dans les services, comment les définir, la granularité des services. Et bien entendu se demander au préalable : quelle valeur ajoutée la SOA va-t-elle me donner ? Car n'oublions pas qu'avec la SOA, même si le mouvement n'est pas neuf, on glisse du technique au métier (= business). Car la SOA touche directement le cœur métier de l'entreprise. " On voit aujourd'hui qu'une approche SOA est pertinente, non pas dans un cadre d'urbanisation, mais pour le problème d'ouverture de son SI avec les partenaires / clients. " précise Pascal Bobon (Cheops, partenaire Oracle).



### Quid de la direction, de la gestion d'une SOA

" Nous avons des clients qui viennent nous voir qui utilisent des services web et ils se posent une question : quid de la gouvernance de ces services ? " précise Xavier Gérard (Sun France). Effectivement, depuis l'apparition de la SOA, la gouvernance tend à devenir un enjeu dans toute architecture de services ayant quelques centaines ou milliers de services. Pour les petites architectures, la gouvernance n'est pas obligatoire cependant il est nécessaire de mettre en place des règles d'orchestration et surtout un référentiel pour gérer et répertorier ses services internes ou externes. Le référentiel est le cœur même de toute SOA. Mercury propose une solution tout en un : Systinet qui inclut le référentiel service et la gouvernance. On peut y classer les services, y définir les entrées / services des services, y gérer les autorisations. " Les plates-formes SOA sont en train de faire un effort autour de la gestion du catalogue de services (Service Registry and Repository) et du BAM (business Activity Monitoring). " commente Alain Gendre. On reprend souvent le terme de " registre de services ". Ce registre sert à tout et notamment à son bus ESB. L'ESB y puise les informations de contrats, les dépendances, les implémentations. Cette notion de registre recouvre deux besoins : la nécessité de posséder un annuaire de services contenant les contrats et le référentiel contenant les contrats, la description des dépendances, l'implémentation, les autorisations utilisateurs, etc. L'annuaire peut reposer sur UDDI (l'annuaire par défaut des services web) ou non. Notons aussi que la spécification SCA concerne le référentiel.

### Systinet 2

Système de suivi et ensemble d'applications de gouvernance SOA

#### Visibilité

Un seul système de suivi pour découvrir et comprendre les services métier

#### Confiance

Qualité, prévision et transparence totale entre les consommateurs et les fournisseurs garantes

#### Contrôle

Gestion du cycle de vie des services métier et gestion efficace des changements

Qu'entend on par gouvernance ? Il s'agit avant tout de pouvoir centraliser la gestion de sa SOA à tous les niveaux pour le développement, la consommation, l'exposé des services et surtout d'en assurer le cycle de vie et d'y surveiller l'activité de son architecture, de chaque service, de surveiller son ESB (bus d'entreprise basé sur XML), de gérer ses policy, la sécurité, les accès et les autorisations de chaque service et surtout des utilisateurs. Bref, la gouvernance doit offrir une tour de contrôle globale de sa SOA et dans ce cas, on peut passer par des tableaux de bord pour agréger les données correspondantes à chaque profil d'utilisateur de la gouvernance. Car un développeur n'aura pas besoin des mêmes informations qu'un architecte, qu'un testeur, etc.

L'autre souci potentiel concerne l'identité des utilisateurs. " Comment gérer l'identité ? On avait connu cela avec l'annuaire LDAP. On a maintenant une granularité plus fine. C'est pour cela que l'on parle de gouvernance. Il faut une plate-forme de collaboration, un annuaire, pour concevoir, exposer, consommer des services. " commente Marc Boullier, directeur technique de Vistali.

D'une manière générale, plus on augmente le nombre de services, plus la SOA devient complexe à gérer, à mettre en œuvre. Il faut bien avouer, qu'aujourd'hui, créer une SOA avec plusieurs milliers de services s'avère problématique pour la montée en charge, la gouvernance.

### Cherchez spécifications désespérément

Depuis peu, les premières spécifications SOA sont disponibles : SDO et SCA. OpenSOA travaille activement à définir un cadre plus formel pour définir l'architecture de service, faciliter

le développement et la mise en place, et assurer aussi une bonne interopérabilité.

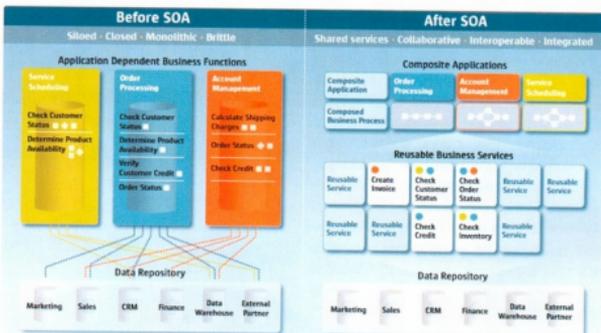
" SCA, SDO sont appelés à rester car ils adressent des problèmes essentiels à une stratégie SOA. Chacun a pour but de rendre le système d'information beaucoup plus flexible et adresser un problème bien spécifique. Avec SDO, l'accès aux données se trouve simplifié et la représentation des données devient indépendante des systèmes sous-jacents. Avec SCA, le travail de développement de services métier est simplifié en cachant la complexité des systèmes sous-jacents, facilitant ainsi l'assemblage de composants. L'émergence et la maturité de ces deux standards correspondent au besoin les plus immédiats d'une approche SOA : 1) la création et l'assemblage de services, 2) l'accès aux données. D'autres standards vont émerger dès que ces problèmes urgents seront adressés", analyse Alain Gendre.

" L'initiative Open SOA ([www.osoa.org](http://www.osoa.org)) regroupe 18 acteurs majeurs de SOA. Le groupe vient d'annoncer que les standards SDO et SCA rejoignent OASIS ([www.oasis-open.org](http://www.oasis-open.org)). En effet, ils sont désormais considérés comme matures, leur arrivée dans OASIS va permettre d'améliorer l'interopérabilité avec d'autres standards importants de SOA comme WS-\*. Certains s'inquiètent de ce que cela pourrait freiner la dynamique autour de SDO et SCA... C'est pour cela que le groupe Open SOA a décidé de continuer ses travaux, notamment pour défricher efficacement de nouveaux axes importants comme le standard DAS (Data Access Service)" précise Eric Samson de Xcalia.

Le développeur ou l'entreprise sera plus ou moins au courant de ces spécifications, les éditeurs jouent donc un rôle important. " On sensibilise le client sur les standards. Les clients en veulent mais ils ne cherchent pas à être standard avec le monde entier", indique Pascal Bobon. Bref, ces spécifications constituent une avancée indéniable. " C'est une bonne chose mais il faut se méfier de la maturité. Il faut environ 3 ans pour que les applications les intègrent. SCA définit un cadre commun, fournit un cadre, normalise l'ensemble", précise Marc Boullier.

## La sécurité dans la SOA

La sécurité demeure un point crucial des applications. Le problème existait avec les web services et il apparaît aujourd'hui avec les architectures de services. Cette faiblesse peut être partiellement contournée et palliée, selon



le contexte de mise en œuvre. Les acteurs de la SOA demeurent souvent discrets sur le domaine de la sécurité, du moins en France. Sans doute une question de maturité du marché. Aux USA, la réflexion de sécurité SOA est en cours et le marché propose déjà quelques outils. Ainsi BEA commercialise AquaLogic Enterprise Security... solution non vendue en France, faute de demande. Par contre, Oracle propose ses mécanismes de sécurité dans sa SOA Suite.

Cependant, la SOA, en France du moins, se limite souvent à l'usage interne dans un Intranet ou dans un extranet avec des fournisseurs, des intervenants définis, identifiés et qualifiés au niveau des accès, des échanges de données et des services (pour la consommation et l'exposition). Cela limite les intrusions. On dispose de tout un arsenal éprouvé pour sécuriser les sessions, les services, les échanges comme SSL, IPsec, les spécifications sécurité WS-étoile, etc.

La qualité des services sera un gage de stabilité et d'intégrité de son architecture. La menace, réelle ou supposée, du " rogue service " ne doit pas être sous-estimée. Il est nécessaire de le prendre en compte, notamment quand on consomme ou orchestre des services étrangers.

## Les menaces potentielles

L'insécurité se situe à plusieurs niveaux : les services, l'architecture, l'infrastructure. Chaque niveau possède ses propres faiblesses. Par exemple sur le service, de type web service, on retrouve l'altération de messages, l'accès aux données par des personnes non autorisées, l'usurpation

d'identité, des dépassements de capacité, l'injection de script, etc. Dans une SOA, la sécurité doit être une démarche volontaire et systématique. Car, on touche directement à l'activité de l'entreprise, à son business. On retrouvera donc les remèdes traditionnels des services tels que l'utilisation des signatures numériques XML, de WS-Security. Plus vous avez de services, plus le risque augmente. Il augmentera d'autant plus rapidement que vous exposez et surtout consommez des services extérieurs ou d'anciens services, pas toujours développés avec la rigueur nécessaire.

Il faut pouvoir assurer la disponibilité des informations (donc des services), l'intégrité des données, et la confidentialité. Deux critères s'avèrent importants : pouvoir tracer un processus, un service, s'assurer de l'identité et d'une manière générale gérer la non-répudiation.

## Quelles solutions possibles ?

" Il s'agit d'un problème significatif. On multiplie les accès. La sécurité restera du ressort de l'implémentation du service. Il faut un mécanisme unique et mettre dans l'infrastructure un premier niveau de sécurité. Il faut aussi avoir une " policy ". On va vers la définition de la sécurité comme un service, tendance que l'on observe", commente Yves Lhéroult de BEA. Pour une bonne sécurité, il faut tenir 5 critères fondamentaux : authentification, confidentialité, intégrité, disponibilité, traçabilité.

## Pour aller plus loin

Collectif, SOA le guide de l'architecte, DUNOD, 2006

■ François Tonic

# Soyez prêts pour les nouvelles architectures SOA!

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

SOFTEAM, 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.

Cartographie

Urbanisation

BPM

Portail

EAI

XML

XML et WebServices

Frameworks J2EE

WebServices en .Net

Méthodologie pour SOA

Méthodologie

Architecture SOA

UML

Architecture

Web Services

WebServices en J2EE

.NET

WebServices en .Net

## SOFTEAM Formation 2007 :

<b>UML, Méthodes et Méthodologie</b>	
Introduction aux technologies objets et à UML	1 j
Les diagrammes UML	2 j
UML pour la maîtrise d'ouvrage et l'expression de besoins	2 j
Analyse et conception avec UML	4 j
Analyse et conception avec UML et un atelier de modélisation	5 j
Processus de développement objets : Etat de l'art	2 j
De UML 1.x à UML 2.0	1 j
<b>SOA</b> Méthodologie pour SOA	2 j
<b>Architecture et Intégration</b>	
Architecture et intégration : la synthèse	3 j
<b>SOA</b> Architecture SOA	2 j
Architecture et approche MDA	3 j
Design Patterns avec Java / J2EE	3 j
Design Patterns avec .Net	3 j
Corba : Concepts et mise en œuvre	4 j
<b>SOA</b> Développement de Web Services en Java	3 j
<b>SOA</b> Développement de Web Services en C#	3 j
<b>Développement Java /J2EE</b>	
Programmation orientée objet avec Java	4 j
Programmation Java avancée	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	3 j
Maîtrise du framework JSF - Java Server Faces	3 j
Maîtrise du framework Spring	3 j
Mapping Objet / Relationnel avec Hibernate	3 j
Développement d'IHM avec Swing	3 j
Développement d'IHM avec Ajax	2 j
Développement d'IHM avec Eclipse RCP	4 j
Développement d'IHM avec SWT	2 j
<b>SOA</b> Développement de Web Services en Java	3 j
Développement d'applications embarquées avec J2ME	3 j
Développement de portlets	3 j
<b>Développement .NET</b>	
Développement d'applications .NET / C#	4 j
<b>SOA</b> Développement de Web Services avec C#	3 j
Principes et architecture de la plate-forme .Net	1 j
Conception et développement avec .Net 2.0	5 j
Développement d'applications Web avec ASP.Net 2.0	5 j
Programmation avec Microsoft ADO.Net	3 j
<b>Développement XML et autres langages</b>	
Développement d'applications avec XML	4 j
Formatage et transformation de documents XML avec XSL	2 j
Développement d'applications vocales avec VoiceXML	3 j
Développement en PHP	3 j
Développement d'applications avec C++	3 j
<b>Atelier Objecteering/UML</b>	
Objecteering UML Requirements	1 j
Objecteering UML Modeler	1 j
Objecteering Java Developer	2 j
Objecteering C# Developer	2 j
Objecteering C++ Developer	1 j
Objecteering UML Profil Builder	2 j
Objecteering MDA Modeler	2 j

Calendrier complet et inscriptions en ligne

[www.softeam.fr](http://www.softeam.fr)



Tél. : 01 53 96 84 00 - Fax : 01 53 96 84 01

Paris : 21 avenue Victor Hugo - 75016

Rennes - Nantes - Sophia Antipolis

**SOFTEAM**  
Think Object

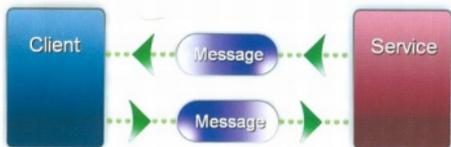
## Windows Communication Foundation : La SOA facile selon Microsoft

Connaissez-vous COM, DCOM, MSMQ, .NET Remoting, Web Services XML, ou Enterprise Services ? Il s'agit d'un ensemble de technologies de communication fournies par Microsoft au cours des 20 dernières années. Elles répondent chacune à un besoin particulier de communication, tantôt synchrones ou asynchrones, transactionnelles ou non, très performantes et propriétaires ou moins performantes, moins fonctionnelles mais interopérables.

L'architecture des applications suit une évolution graduelle depuis plusieurs décennies pour résoudre les problématiques de communication entre applications. Aujourd'hui, on entend beaucoup de buzz autour des Architectures Orientées Services (SOA), mais de quoi s'agit-il en fait ? Le .NET Framework 3.0 qui contient Windows Communication Foundation (WCF) est-il la réponse de Microsoft pour construire facilement des Applications Orientées Services pour les besoins métier d'aujourd'hui et de demain ? Nous allons essayer d'y répondre à travers cet article.

### Passer d'une Architecture Orientée Service à WCF

Le développement d'Architectures Orientées Services est aujourd'hui le moyen le plus sûr de construire des applications robustes, sécurisées, et maintenables. Une SOA peut être simplement définie comme un concept visant à utiliser un ensemble de services pour réaliser une fonctionnalité désirée. Un service est un système autonome qui accepte des requêtes et retourne des réponses via un ensemble d'interfaces bien connues et exposées. Contrairement à des architectures traditionnellement fortement couplées, SOA implémente un ensemble de services faiblement couplés qui contribuent collectivement au résultat voulu.



Lorsqu'on développe une application orientée service, on découple le code du service de la technologie et de la plate-forme utilisée côté client. Parce que l'interaction entre le client et le service est basée sur des standards, la plomberie permettant de décrire comment sécuriser l'appel, gérer les transactions, ou gérer la fiabilité est déjà implémentée. Si la plomberie évolue, l'application n'est pas impactée, puisque découplée de ces aspects. Une application orientée service est robuste car les développeurs peuvent s'appuyer sur une plomberie éprouvée et testée. Il en résulte une meilleure productivité, car se concentrant plus sur les fonctionnalités que sur la plomberie.

L'équipe de WCF décrit les quatre piliers fondamentaux d'un service :

- Des frontières délimitées
- Les services sont autonomes
- On partage des contrats pas des objets

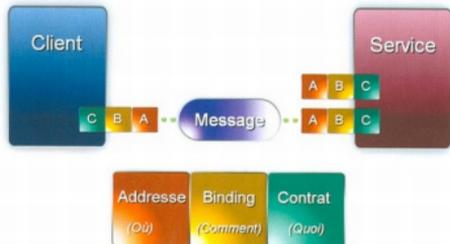
- Des règles définissent la compatibilité

Au final, ces règles se traduisent dans la pratique comme suit :

- Services isolés
- Indépendant de la localisation
- Indépendant du transport, du protocole et du format
- Indépendant des plates-formes et de l'implémentation
- Doit pouvoir fonctionner quels que soient les dispositifs de montée en charge mis en place
- Doit pouvoir fonctionner même lorsque l'une des parties n'est pas disponible (asynchrone)

### Implémentation d'une SOA avec WCF

Concrètement, un service WCF est un élément logiciel qui répond à des communications réseau. La description d'un service WCF est similaire à celle d'un Web Service dans le sens qu'il doit préciser où il se trouve, comment il communique et ce qu'il fait. Pour un Web Service, cette description doit être consignée dans un fichier WSDL. Ces informations sont définies dans les éléments service, binding et portType. Dans la terminologie de WCF, ces informations s'appellent Address, Binding et Contract. Ces termes sont plus facilement mémorisables puisque se traduisant par les acronymes A, B, C. Un service WCF peut exposer un ou plusieurs points de communication, chacun d'eux étant constitué d'une adresse, d'un binding et d'un contrat.



### Le contrat

Le contrat permet au service de décrire explicitement la liste des opérations qu'il expose. La définition du contrat est très simple pour un développeur .NET puisque cela consiste à déclarer une interface au sens langage .NET, à définir un certain nombre de méthodes et à la décorer de quelques attributs.

```
[ServiceContract()]
public interface IService1
{
    [OperationContract]
    string MyOperation1(string myValue);
    [OperationContract]
    string MyOperation2(DataContract1 dataContractValue);
}
```

Il est à noter que seules les méthodes décorées de l'attribut [OperationContract] pourront être exposées par le service. Le contrat définit la liste des opérations implémentées par le service. Il peut également spécifier des comportements orthogonaux par l'ajout de propriétés sur les attributs ServiceContract ou OperationContract pour préciser notamment :

- La fourniture d'un contrat bidirectionnel permettant non seulement d'initier une communication du client vers le serveur, mais aussi du serveur vers le client.
- Le support d'une session pour conserver l'état entre l'appel de plusieurs opérations
- L'attente ou non d'un message d'acquiescement suite à l'exécution d'une opération ne retournant pas de résultat.
- Une implémentation explicite de l'opération en mode asynchrone. Notons que ce choix ne préjuge en rien la façon dont le client devra effectuer son appel.

L'étape suivante consiste à implémenter le service à proprement parler. Il suffit pour cela de créer une classe qui implémente le contrat. Dans le cas où une structure de données doit être échangée entre le fournisseur et le consommateur de service, il devient intéressant de spécifier un ou plusieurs contrats de données (DataContract)

```
public class service1 : IService1
{
    public string MyOperation1(string myValue)
    {
        return "Hello: " + myValue;
    }
    public string MyOperation2(DataContract1 dataContractValue)
    {
        return "Hello: " + dataContractValue.FirstName;
    }
}
```

```
[DataContract]
public class DataContract1
{
    string firstName;
    string lastName;
```

```
[DataMember]
public string FirstName
{
    get { return firstName; }
    set { firstName = value; }
}
[DataMember]
public string LastName
```

```
{
    get { return lastName; }
    set { lastName = value; }
}
```

Lors de l'implémentation du service, il est possible de préciser des comportements supplémentaires tels que :

- Le support d'un appel concurrent multi thread ou non
- La coordination d'une transaction sur le service initiée depuis le client, en précisant le niveau d'isolation
- La conservation du contexte au niveau d'une méthode, de la session, ou en mode singleton
- L'impersonnification sur l'exécution d'une méthode

### Le Binding

Un binding permet de décrire le protocole de communication utilisé pour se connecter à un point de communication d'un service WCF. Le binding définit plusieurs axes paramétrables dont le protocole de transport mais pas uniquement !

- Méthode de transport utilisé pour la communication entre les applications situées sur une même machine ou non. (HTTP, HTTPS, TCP, MSMQ, named piped)
- Format d'encodage (Text, Binaire, MTOM)
- Contraintes de sécurité
- Contraintes de garantie de remise
- Support des transactions

Afin de faciliter le choix parmi les nombreuses combinaisons possibles, WCF est livré avec neuf bindings prédéfinis, adaptés à la plupart des scénarios classiques.

### L'adresse

Toute communication avec un service Windows Communication Foundation (WCF) se fait en passant par un point de communication. Un point de communication consiste en une adresse, un binding et un contrat. L'adresse spécifie où se trouve le service. Elle est définie par une URI. Bien qu'un point de communication puisse être spécifié de façon impérative dans le code ou déclarativement dans le fichier de configuration, la méthode déclarative est à privilégier. Les choix effectués en développement pour le binding et l'adresse étant bien souvent différents de ceux utilisés lors du déploiement du service, il est préférable de les rendre facilement modifiables par configuration plutôt que d'être obligés de modifier le code et de le recompiler suite à des changements.

Un même service WCF peut exposer plusieurs points de communication et donc être accessible depuis plusieurs adresses. Voici un exemple de point de communication utilisant le contrat IService1, le binding WSHttpBinding, et l'adresse "http://localhost:8080/service1"

```
<endpoint address="http://localhost:8080/service1"
    binding="wsHttpBinding"
    contract="WCFServiceLibrary1.IService1"
/>
```

### L'hébergement d'un service WCF

Un service WCF peut être hébergé par n'importe quelle application .NET. On utilise habituellement une application console ou Windows Forms

pendant la phase de développement d'un service pour faciliter sa mise au point. Voici l'exemple de code permettant d'héberger notre service dans une application console.

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

using System.ServiceModel;

namespace WCFServiceHost
{
    class Program
    {
        static void Main(string[] args)
        {
            MyServiceHost.StartService();
            Console.WriteLine("Le service est lancé...");
            Console.ReadKey();
            MyServiceHost.StopService();
            Console.WriteLine("Le service est arrêté...");
        }
    }
}

internal class MyServiceHost
{
    internal static ServiceHost myServiceHost = null;

    internal static void StartService()
    {
        myServiceHost = new ServiceHost<typeof(WCFServiceLibrary1.service1)>;
        myServiceHost.Open();
    }

    internal static void StopService()
    {
        if (myServiceHost.State != CommunicationState.Closed)
            myServiceHost.Close();
    }
}
```

Pour lancer le service, il n'y a que deux lignes de code à écrire:

- Créer une instance de la classe ServiceHost, en lui passant dans le constructeur le type du service.
- Appeler la méthode Open()

Les informations de binding et adresse sont définies dans le fichier de configuration app.config suivant :

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.serviceModel>
    <services>
      <service name="WCFServiceLibrary1.service1" behaviorConfigura
tion="httpGetEnabledBehaviors">
```

```
<host >
  <baseAddresses>
    <add baseAddress="http://localhost:8080/service1"/>
  </baseAddresses>
</host>
<endpoint address=""
  binding="wsHttpBinding"
  contract="WCFServiceLibrary1.Service1"
  />
</service>
</services>
<behaviors>
<serviceBehaviors>
  <behavior name="httpGetEnabledBehaviors" >
    <serviceMetadata httpGetEnabled="true" />
  </behavior>
</serviceBehaviors>
</behaviors>
</system.serviceModel>
</configuration>
```

Afin de ne pas avoir à copier sur le client le contrat de service (ServiceContract) et le contrat de données (DataContract) manuellement, le "serviceBehavior" httpGetEnabled permet de les exposer sur http comme le font les Web Services .ASMX pour leur fichier WSDL.



Fenêtre d'exécution du serveur

Pour un hébergement en production, il est préférable de passer par une application de type "Service Windows", Internet Information Services (IIS), ou encore "Windows Process Activation

Service (WAS)". WAS est le nouveau mécanisme d'activation de processus disponible dans Windows VISTA et qui sera également disponible dans "Windows Server Longhorn". IIS 7.0 utilise WAS pour accomplir l'activation basée sur des messages transmis par http. Utiliser directement WAS pour l'hébergement des services WCF permet de bénéficier des mêmes avantages de robustesse ou de monitoring fournis par IIS non seulement sur http, mais aussi sur TCP, MSMQ et les Named Pipes.

## La consommation d'un service WCF

Une application cliente doit connaître les contrats qui correspondent au service auquel elle doit accéder. La meilleure approche consiste à utiliser l'outil Svcutil. Cet utilitaire est capable de récupérer les contrats, de générer le code proxy client permettant de faciliter la manipulation du service WCF et de générer le fichier de configuration client

```
svcutil http://localhost:8080/service1 /out:c:\proxy.cs /config:c:\app.config
```

il suffit ensuite de créer un nouveau projet .NET, d'ajouter une référence sur la dll WCF System.ServiceModel.dll, d'incorporer les fichiers générés par svcutil, et de jouer avec une instance de la classe proxy.

```
class Program
{
    static void Main(string[] args)
    {
```

```
Console.WriteLine("Lancement du client\nAppuyez sur une touche.");
Console.ReadKey();
```

```
Service1Client obj = new Service1Client("Endpoint1");
string resultat = obj.MyOperation1("Daniel");
Console.WriteLine("Résultat appel opération 1 :{0}", resultat);
```

```
WCFServiceLibrary1.DataContract1 objData = new WCFService
Library1.DataContract1();
objData.FirstName = "Bill";
objData.LastName = "Gates";
resultat = obj.MyOperation2(objData);
Console.WriteLine("Résultat appel opération 2 :{0}", resultat);

Console.ReadKey();
}
```

On peut remarquer que dans le constructeur de la classe proxy on peut spécifier un paramètre qui correspond au nom du point de communication défini dans le fichier de configuration.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
<system.serviceModel>
<client>
<endpoint name="Endpoint1"
binding="wsHttpBinding"
address="http://localhost:8080/service1"
contract="Service1"/>
</client>
```

```
</system.serviceModel>
</configuration>
```



Voici le résultat de l'exécution de l'application sur le client

## Conclusion

Nous avons vu au regard de cet article la puissance et la simplicité du modèle de programmation des services avec WCF. Windows Communication Foundation est la nouvelle couche de communication qui rend obsolète la façon de programmer les WebServices, le .NET Remote ou MSMQ que nous avons jusqu'à présent sur la plate-forme .NET. Les développeurs y trouveront un moyen de consommer n'importe quel service de la même manière, quel que soit le protocole de communication sous-jacent. WCF permet aux développeurs de se concentrer sur le service en lui-même puisque toute la tuyauterie de communication est maintenant fournie de base sous forme déclarative et indépendante de l'implémentation du service.

WCF est non seulement adaptée pour la communication entre applications 100% .NET et WCF, mais reste également le meilleur choix pour assurer une interopérabilité cross-plates-formes avec les tous derniers standards de communication. Windows Communication Foundation est donc sans aucun doute une plate-forme de communication et solution technique qui séduira les architectes pour implémenter une SOA.



■ Daniel TIZON - Winwise  
Microsoft Regional Director

Email : [daniel.tizon@winwise.fr](mailto:daniel.tizon@winwise.fr)

Blog : <http://blogs.codes-sources.com/daniel/>

Site : <http://www.winwise.fr>

www.programmez.com

Votre portail web !

**NOUVEAU**

Emploi

Un espace emploi dédié

Riche en actualité et en articles dédié à l'emploi, Programmez vous permet enfin de consulter les offres publiées et de poster directement en ligne



Interactivité

Blogs

Who's Who

Forums

Boutique

Achetez les magazines et les articles en PDF



Nouveau Espace HUMOUR

Les histoires drôles, photos et vidéos des informaticiens

## La SOA pragmatique

Adopter une approche SOA est facile en théorie mais un peu moins en pratique. Il faut en effet éviter deux écueils : l'écueil du "Marteau sans maître", et l'écueil "Terminator". Le marteau, c'est le mot clef "service" : avec ce mot magique, tout devient alors service dans la conception de nos applications : or les utilisateurs finaux (ceux qui nous font vivre...) n'attendent pas une architecture de services logiciels, mais des réponses à leur besoin. Terminator, c'est les Web Services : non seulement on voit tout sous forme de service, mais en plus il faudrait les coder systématiquement à l'aide de l'attrail des normes WS-\*, sans plus attendre.

L'objet de cet article est de proposer une approche raisonnée de l'usage des principes SOA. Cette approche part du principe qu'il y a plusieurs possibilités de mise en œuvre de ces principes, selon ce qu'on peut appeler des patterns architecturaux. Pour comprendre ce que sont ces patterns, on introduira au préalable un modèle du système d'information. L'article présentera ensuite les différents outils associés à ces patterns, en citant quelques pièges à éviter.

### D'abord modéliser

Le modèle présenté dans la figure 1 part du principe qu'un système d'information (SI) doit aider l'entreprise à traiter les événements métier qu'elle reçoit. Un SI moderne doit aider l'entreprise à traiter ces événements **en temps réel ou quasi réel**. Le modèle proposé distingue donc plusieurs couches :

- La couche de récupération des événements métier
- La couche processus métier
- La couche service métier
- La couche service technique
- La couche bases de données & référentiels

Ce modèle ne suffit pas nécessairement pour représenter un SI très hétérogène, organisé en silos, ce qui est souvent le cas des très grandes entreprises. Il est nécessaire de prévoir alors une couche de communication entre ces silos, afin de faciliter la mise en place de processus transverses, comme l'illustre la figure 2.

### La couche de gestion des événements entrants

Quel que soit le canal d'arrivée de ces événements (fichier plat, message XML, formulaire rempli sur le portail Internet, etc.), cette couche a pour objectif de router l'événement

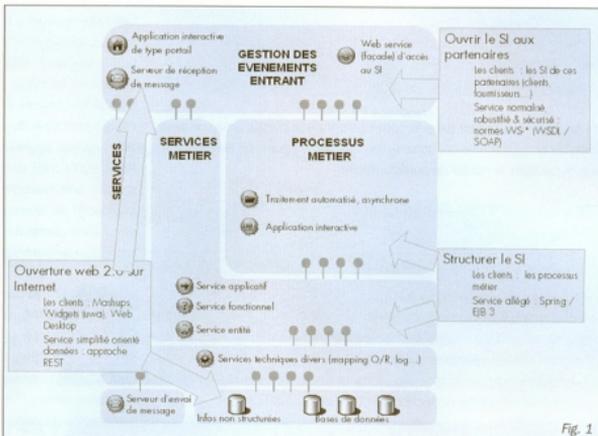


Fig. 1

métier reçu vers le processus métier qui devra traiter l'événement.

Elle peut bien évidemment assurer d'autres tâches, telles que la validation syntaxique des événements entrants ou la sauvegarde systématique de ces événements à des fins de robustesse (aucun événement ne doit être perdu) ou d'audit (quels types d'événements sont reçus via quel canal d'entrée ?). Elle s'appuiera pour ce faire sur des services techniques (log...) ou métier (service de validation, service de routage).

Cette couche peut offrir comme canal privilégié d'interaction des (web) services accessibles directement par les systèmes d'information des partenaires et clients de l'entreprise.

### La couche des processus métier

Cette couche a pour objectif de traiter les événements qui lui sont adressés. Un processus

métier peut revêtir plusieurs formes. Dans un contexte SOA, ce sera un processus automatisé, orchestrant l'appel aux services métier (un processus étant décrit via le langage BPEL). Un processus BPEL est lui-même un service, accessible par la couche de gestion des événements entrants.

Cette approche est la seule qui permette d'assurer que tous les événements seront traités en temps et en heure.

Mais dans certains cas, un événement peut/doit être routé vers un acteur humain, cet acteur utilisera alors une application interactive pour traiter l'événement. L'application interactive sera, elle aussi, cliente des services métier.

Enfin, il ne faut pas oublier les batches, qui peuvent traiter un lot d'événements entrants lorsque c'est pertinent (ensemble de commandes émises par la même entreprise, lot de chèques, etc.).

### La couche des services métier

Chaque service métier doit offrir un ensemble cohérent de traitements métier : cela peut concerner par exemple l'accès à la " vision client " (obtenir un ensemble synthétique d'informations sur le client, l'état de ses comptes, l'historique de ses relations avec l'entreprise...), ou la simulation d'un devis, etc.

Un service métier peut être soit un service d'accès à des informations, soit un service de calcul et de vérification de règles métier, soit une composition des deux. La composition de service devrait jouer un grand rôle dans nos futures pratiques de développement, nous y revenons un peu plus loin.

### La couche des services techniques

Le rôle d'un service technique est de donner accès à une ressource technique donnée : on citera par exemple l'accès aux bases de données relationnelles, au CICS d'un mainframe, à un outil de log, etc.

### Ensuite structurer : les patterns SOA

Comme la figure 1 le met en évidence, on distinguera donc 3 patterns architecturaux de mise en œuvre de l'approche SOA, en abrégé " patterns SOA ".

Chaque pattern a pour objectif la conception et le déploiement de services. Ces patterns sont présentés dans le tableau comparatif ci-contre. Mettre en œuvre SOA, c'est donc dans un premier temps déterminer le type de Pattern Architectural à appliquer. Certains projets ne nécessiteront qu'un seul de ces patterns. D'autres au contraire en nécessiteront deux, voire trois.

Dans ce qui suit, nous nous intéresserons aux patterns 1 et 3. Le pattern 2 repose sur des outils (Spring, EJB 3) bien connus des lecteurs de Programmez.

### Mettre en œuvre les patterns architecturaux : quels outils ?

#### Outils pour le pattern " ouvrir le SI "

Les services à mettre en place pour cette ouverture ont trois caractéristiques :

- ils sont naturellement distribués : c'est-à-dire que les clients de ces services sont distants, et accèdent aux services via Internet (cas de la figure 1) ou via un réseau interne (cas de la figure 2). Cette contrainte incontournable implique de prendre des précautions en terme de **performance** ;
- ils doivent dialoguer avec leur client indépén-

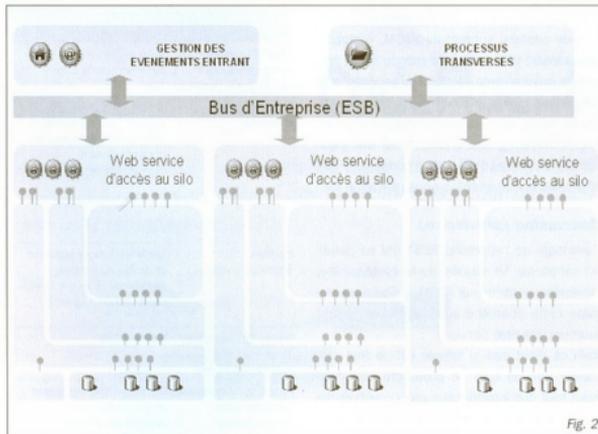


Fig. 2

damment de la technologie respective d'implémentation du service comme du client : **l'interopérabilité** est une condition sine qua non du succès.

- enfin, ils doivent offrir un minimum (voire un maximum !) de **robustesse** et de **sécurité** : la robustesse comme la sécurité ne sont pas des contraintes purement techniques ! Elles comportent un enjeu business fort. En effet, l'accès à ces services peut être payant : un partenaire ne paiera pas pour un service non robuste ! Par ailleurs un service non sécurisé peut conduire à des dégâts dont on connaît l'impact économique potentiel.

La nécessité de rendre interopérable les mondes Java, .Net, PHP, mainframe etc. ont conduit à l'éclosion des normes WS\*.

Afin d'offrir des garanties en terme de robustesse, un service exposé sur Internet doit publier ses engagements de service (SLA (2) de 99% par exemple) et se donner les moyens de les respecter.

Les engagements que prend un fournisseur de service (vous !) en matière de disponibilité sont décrits dans ses conditions générales de vente, et éventuellement dans un fichier au format WS-Policy. Cette norme a en effet été créée pour permettre d'explicitier la politique d'un fournisseur de service.

Mais attention : pour pouvoir respecter sa SLA, un fournisseur de service doit monitorer ses accès. Il doit aussi réguler ses accès en authentifiant les consommateurs du service.

L'authentification permet de réguler les utilisateurs simultanés, mais aussi d'assurer un accès restreint à des données protégées. Elle peut se faire simplement par identifiant / mot de passe ou avec des systèmes plus sophistiqués (certificats numériques, tickets Kerberos, etc.).

Par ailleurs, pour assurer la confidentialité des flux échangés, le fournisseur de service pourra utiliser SSL, ou bien des méthodes plus sophistiquées avec WS-Security. Cette norme décrit comment crypter et signer des documents XML transportés via SOAP. Plus généralement, il est possible de décrire l'ensemble de la politique de sécurité appliquée (règles sur l'authentification et la confidentialité) dans un fichier au format WS-SecurityPolicy.

#### Outils pour le pattern Web 2.0

L'approche REST (3) repose sur deux principes :

- un service REST est un service d'accès à une ressource informationnelle identifiée par une adresse Internet (URL). Exemples de ressource : news d'un site web, entrée d'un blog, page d'un wiki...
- invoquer un service REST, c'est-à-dire accéder à la ressource, c'est utiliser directement le protocole HTTP

HTTP permet d'envoyer à un serveur web des requêtes définies par des mots clefs : GET, POST, PUT, DELETE, TRACE... Appliquer le pattern REST consiste essentiellement à définir pour chacun des mots clefs HTTP une sémantique spéciale.

Illustrons cela en montrant comment un serveur de contenu supportant ATOM, protocole couramment utilisé dans le monde de la gestion de contenu web, définit les services qu'il offre à ses clients. ATOM manipule deux entités de base, les entrées (ATOMEntry) et les flux (ATOMFeed). Une entrée décrit une information, un flux est associé à l'ensemble des informations rassemblées sous un même toit.

## Conclusion (provisoire)

L'avantage de l'approche REST est sa (relative) simplicité. Le succès de la blogosphère, qui repose en partie sur ATOM, a d'ailleurs propulsé cette approche comme un concurrent potentiel des Web Services.

Mais ce n'est pas si simple ! Il ne faut pas oublier que ce type de protocole manipule avant tout des informations pas ou peu structurées. Dans le monde des SI de gestion, manipulant des informations structurées, cela peut poser des problèmes :

- comment récupérer non seulement une commande, mais également les lignes de cette commande, la description du produit commandé par ligne, le ou les bons de livraison, etc. ? Avec REST, il faudra faire autant de demande qu'il y aura de ressource concernée, et de plus il faudra utiliser les ATOM-Feed dans un contexte non prévu au départ.
- Comment rechercher non pas les " x dernières commandes ", mais les commandes " reçues entre telle et telle date et non encore livrées " : autrement dit, comment mettre en œuvre - simplement - des critères de recherche ?
- comment demander un devis, c'est à dire la valorisation d'une commande ? L'approche REST est orientée " Données " et non pas " Traitement ", or les services SOA ne sont pas uniquement orientés Données.
- Comment garantir sécurité, robustesse et performance ?

En conclusion, quel est le futur de SOA ? En ce qui concerne l'ouverture du SI (les patterns 1 et 3), on pariera sur un rapprochement des approches WS-\* et REST/ATOM pour répondre à certains besoins précis. La disparition de l'approche WS-\* au profit de l'approche REST paraît en revanche une annonce nettement prématurée.

Quand à l'approche " les services sans les web services " illustrée par le pattern 2, le combat entre SPRING 2.x et les EJB 3 est encore indécis à long terme. Mais l'approche

	Pattern 1 : " ouvrir le SI(ie) "	Pattern 2 : " modulariser le SI "	Pattern 3 : " Web 2.0 "
Client d'un service conçu via le pattern	Système d'information des partenaires de l'entreprise	Application composite (processus BPEL, application interactive MVC, batch)	Interne, via des applications de type Mashup ou Widgets
Caractéristique du service	Service " gros grain ", distribué, avec une garantie de sécurité et de robustesse, et éventuellement de performance	Service " gros grain " et " grain moyen ". La distribution doit être possible mais elle n'est pas nécessaire pour les services métier. La sécurisation n'est pas indispensable (dialogue client / service interne au SI)	Service de type " flux d'information " (RSS,Atom)
Modalité d'accès au service	Synchrone lorsque le service est accédé via Internet. Asynchrone lorsque le service ouvre un silo (cf. figure 2).	Synchrone lorsqu'il s'agit d'un service métier ou d'un service technique. Asynchrone lorsqu'il s'agit d'un processus.	Synchrone
Format de description du service et des opérations qu'il offre	Norme WSDL	Interfaces JAVA ou C#	Standard de fait REST (formalisé via ATOM par exemple)
Format des informations échangées entre client et service	Document XML (respectant un schéma de référence) associé à des informations structurées (objets métier)	Objets (éventuellement sérialisés)	Document XML associé à des informations non structurées : pages XHTML, document PDF...
Protocole de communication réseau	SOAP sur HTTP ou autre	Les services sont co-localisés : pas de distribution a priori. En cas de nécessité on utilise RMI (via des EJB Session) ou JMS (via des MDB (1))	HTTP " de base "
Conteneur de service	Axis, Xfire	Spring... Container EJB 3 (avec ou sans distribution)	Toute application sachant répondre à des requêtes REST (questionnaire de blog respectant ATOM par exemple)
Exemple d'utilisation du pattern architectural	Relation automatisée entre le SI d'une banque distribuant des produits d'assurance, et le SI de l'assureur assurant la gestion back office.	Toute nouvelle application " orientée objet " peut être structurée ainsi : ce pattern est une évolution des architectures objet n. tiers.	Un service est accédé via une petite application dédiée, installée dans un " container " dédié : Widget de Nevibes (UJWA), Gadgets de Google Desktop...

Service ATOM	Mot clef http	Sémantique	Résultat
Créer une information	POST	Permet de créer une nouvelle information, en utilisant une requête POST contenant un ATOMEntry	Une URL sur l'information nouvelle créée sur le serveur
Editer une information	GET PUT DELETE	Rechercher l'information Mettre à jour l'information Supprimer l'information	Un ATOMEntry Un status HTTP (ex : 200 OK) Un status HTTP
Rechercher des informations	GET	Interroge le serveur pour récupérer les derniers informations créés.	Un ATOMFeed, contenant les informations demandées en particulier les x derniers documents créés, plus un lien vers les y derniers documents créés

SPRING par sa simplicité est en tout cas à court et moyen terme un cheval de bataille pragmatique et désormais éprouvé.



**Pascal Grojean,**  
Directeur  
de Sqli Consulting

- (1) Message Driven Bean
- (2) Service Level Agreement
- (3) REST = Representational State Transfer. D'après Roy Fielding, le créateur de cet acronyme, "REST is intended to evoke an image of how a well-designed Web application behaves: a network of web pages (a virtual state-machine), where the user progresses through an application by selecting links (state transitions), resulting in the next page (representing the next state of the application) being transferred to the user and rendered for their use".

# UML pour la cartographie et SOA

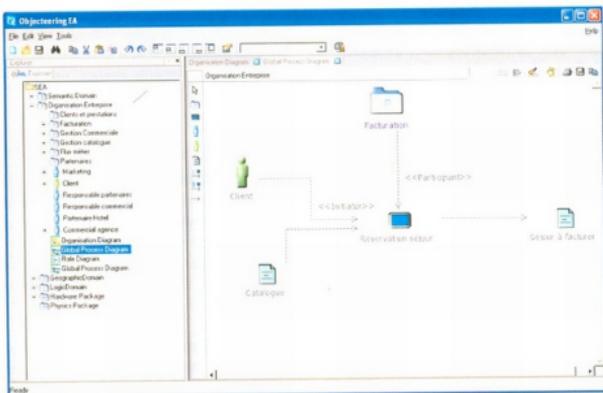
Objecteering 6 supporte la modélisation d'architecture d'entreprise orientée services en vous guidant à travers une démarche dédiée : la démarche SEA.

Comment prendre en compte l'intégralité des aspects d'un système d'information et tirer parti des mécanismes de la SOA, en s'appuyant sur une démarche moderne appuyée sur les standards ?

L'approche SEA (Services oriented Enterprise Architecture) adresse tous les intervenants dans la construction d'un système d'information, en fournissant des vues dédiées MOA (maîtrise d'ouvrage) et des vues dédiées MOE (maîtrise d'œuvre). SEA fournit une démarche pour SOA en s'appuyant sur les standards et technologies les plus récents de modélisation (UML2, MDA) et en apportant des extensions adaptées. SEA s'appuie sur les techniques de modélisation d'architecture d'entreprise, et reprend les apports méthodologiques de l'initiative ouverte Praxeme ([www.praxeme.fr](http://www.praxeme.fr)).

A travers la notion d'aspect, Praxeme adresse les domaines de préoccupation spécifiques de chaque intervenant : les données métier, l'organisation et les processus métier, l'architecture fonctionnelle, l'architecture applicative, l'architecture technique et les vues logicielles détaillées.

En s'appuyant sur les standards et les pratiques internationalement reconnues, SEA permet une continuité des modèles depuis la MOA jusqu'à la MOE, pour ensuite bénéficier des facultés de production automatisée de code depuis UML en phase de réalisation, et de la puissance de la technologie MDA.



## Objecteering EA

### L'outil support de SEA pour chaque type d'intervenant

Objecteering EA est composé de Objecteering 6 UML Modeler et du module d'extension SEA. Objecteering EA supporte la modélisation de la démarche SEA, en s'appuyant sur le standard UML et des extensions adaptées (profil SEA). Il procure les modèles dédiés à chaque type de problème et d'intervenant (les aspects Praxeme), assiste la cohérence entre ces modèles, tout en vous faisant bénéficier des avantages fonctionnels de l'atelier Objecteering (UML, MDA, cohérence des modèles, travail de groupe, automatisation du développement, etc.)

Les services additionnels de l'atelier, comme par exemple Objecteering Requirements pour la gestion des exigences intégrée à la modélisation, renforcent les possibilités de Objecteering EA en modélisant les exigences de manière synchrone et cohérente avec le modèle. Les générations de code vers les cibles Java, C#, C++ ou SQL, supportant des frameworks tels que Hibernate (mapping Objet/Relationnel) apportent des gains substantiels en qualité et en productivité pour le développement de l'applicatif et la gestion des données. La génération de documentation paramétrable de l'atelier permet de fournir des documents du modèle dédiés à chaque type d'acteur sur le système d'information. Objecteering EA intègre l'ensemble de ces modèles dédiés à chaque

type d'acteur au sein d'un référentiel unique, garant de la cohérence, de la traçabilité et de la non redondance des informations.

Objecteering EA masque la complexité d'UML pour n'offrir à chaque type d'usager que la partie qui le concerne, en utilisant une notation intuitive et illustrative dédiée. Bien qu'appuyé sur le standard, chaque intervenant dispose d'une notation simplifiée, pour offrir ensuite la vision UML nécessaire en phase de développement. Les travaux guidés par le modèle guident et automatisent la modélisation, tout en assurant la cohérence entre chacune des vues.

## Objecteering

SOFTWARE

Objecteering Software, éditeur de l'atelier Objecteering 6, est le spécialiste français UML/MDA pour le développement d'applications guidé par le modèle.

SA suite d'outils couvre le cycle de vie de la gestion des exigences jusqu'au déploiement de l'application pour les cibles Java/J2ee, C#.Net, C++, SQL, Corba et Fortran.

Pour plus d'informations :

[www.objecteering.com](http://www.objecteering.com)

[info@objecteering.com](mailto:info@objecteering.com)

Tél. : 01 30 12 16 60



(1) Le livre blanc sur l'approche SEA est téléchargeable à l'adresse : [http://www.softteam.fr/pdf/fr/SEA\\_architecture\\_d\\_entreprise\\_orientee\\_service.pdf](http://www.softteam.fr/pdf/fr/SEA_architecture_d_entreprise_orientee_service.pdf)

## Panorama des outils orientés SOA

*Avec la multiplication des offres SOA, difficile de se faire une idée précise du marché entre les players, les pure players et les global players. Nous vous proposons quelques pistes.*

Les outils de base sont aujourd'hui : un outil de développement, un référentiel, un ESB, un moteur de règle, un environnement de gestion ou de gouvernance, des fonctions de tests et de monitoring, une gestion d'identité et d'orchestration. Ces solutions peuvent être totalement intégrées ou être un ensemble de différents outils. À vous de vérifier la compatibilité et l'interopérabilité entre ces outils.

**BEA l'offre 360** : SOA 360 est sans doute l'offre la plus homogène et une des plus complètes du marché (incluant WebLogic, AquaLogic, Tuxedo). Elle s'appuie sur les micro services (MSA) offrant une souplesse d'utilisation peu commune. On y trouvera des mécanismes de gouvernance, de gestion, d'orchestration, de développement. Le cœur de la solution est l'outil de collaboration : Workspace 360.

**Software AG + WebMethod = CrossVision** : CrossVision est une suite d'outils couvrant l'ensemble des besoins, incluant des modules de gouvernance, de composition d'applications composites, de BPM, des fonctions d'intégration de l'infrastructure applicative existante, des outils de modélisation et d'orchestration. Également inclus, un référentiel unique. L'éditeur vient de racheter WebMethod.

**IBM SOA Foundation** : IBM ne propose pas une unique solution SOA mais plusieurs entités. On retrouve ainsi SOA Foundation regroupant l'ensemble des outils IBM liés aux architectures SOA : modélisation, processus métiers, serveur, monitoring, ESB... SCA est pris en compte dans certains outils de la gamme. Pour étendre le cycle de vie de SOA Foundation, on dispose de SOA Governance, lié à la gouvernance des architectures de services. L'éditeur mise aussi sur la qualité des services avec Rational Tester for SOA Quality et Rational Performance Tester Extension for SOA Quality.

**Sun Java CAPS** : L'environnement phare de Sun est Java CAPS. Cette solution reprend les

outils SeeBeyond (éditeur racheté par Sun). Cette suite logicielle intègre un référentiel, des designers, des outils de gestion, une partie développement (SeeBeyond eVision Studio). On dispose aussi d'un module BPM / BPEL.

**Oracle SOA Suite** : SOA Suite fait partie de la grande famille Oracle Fusion. Cette suite permet de développer, exposer, composer, consommer, orchestrer des services et SOA. On dispose aussi d'un module de processus métiers (BPM / BPEL). On dispose aussi d'un ESB.

**Red Hat JBoss JEMS** : Connu pour son serveur d'application, JBoss dispose aussi d'une solide offre open source pour le middleware d'entreprise avec JBoss JEMS. JEMS constitue une fondation SOA comprenant tous les outils pour créer son middleware et mettre en place des services. Il s'est doté d'un moteur de règles métiers et surtout il dispose d'un ESB.

**TIBCO ActiveMatrix et BusinessWorks** : L'éditeur Tibco propose une gamme SOA autour d'ActiveMatrix et de BusinessWorks. BusinessWorks est une plate-forme d'intégration Il sert aussi de bus d'entreprise (ESB). La gamme ActiveMatrix comprend ActiveMatrix Registry pour la découverte, le catalogue et l'exposition des services ainsi qu'ActiveMatrix Service Grid pour créer des services et des applications composites en .Net, Java et BPEL.

**Microsoft Biztalk et .Net** : Pour l'éditeur, la SOA est un impératif du système d'information. L'éditeur s'appuie sur le framework .Net et en particulier sur le .Net 3.0 qui représente

les bases d'une SOA avec le Workflow Foundation et le Communication Foundation. L'outil phare SOA de Microsoft est Biztalk Server.

**Rogue Wave Hydra** : Rogue Wave, connu pour son expertise en C++, s'adresse aux services web et aux architectures SOA avec HydraSuite et en particulier avec les outils HydraSCA, HydraSDO et HydraExpress. Il est possible de construire des SOA avec des web services C++, Java ou C#. C'est avant tout un environnement de conception, de développement et de déploiement.

**Xcalia pour les spécifications SOA** : L'éditeur français mise beaucoup sur les spécifications SCA et SDO. L'éditeur propose deux outils phares pour faire de la SOA : Intermediation, un composant SOA se plaçant entre les données et les aspects métiers afin d'aligner le processus métier avec les applications et Xcalia Studio, un outil de mapping graphique.

**HP Systinet 2** : Outil de gouvernance SOA, Systinet offre un environnement de gestion très complet que ce soit pour la gouvernance, le référentiel, l'annuaire, l'exposition et la consommation de services. Permet de mettre en place des politiques d'accès entre les utilisateurs et les fournisseurs de services.

**Objecteering la SOA et la SEA** : L'éditeur Objecteering propose de modéliser son système d'information avec son outil de modélisation éponyme, dans une approche SEA (Service Enterprise Architecture). Cela se fait grâce à des profils UML spécifiques. Il propose un référentiel de modélisation interne. La SEA s'intègre parfaitement dans une démarche SOA.

### Se former à SOA

- **Softeam** : société de conseil et de formation. Softeam propose des cursus SOA très complets ainsi que sur la modélisation, MDA, Java EE, les web services.
- **SQI** : le groupe propose une synthèse de 2 jours sur le sujet en abordant l'ensemble des aspects SOA.
- **Capgemini** : propose une formation sur l'état de l'art de la SOA.
- **Unilog** : une offre très complète sur le sujet (conception, l'essentiel, les ESB, etc.)

## Le débogage avec Visual Studio

Visual Studio propose de nombreuses fonctionnalités de débogage avancées, nous allons voir dans cet article les possibilités offertes par cet éditeur.

Une opération de debug consiste à analyser le fonctionnement d'un programme à partir de son code source en inspectant le contenu des variables à un instant t.

### Le mode debug de Visual studio : les points d'arrêts

Pour pouvoir déboguer un programme, il faut lancer celui-ci en mode debug et utiliser les points d'arrêts qui permettent d'arrêter l'exécution d'un programme sur cette ligne. Pour rajouter un point d'arrêt, le plus simple est de cliquer dans la marge de gauche de votre code ou d'appuyer sur la touche F9, ensuite lancez votre programme en appuyant sur la touche F5. A partir de maintenant, vous avez accès à toutes les fonctionnalités de debug offertes par Visual Studio. Une nouvelle barre d'outils est apparue dans Visual Studio.



Le premier bouton (touche F5) permet de reprendre l'exécution du code, le second arrête l'exécution du code à l'endroit où il se trouve, le suivant stoppe le programme et le quatrième le relance. La flèche jaune permet d'afficher la ligne qui va être exécutée, celle-ci est surlignée en jaune dans Visual Studio. Les 3 boutons qui suivent sont très intéressants, puisqu'ils permettent d'exécuter du code en pas à pas, afin de savoir exactement par où notre code passe. Le premier bouton va à l'instruction suivante, dans notre exemple il ira dans le constructeur de l'objet Personne, le bouton suivant va à la prochaine exécution de la même fonction, dans notre cas le débogueur ne s'arrêtera pas dans le constructeur de l'objet Personne, mais ira directement à la ligne 14, le troisième bouton retourne à la fonction appelée, dans notre cas cela arrête le débogueur puisque la fonction Main est la première méthode appelée.

### Inspection et modification de variables

Le mode debug de Visual Studio permet de consulter et modifier le contenu des variables.

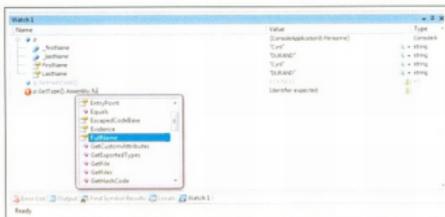
En survolant une variable on voit apparaître un tooltip qui affiche les différentes variables, si votre objet contient des propriétés, le débogueur vous permet de les afficher et même de les modifier.

Par exemple, j'ai créé une instance de l'objet Personne nommé "Cyril DURAND" que j'ai renommé en "Maud", le reste de l'exécution du programme travaillera avec "Maud" plutôt que "Cyril".

### Les fenêtres d'aide au debug

Il existe de nombreuses fenêtres d'aide au debug dans Visual Studio. La fenêtre Locals permet de consulter les différentes variables accessibles dans le contexte local.

La fenêtre Watch permet aussi de consulter le contenu des variables mais permet en plus d'appeler des méthodes :



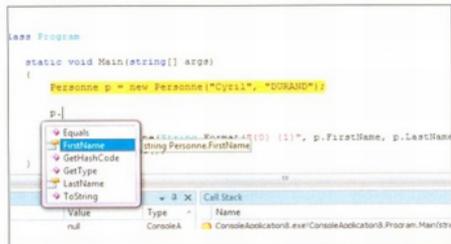
On peut noter que l'on possède l'IntelliSense partout dans Visual Studio ce qui est un plus non négligeable.

La fenêtre Call Stack permet de connaître la pile des méthodes appelées, en double cliquant sur l'une d'elles, on remonte la pile et on peut consulter le contenu des variables à l'instant où la fonction a été appelée.



### Modification du code à la volée

Visual Studio permet la modification des lignes de code tout en débogant :



Une autre fonctionnalité moins connue est la possibilité de modifier l'ordre d'exécution du code, en déplaçant la flèche jaune à gauche vers une autre ligne de code, l'exécution du programme reprend à cet endroit.

### Conclusion

Nous avons vu quelques-unes des fonctionnalités pour déboguer une application avec Visual Studio 2005, il existe de nombreuses autres astuces comme les Visualizer, les logs de debug, l'attribut DebuggerDisplayAttributes et bien d'autres choses encore... Nous y reviendrons prochainement.

### ■ Cyril DURAND

Membre de l'équipe de développement CodeS-Source5  
<http://blogs.CodeS-Source5.com/cyril/>

## Ma première application Apollo

Apollo est le nom de code d'un nouveau runtime universel publié en version alpha sur le laboratoire d'Adobe ([labs.adobe.com](http://labs.adobe.com)). Il propose d'étendre les capacités de ses applications web en dehors du navigateur et de les transformer ainsi en applications riches bureautiques.

Dans l'exemple qui suit, nous allons développer une application riche avec le SDK gratuit Flex 2, et la transformer en client riche sur le bureau à l'aide du SDK Apollo. L'objectif est de personnaliser le chrome pour que le look de ma fenêtre ne dépende pas de l'OS.

### Développement de l'application avec le SDK

Le fichier myApolloWindow.mxml décrit en XML la mise en forme et le comportement de mon application. L'application est constituée de 5 composants : TitleWindow, Button, HTML, Image et HSlider.



(Fig. 1)



(Fig. 2)

```
<mx:TitleWindow close="onClose()" showCloseButton="true" title="En attendant la beta...">
  <mx:Button label="Minimise moi" click="onMinimize()"/>
  <mx:HTML location="http://labs.adobe.com/" scaleX="{zooming.value}" scaleY="{zooming.value}"/>
</mx:TitleWindow>
<mx:Image id="monImage" source="assets/logoApollo.png"/>
<mx:HSlider snapInterval="0.1" liveDragging="true" id="zooming" minimum="0.2" value="1" maximum="1.5"/>
```

Le composant TitleWindow permet nativement d'afficher un bouton de fermeture de fenêtre (showCloseButton). Nous verrons ensuite comment se servir de l'événement Close pour quitter l'application Apollo. Nous utiliserons aussi l'événement click sur le bouton pour minimiser l'application dans la barre des tâches. Le composant Image affiche le logo d'Apollo (PNG transparent). Le composant HTML ne fait pas partie du SDK Flex, c'est une nouveauté du SDK Apollo. Il utilise le navigateur webKit (projet OpenSource) embarqué dans le runtime Apollo pour afficher et interagir avec un site web. Ici le composant va afficher la page d'accueil de labs d'Adobe. Son échelle sera directement liée à la valeur renvoyée par le composant HSlider nommé ici " zooming ". Nous pourrions donc zoomer librement sur la page. Les accolades de {zooming.value} indiquent que cette propriété est " bindable ". Cela va donc automatiquement rajouter un listener sur la propriété " value " du composant " zooming " et ainsi modifier en live l'échelle de la page web.

### Ajout du code ActionScript 3

ActionScript 3 est le langage de script de la nouvelle machine virtuelle du Flash Player 9 et du coup, du runtime Apollo. Standard ECMAScript 262, l'ActionScript 3 est un langage objet très simple à prendre en main. Je vais ici rajouter du code pour gérer la minimisation, la fermeture de l'application et le déplacement de la fenêtre. Pour ce faire, je rajoute une balise <mx:Script> et déclare quelques fonctions.

```
private function onMinimize():void
{
    stage.window.minimize();
}
```

```
private function onClose():void
{
    stage.window.close();
}
```

La même classe propose une méthode close. La fonction onClose est appelée par l'événement " close " du composant TitleWindow.

```
private function initApp():void{
    this.monImage.addEventListener(MouseEvent.CLICK, onMouseDown);
}
private function onMouseDown(evt:MouseEvent):void
{
    stage.window.startMove();
}
```

Pour gérer le déplacement de la fenêtre, vous pouvez librement décider de la zone graphique qui autorisera le " glisser-déposer " de l'application. Très souvent, on utilise l'en-tête de la fenêtre, dans cet exemple je vais désigner l'image (le logo Apollo) comme zone qui permettra le déplacement. Je rajoute donc un listener sur le composant Image qui surveille le clic de souris (MOUSE\_DOWN) et exécutera alors la fonction onMouseDown dans laquelle j'appelle la méthode startMove.

Pour exécuter la fonction initApp(), je complète la balise <mx:Application> avec l'événement creationComplete :

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" height="312" width="445" creationComplete="initApp"/>
```

### Création d'une fenêtre transparente

Si on compile à ce stade mon application avec le SDK Apollo, elle s'exécute dans une fenêtre qui est liée à l'OS.(Fig.1). Pour personnaliser le chrome de l'application, on doit la rendre transparente et du coup, indépendante de l'OS. Première étape, je vais modifier le style global de

mon application. On pourrait soit charger une feuille de styles CSS, soit directement rajouter une balise `<mx:Style>` dans mon fichier mxml :

```
<mx:Style>
  Application
  {
    background-color:"";
    background-image:"";
    padding: 0px;
  }
</mx:Style>
```

Cette application d'une feuille de styles n'est pas suffisante. Il faut indiquer au runtime qu'il s'agit d'une application transparente. Pour chaque application Apollo, vous devez renseigner un manifeste XML qui décrit l'application et ses propriétés. Outre la saisie de l'auteur, de la description ou de l'icône de l'application, une partie est réservée au chrome. Pour rendre l'application transparente, voici les paramètres à renseigner :

```
<rootContent systemChrome="none" transparent="true" visible="true">
```

La valeur par défaut de l'attribut `systemChrome` est "standard", ce qui ouvre une application Apollo dans une fenêtre avec des éléments spécifiques à l'OS (barre de titre, boutons pour minimiser ou fermer). Le fait de passer l'attribut à "none" et d'activer la transparence nous rend responsable de ces éléments. Après compilation, l'application est bien transparente. (Fig. 2) Cette petite application a été réalisée avec moins de 40 lignes de code. Vous pouvez développer une application Apollo à partir de vos applications Flex (comme cet exemple), vos applications Flash (un swf suffit) ou web (HTML et JavaScript). Le SDK d'Apollo permet aussi de lire/écrire dans des fichiers sur son poste, de gérer le fait d'être connecté ou non à Internet ou de gérer la communication entre plusieurs fenêtres. De quoi étendre les capacités de nos applications web et innover.



■ Michaël CHAIZE

Consultant Avant-Vente Adobe

## Guide du développeur Apollo

Le kit du développeur Apollo se trouve sur le site [labs.adobe.com](http://labs.adobe.com). Adobe y poste régulièrement les versions alpha et bêta de ses logiciels plusieurs mois avant les dates de sorties officielles. Une version bêta d'Apollo a été annoncée pour cet été et la version finale sera disponible publiquement avant la fin de l'année. Rendez vous sur ce lien : <http://labs.adobe.com/technologies/apollo/> et cliquez sur "Get the Apollo SDK and Developer downloads".

En premier lieu, il faut télécharger et installer le runtime Apollo. Il est pour l'instant disponible en alpha sur Windows (6Mo) et Mac (8Mo) et sera à terme porté sur Linux.

Ensuite, il faut récupérer le SDK Apollo qui permet de compiler les applications. Une documentation détaillée est disponible sur cette même page ainsi que quelques exemples avec les sources commentées.

Les développeurs Flex qui souhaitent réaliser en quelques secondes une application Apollo peuvent télécharger et installer le plug-in pour Flex Builder 2.0.1 (environnement de développement de Flex basé sur Eclipse).

## Code source de l'application

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
  layout="absolute" height="312" width="445" creationComplete="initApp()">
  <mx:Script>
  <![CDATA[
    private function initApp():void{
      this.monImage.addEventListener(MouseEvent.CLICK,onMouseDown);
    }
    private function onMouseDown(evt:MouseEvent):void
    {
      stage.window.startMove();
    }
    private function onMinimize():void
    {
      stage.window.minimize();
    }
    private function onClose():void
    {
      stage.window.close();
    }
  ]]>
</mx:Script>

<mx:Style>
  Application
  {
    background-color:"";
    background-image:"";
    padding: 0px;
  }
</mx:Style>

<mx:TitleWindow id="maFenetre" close="onClose()" x="112" y="12"
  width="319" height="275" layout="absolute" showCloseButton="true"
  title="En attendant la beta..." borderColor="#ff0000" color="#ffffff">

  <mx:Button label="Minimise moi" right="10" top="10" click="onMinimize()" color="#000000"/>

  <mx:HTML location="http://labs.adobe.com/" scaleX="{zooming.value}"
  scaleY="{zooming.value}" left="5" top="35" right="5" bottom="5"/>

</mx:TitleWindow>

<mx:Image id="monImage" x="0" y="10" source="assets/logoApollo.png"/>

<mx:HSlider snapInterval="0.1" liveDragging="true" id="zooming" x="142"
y="58" minimum="0.2" value="1" maximum="1.5"/>

</mx:Application>
```

## Construire un composant "glisser - déposer" avec Google Web Toolkit

Le Google Web Toolkit (GWT) est la plate-forme de développement d'applications Ajax proposée par Google. Elle repose sur un principe simple et révolutionnaire : développer, tester, déboguer une application que l'on écrit en Java, et la déployer ensuite, en la traduisant – pour sa partie cliente – en JavaScript. Mais ce n'est pas tout.

Le paradigme de programmation de GWT est proche de celui utilisé traditionnellement dans des applications clients lourds : on construit l'interface en combinant des objets d'IHM appelés "Widget". Ces "Widgets" émettent des événements (clic, sélection, modification, etc.) qui peuvent être interceptés et traités par des méthodes réflexes. GWT propose un jeu de Widgets standard, assez frustré au demeurant. Il permet surtout de construire facilement de nouveaux types de Widgets.

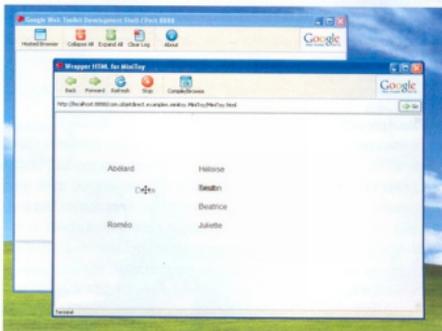
Autre point : le GWT ne répudie pas JavaScript. Il est possible – et même facile – d'intégrer du JavaScript dans une application GWT. Ce code n'aura pas, bien sûr, besoin d'être ensuite traduit. Inconvénient : en phase de développement, il échappe à la sagacité du compilateur et du débogueur de votre IDE préféré. Nous allons voir que cette faculté est particulièrement intéressante pour intégrer des bibliothèques JavaScript telles que Dojo, Prototype, Script.acu.lus.

### Intégrer une bibliothèque JavaScript

Combinons tout cela pour enrichir le GWT d'une nouvelle Widget qui apporte une fonctionnalité manquante dans la version de base : le glisser-déposer (Drag and Drop). Nous allons pour cela utiliser la bibliothèque Yahoo. Pourquoi Yahoo ? Parce que cette bibliothèque fonctionne bien, qu'elle est bien documentée et qu'elle propose un panel de fonctionnalités riches et modulables.

La première chose à faire est d'associer les fichiers de scripting Yahoo à notre application GWT, afin que les scripts de Yahoo soient chargés sur le navigateur de l'utilisateur avec la partie cliente du module GWT demandé. Avant d'expliquer plus en détail la marche à suivre, rappelons succinctement quelle est la structure d'un projet GWT :

- Une application GWT est divisée en "modules". Un module correspond à une page animée. C'est l'équivalent AJAX d'une "applet".
- Les sources et ressources associées à un module sont regroupées au sein d'une même arborescence de packages, par exemple : "com.objetdirect.exemples.minityou". Cette arborescence est la "racine" du module.
- Cette racine doit contenir trois sous-packages aux noms définis par GWT : "public", "client" et "server".
- Le sous-package "public" contient les ressources associées au module : feuille CSS, images, scripts.
- Le sous package "client" contient le code de la partie cliente du module. Ce sont les sources JAVA qui seront traduits in fine en JavaScript.
- Le sous package "server" contient le code de la partie serveur du module. Ce sont les sources JAVA qui doivent s'exécuter sur le serveur Web, en réponse aux requêtes émises par la partie cliente. Elles



ne sont pas traduites en JavaScript.

- Le module est configuré par un fichier XML appelé <nom-module>.gwt.xml. Ce fichier est placé dans l'arborescence racine du module ("com.objetdirect.exemples.minityou" dans notre cas).

Pour intégrer les bibliothèques JavaScript Yahoo qui implémentent le glisser-déposer, il faut d'abord déposer les scripts Yahoo idoines dans le répertoire "public" de notre module. Il s'agit, des fichiers yahoo.js, dom.js, event.js, dragdrop.js et DDPlayer.js :

- yahoo.js, dom.js, event.js contiennent les routines qui forment le cœur de la bibliothèque proposée par Yahoo. Ces scripts sont utilisés par les différentes fonctionnalités de cette bibliothèque (glisser-déposer, animation, composants, etc.)
- dragdrop.js contient l'implémentation de la fonctionnalité "glisser-déposer".
- DDPlayer.js est un exemple proposé par Yahoo qui implémente une fonctionnalité générique : celle d'associer des zones de dépôt à des objets glissants. Elle peut être utilisée en l'état. Je me demande pourquoi Yahoo ne l'a pas mise dans le cœur de sa bibliothèque.

Il faut ensuite déclarer ces scripts dans le fichier de configuration du module, c'est-à-dire le fichier <nom-module>.gwt.xml. Pour cela, on rajoute, pour chaque fichier script, une section "script" au sein de la balise XML racine ("module"), comme suit :

```
<module>
...
<script src="yahoo.js"><![CDATA[
```

```

if ($wnd.yahoolsOk)
    return true;
else
    return false;
}]>
</script>
...
<module>

```

Cette section indique que le module a besoin du script " yahoo.js ". Il contient aussi le morceau de JavaScript qui s'assure que le script a été complètement chargé. Attention, ce n'est pas un gadget. Ignorer ce " détail " mène généralement à un comportement incohérent du module GWT. Il semble en effet que les scripts sont chargés de manière asynchrone par le module, de façon concurrente avec son propre démarrage. Bien sûr, Yahoo n'a pas prévu que sa librairie puisse être intégrée dans la plate-forme de son grand concurrent. Il n'a donc pas prévu de méthode pour s'assurer du bon chargement de son script. C'est à nous, pauvres développeurs de " terminer " le travail en rajoutant à l'extrême fin du fichier yahoo.js la définition suivante :

```

function yahoolsOk() {
}

```

La snippet JavaScript contenue dans le fichier .gwt.xml teste seulement l'existence de la fonction " yahoolsOk ". L'implémentation de cette fonction n'a donc aucun intérêt. C'est pourquoi nous la laissons à vide. J'admets que devoir modifier des composants fournis par un tiers n'est pas sans inconvénient. Mais c'est la seule solution que je connaisse qui soit à la fois simple et sûre. Se reposer sur la dernière définition que contient le fichier yahoo.js nous expose de toute façon à devoir inspecter ce fichier à chaque nouvelle importation de la dernière version de la bibliothèque Yahoo. Il faut bien sûr définir une méthode différente pour chaque script : yahoolsOk pour yahoo.js, dragdroplsOk pour dragdrop.js, et ainsi de suite.

### Invoker du JavaScript au sein du code Java

Désormais, les scripts dont nous avons besoin sont disponibles au sein de notre module. Mais ils restent en JavaScript. Il n'y a pas de traduction inverse JavaScript vers Java ! Si nous voulons faire appel aux routines de ces scripts, il nous faut invoquer du JavaScript. Heureusement, GWT nous propose un mécanisme simple pour le faire. C'est JSNI (comme JavaScript Native Interface), un dérivé GWT du mécanisme standard de JNI (Java Native Interface). Voici un exemple illustrant JSNI :

```

public native static void setLocation(
    Element e,
    int leftPos, int topPos)
/*-{
    $wnd.YAHOO.util.Dom.setStyle(e, 'left', leftPos+'px');
    $wnd.YAHOO.util.Dom.setStyle(e, 'top', topPos+'px');
}*/;

```

La méthode setLocation est une méthode Java déclarée comme étant " native ", c'est-à-dire que son implémentation n'est pas donnée en

Java. Ici, on est encore dans JNI. Cette implémentation est fournie sous la forme d'un commentaire particulier, définie par les balises " /\*-{ " et " }-\*/ ". Le contenu de ce commentaire est le JavaScript qui devra être exécuté lorsque setLocation est invoquée. Ici on est dans JSNI. Ces commentaires ne sont reconnus que par GWT. N'essayez pas de les utiliser sur d'autres plates-formes !

Voilà. C'est tout simple. Il n'y a rien d'autre à faire : pas de compilateur à lancer, de script ant à exécuter. Notez aussi que les paramètres passés à notre méthode Java sont utilisables au sein de notre code JavaScript. Le fait que setLocation soit statique n'est pas une obligation. Nous aurions pu tout aussi bien la définir comme méthode d'instance. Si vous inspectez le code de dom.js, vous trouverez la routine " YAHOO.util.Dom.setStyle ". Cette méthode est définie au sein de la fenêtre courante. Cette fenêtre est accessible au sein d'une routine JSNI via la variable standard " \$wnd ". N'oubliez pas de préférer tous vos appels aux routines Yahoo par cette référence.

### La première version de notre composant

Nous allons développer un composant de type panel (une widget conteneur d'autres widgets) qui contient des composants capables de glisser sur le panel. Pour cela, il suffit de créer une classe JAVA dérivant d'une classe Panel du Google Web Toolkit. Nous allons utiliser ici la classe AbsolutePanel qui crée un panel permettant un positionnement absolu des widgets filles, c'est-à-dire que l'on doit préciser la position en x, y de chacune d'elles.

La voici dans son intégralité, le code définissant notre panel " glisser déposer " :

```

public class SimpleDragAndDropPanel extends AbsolutePanel
{
    public void addDraggableWidget(
        Widget widget,
        int left, int top,
        String affordance)
    {
        add(widget, left, top);
        addDraggableSlot(widget.getElement(), affordance, counter++);
    }

    public void addTargetWidget(
        Widget widget,
        int left, int top,
        String affordance)
    {
        add(widget, left, top);
        addTargetSlot(widget.getElement(), affordance, counter++);
    }

    public static native void addDraggableSlot(
        Element e,
        String affordance,
        int counter)
    /*-{
        e.id = 'drag_'+counter;

```

```
new $wnd.YAHOO.example.DDPlayer(e.id, affordance);
}*/;

public static native void addTargetSlot(
    Element e,
    String affordance,
    int counter)
/*{
    e.id = 'target_'+counter;
    new $wnd.YAHOO.util.DDTarget(e.id, affordance);
}*/;

int counter;
}
```

Simple n'est ce pas ? Qui a dit que le GWT ne dispose pas d'un composant " glisser déposer " ?

On réutilise bien sûr l'essentiel des fonctionnalités offertes par AbsolutePanel. La seule action supplémentaire de notre classe est d'associer à l'avatar JavaScript de chaque widget (une instance de la classe Element), un objet Yahoo qui en gère le déplacement. Si on associe un objet de type YAHOO.example.DDPlayer la widget pourra être glissée. Si on lui associe un objet de type YAHOO.util.DDTarget, elle pourra être la cible d'un dépôt.

L'avatar JavaScript d'une widget est obtenu par la méthode getElement de la Widget. C'est un objet qui peut être utilisé dans le code JavaScript (alors que le type widget est inconnu de JavaScript).

Notre classe panel permet aussi de préciser la compatibilité entre widgets glissantes et zone de dépôt. Il s'agit du paramètre " affordance ". Pour qu'une widget glissante puisse être déposée sur une widget zone de dépôt (ou cible), il faut qu'elles partagent la même " affordance ". Dans le cas contraire, la widget glissante revient automatiquement sur sa position de départ.

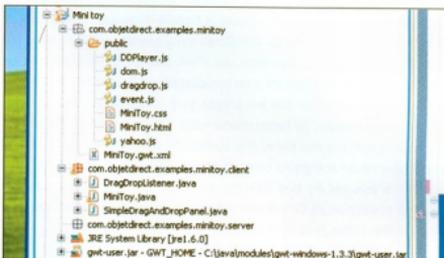
Notez que notre composant n'oblige pas à affecter comme position initiale d'une widget glissante, la position d'une widget de dépôt. Ensuite, elle ne peut se déplacer que sur des widgets de dépôt compatibles.

Voici un exemple d'utilisation de notre nouveau composant :

```
SimpleDragAndDropPanel amoursCelebres = new SimpleDragAndDropPanel();
Label abelard = new Label("Abélard");
amoursCelebres.addDraggableWidget(
abelard, 10, 10, "heloise_et_abelard");
Label heloise = new Label("Héloïse");
amoursCelebres.addTargetWidget(
heloise, 200, 10, "heloise_et_abelard");
Label tristan = new Label("Tristan");
amoursCelebres.addDraggableWidget(
Tristan, 10, 50, "tristan_et_iseult");
```

```
Label iseult = new Label("Iseult");
amoursCelebres.addTargetWidget(
iseult, 200, 50, "tristan_et_iseult");
```

Vous pouvez maintenant faire mentir l'histoire en réunissant vous-mêmes ces amants malheureux. Notez que vous ne pourrez former que les couples présentant la même affinité. Si vous forcez une rencontre, le galant reviendra tout seul à sa place initiale.



## En guise de conclusion

Cette première itération de notre composant de " Glisser-déposer " est illustratif de la capacité du Google Web Toolkit à intégrer très simplement des éléments qui lui sont complètement exogènes. En moins de 70 lignes, nous avons mis en œuvre un mécanisme qui manque cruellement à la version de base de GWT !

L'exemple de la librairie Yahoo peut être reproduit pour d'autres librairies comme Prototype ou Script.aculo.us. GWT ne vient donc pas concurrencer ces travaux : il permet au contraire d'en profiter plus facilement. On remarque aussi la facilité qu'il y a à mixer JAVA et JavaScript grâce à JSNI. Comme les parties écrites en JavaScript ne sont pas facilement débogables, on aura intérêt à les limiter au strict minimum. L'utilisation la plus judicieuse de JSNI est donc bien de servir de " colle " entre le programme JAVA et des librairies JavaScript du marché.

Notre composant n'est pas complet. Il ne permet pas de réagir aux opérations de " Glisser-déposer ". Nous verrons dans une seconde partie comment rajouter un mécanisme événementiel à notre composant panel. Cela nous permettra de voir comment nous pouvons appeler du code GWT Java à partir d'un code GWT JavaScript.



### ■ Henri DARME

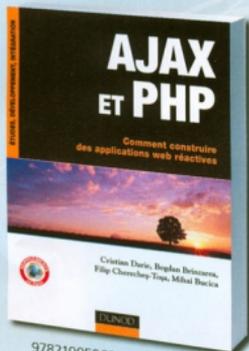
Directeur Technique

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](http://www.objetdirect.com)

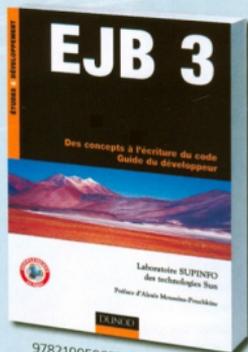
Les auteurs dialoguent avec vous, sur le FORUM  
[www.programmez.com/forum](http://www.programmez.com/forum)

# Les tout derniers savoirs du développeur



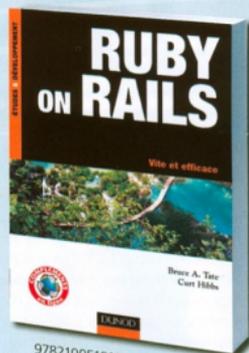
9782100506842 • 312 pages

35 € Cristian Darie,  
Bogdan Brinzarea,  
Filip Chereches-Tosa,  
Mihai Bucica



9782100506231 • 352 pages

38 € SUPINFO



9782100510115 • 208 pages

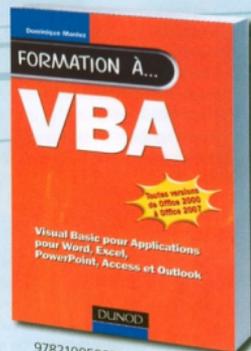
19,90 € Bruce A. Tate  
Curt Hibbs  
Patrick Fabre

▶ Et pour tout savoir  
sur la blogosphère



9782100500598 • 224 pages

19,90 € Loïc Le Meur  
Laurence Beauvais



9782100508723 • 480 pages

17 € Dominique Maniez

LE MOIS  
DE L'INFORMATIQUE  
EN MAI  
CHEZ VOTRE LIBRAIRE



GRATUIT  
pour l'achat  
de 2 livres

L'informatique, ça peut aussi être amusant !

# Mécanisme du fork() avec l'EDI Code::Blocks

La programmation système Linux repose sur les processus pour satisfaire les demandes de l'utilisateur.

Nous allons examiner de plus près le système le plus commun, les processus lourds, qui cohabitent avec d'autres variantes comme les processus légers ou encore l'appel vfork(). La principale différence entre un processus léger et un processus lourd, provient du partage ou du non-partage mémoire entre les différents processus issus du même programme. Code::Blocks, logiciel Open Source (GPL2), est un environnement intégré de développement (IDE) basé sur WxWidgets (multi-plate-forme), qui utilise des plug-in pour s'étendre. Sous Linux il détectera automatiquement le compilateur GCC installé. Dans le menu "Settings/Compiler and debugger", nous avons coché "Enable all compiler warnings" (option -Wall) et "In C mode support all ISO C90 programs" (option -ansi). Les codes sources ont été écrits en C. Code::Blocks travaille sur la base d'un espace de travail (workspace) à partir duquel vous créerez un projet. Un assistant vous guidera pas à pas. Sur cette base nous avons créé les projets fork\_1 à 6. Vous devez juste faire attention à sauvegarder votre projet en indiquant une arborescence sous laquelle l'utilisateur courant (qui a lancé Code::Blocks) a tous les droits. Nous avons utilisé le plug-in "Astyle" pour formater le code source (ceci peut aussi être réalisé en ligne de commande). La construction et l'exécution du projet se réalise par l'appui sur la touche F9. C'est rapide et convivial.

## Dans quel but faire appel à fork() ?

Sous Unix il permet à un exécutable en mémoire, c'est-à-dire à un processus en cours d'exécution, de se dupliquer. Le créateur du processus forké est nommé père ou parent et le nouveau processus, enfant ou fils. Il est intéressant de noter que toutes les caractéristiques de ce processus parent sont recopiés et transmis à l'enfant. En clair avec un fork(), par exemple tous les descripteurs de fichiers qui sont ouverts le restent ! Par contre, si deux processus qui n'ont pas de lien de filiation veulent travailler sur un même fichier ouvert, ils devront communiquer via un socket un message de service contenant les informations sur ce descripteur de fichier ouvert.

## Fonctions fork et sleep

Notre premier exemple illustre le mécanisme de duplication d'un processus. Le noyau recopie la mémoire du processus parent vers son enfant octet par octet, ainsi que les fichiers ouverts. Le code de l'enfant sera exécuté à partir de l'instruction suivant l'appel de la création du processus. L'ID du processus enfant sera cependant unique. Pour le processus parent la valeur de retour de fork devient l'ID du processus enfant créé. Pour le processus enfant la valeur de retour sera 0. Si une erreur de duplication s'est produite la valeur retournée sera -1.

### fork\_1.c

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <errno.h>
#include <sys/types.h>
#include <stdlib.h>
```

```
int main(void)
{
    printf("Exemple d'utilisation de la fonction fork()\n\n");
    pid_t pid; /* Déclaration de la variable pid qui recevra l'ID du
    processus enfant */
    pid = fork(); /* Duplication du processus en cours dans le but de
    créer un processus enfant */

    if ( pid == -1 )
    {
        fprintf(stderr,
            "%s: Une erreur s'est produite avec fork()\n",
            strerror(errno));
        exit(13);
    }
    else if ( pid == 0 ) /* pid est à zéro dans le cas de l'exécution du processus
    enfant ! */
    {
        sleep(1); /* On attend un moment */

        printf("PID %ld: Je suis le processus enfant, l'ID de mon parent est %ld.\n",
            (long)getpid(), /* PID actuel enfant */
            (long)getppid()); /* PID du parent */

        pid_t pidge; /* Déclaration de la variable pid qui recevra l'ID
        du processus 'petit-enfant' */
        pidge = fork(); /* Duplication du processus en cours dans le
        but de créer un processus enfant de l'enfant */

        if ( pidge == -1 )
        {
            fprintf(stderr,
                "%s: Une erreur s'est produite avec fork()\n",
                strerror(errno));
            exit(13);
        }
        else if ( pidge == 0 ) /* pid est à zéro dans le cas de l'exécution
        du processus petit-enfant ! */
        {
            sleep(2); /* On attend un moment */
            printf("PID %ld: Je suis le processus petit enfant, l'ID de mon parent
            est %ld.\n",
                (long)getpid(), /* PID actuel du petit enfant */
                (long)getppid()); /* PID du parent */

        }
    }
    else
    {
        printf("PID %ld: Je suis le processus parent, l'ID de mon enfant est
```

```
%ld.\n",
    (long)getpid(), /* PID actuel du parent */
    (long)pid; /* PID de l'enfant */
}
sleep(4); /* On attend un moment */
return 0;
}
```

### S ./fork\_1

Exemple d'utilisation de la fonction fork()

PID 14103: Je suis le processus parent, l'ID de mon enfant est 14104.  
 PID 14104: Je suis le processus enfant, l'ID de mon parent est 14103.  
 PID 14106: Je suis le processus petit enfant, l'ID de mon parent est 14104.

La fonction sleep permet de réaliser une pause de sorte que le processus enfant ne devienne pas orphelin à la suite d'une fin d'exécution prématurée de son parent (ou grand-père). Pour connaître le numéro de PID de son parent un processus peut appeler la fonction getppid(). L'enfant qui reçoit 0 comme retour de pid appellera cette fonction pour déterminer le numéro PID de son parent. Pour connaître le numéro de son propre PID il appellera getpid().

### Un pas plus loin : les fonctions fork et wait

Avec sleep() le parent "attend" que l'enfant effectue son traitement. Cette solution fonctionne mais il serait préférable d'appeler une fonction qui attende de manière stricte le signe de la fin de son enfant. La fonction wait() ou waitpid() remplira ce rôle. Lorsqu'un processus prend fin, l'état de sortie est enregistré dans une table des processus gérée par le noyau. Cette entrée dans la table n'est pas libérée bien que le processus soit terminé. Elle ne sera libérée qu'à la condition que le processus parent s'informe de l'état de sa "progéniture". Si le processus parent continue son exécution sans se préoccuper de son enfant, et que le processus enfant a pris fin, le processus sera qualifié de "zombie". En clair, un processus est zombie si un processus enfant se termine sans que son parent en soit informé. La commande ps indiquera "Z" comme STAT (état). Wait() prend comme argument un pointeur devant recevoir l'état de retour. Avec waitpid vous pouvez préciser le numéro PID à attendre.

### fork\_2.c

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <errno.h>
#include <sys/types.h>
#include <stdlib.h>
#include <sys/wait.h>
int main(void)
{
    printf("Exemple d'utilisation de la fonction fork() avec wait()\n");
    pid_t pid; /* Déclaration de la variable pid qui recevra l'ID du
    processus enfant */
    pid = fork(); /* Duplication du processus en cours dans le but de
    créer un processus enfant */
    pid_t pid2; /* ID du processus retourné par wait() */

    if (pid == -1 )
```

```
{
    fprintf(stderr,
        "%s: Une erreur s'est produite avec fork()\n",
        strerror(errno));
    exit(13);
}
else if ( pid == 0 ) /* pid est à zéro dans le cas de l'exécution du
processus enfant ! */
{
    printf("PID %ld: Je suis le processus enfant, l'ID de mon parent est
%ld.\n",
        (long)getpid(), /* PID actuel enfant */
        (long)getppid()); /* PID du parent */
    pid = fork(); /* Duplication du processus en cours dans le but
de créer un processus enfant de l'enfant */
    pid2 = waitpid(pid, NULL, 0);

    if ( pid == -1 )
    {
        fprintf(stderr,
            "%s: Une erreur s'est produite avec fork()\n",
            strerror(errno));
        exit(13);
    }
    else if ( pid == 0 ) /* pid est à zéro dans le cas de l'exécution du
processus petit-enfant ! */
    {
        sleep(2);
        printf("PID %ld: Je suis le processus petit enfant, l'ID de mon parent
est %ld.\n",
            (long)getpid(), /* PID actuel du petit enfant */
            (long)getppid()); /* PID du parent */
    }
}
else
{
    pid2 = waitpid(pid, NULL, 0);
    printf("PID %ld: Je suis le processus parent, l'ID de mon enfant est
%ld.\n",
        (long)getpid(), /* PID actuel du parent */
        (long)pid; /* PID de l'enfant */
    )
    return 0;
}
}
```

### ./fork\_2

Exemple d'utilisation de la fonction fork() avec wait()

PID 6510: Je suis le processus enfant, l'ID de mon parent est 6509.  
 PID 6511: Je suis le processus petit enfant, l'ID de mon parent est 6510.  
 PID 6509: Je suis le processus parent, l'ID de mon enfant est 6510.

### Le code de sortie et les macros d'états

La structure alternative choisie pour le fork des deux premiers exemples est assez lourde. Nous pourrions la remplacer de manière plus élégante par un switch. De plus, il nous manque un renseignement : l'état de

retour qui est indiqué comme argument lors de l'appel de la fonction wait() ou waitpid(), est composé de sous-champs. Pour tester ces valeurs de champs, nous pouvons faire appel à des macros. WEXITSTATUS est une macro qui renverra le code retour. WIFEXITED renverra TRUE si le processus s'est terminé normalement. Etc.

## fork\_3.c

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

int main(void)
{
    pid_t pid;
    int rv;

    switch (pid=fork())
    {
        case -1:
            perror("fork"); /* Quelque chose s'est mal passé */
            exit(1); /* sortie */

        case 0:
            printf("Processus Enfant\n");
            printf("PID Enfant %d\n", getpid());
            printf("PID Parent %d\n", getppid());

            switch (pid=fork())
            {
                case -1:
                    perror("fork"); /* Quelque chose s'est mal passé */
                    exit(1); /* sortie */

                case 0:
                    printf("Processus Petit Enfant\n");
                    printf("PID Petit Enfant %d\n", getpid());
                    printf("PID Parent-Enfant %d\n", getppid());
                    exit(1);

                default:
                    wait(&rv);
                    printf("Processus Parent\n");
                    printf("PID Petit Enfant %d\n", pid);
                    printf("PID Parent-Enfant %d\n", getppid());
                    printf("Mon processus petit enfant s'est terminé avec le code de
retour: %d\n", WEXITSTATUS(rv));
                    return (2);
            }

            printf("Entrez un code de retour : ");
            scanf("%d", &rv);
            exit(rv);
    }
}
```

default:

```
wait(&rv);
printf("Processus Parent\n");
printf("PID Enfant %d\n", pid);
printf("PID Parent %d\n", getppid());
printf("Mon processus enfant s'est terminé avec le code de retour:
%d\n", WEXITSTATUS(rv));
return (0);
}
}
```

## ./fork\_3

```
Processus Enfant
PID Enfant 9171
PID Parent 9170
Processus Petit Enfant
PID Petit Enfant 9172
PID Parent-Enfant 9171
Processus Parent
PID Petit Enfant 9172
PID Parent-Enfant 9171
Mon processus petit enfant s'est terminé avec le code de retour: 1
Processus Parent
PID Enfant 9171
PID Parent 9170
Mon processus enfant s'est terminé avec le code de retour: 2
```

## Travaillons avec les signaux

La macro WIFSIGNALED renvoie TRUE si le processus a reçu un signal qu'il n'a pas intercepté. Un processus peut intercepter ces signaux en créant un handler dédié à la réception d'un signal précis (ou de plusieurs signaux). Cette opération est réalisée à l'aide de la fonction signal. Le listing fork\_4 nous montre comment mettre en place cette possibilité : si le signal hangup est lancé par un autre processus à l'encontre du processus en cours d'exécution, le signal sera intercepté.

## fork\_4.c

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <signal.h>

pid_t pid;
int etat;

void handler(int sig)
{
    pid_t pid2;
    pid2 = wait(NULL);

    printf("signal %d reçu du processus ayant le PID %d.\n", sig, pid2);
}
```

```
int main(void)
{
    signal(SIGCHLD,&handler);

    if (!fork())
    {
        printf("Enfant en exécution %d\n", getpid());
        return(0);
    }
    else
    {
        printf("Le PID du Parent est %d\n", getpid());
        return 0;
    }
}
```

#### ./fork\_4

```
Enfant en exécution 12689
signal 17 reçu du processus ayant le PID 12689.
Le PID du Parent est 12688
```

### Communication inter-processus à l'aide de pipes

Si un processus envoie un signal à un autre il lui communique une information. Le programmeur peut aller beaucoup plus loin en communiquant des données. Pour rappel, les processus lourds évoluent dans des espaces mémoires séparés. Nous allons utiliser le mécanisme des pipes pour transmettre des informations d'un processus vers l'autre.

La fonction pipe crée une paire de descripteurs de fichiers pointant sur un i-noeud du tube et les place dans un tableau. Vers\_Enfant[1] est utilisé pour l'écriture tandis que Vers\_Enfant[0] est utilisé pour la lecture. Les deux processus vont se partager le tube, et utiliser les fonctions read et write pour se transmettre des données.

#### fork\_5.c

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
#include <errno.h>
#include <sys/wait.h>
#include <signal.h>

int main()
{
    pid_t child_pid;
    int De_Enfant[2];
    int Vers_Enfant[2];

    /* Création des pipes*/
    if ( pipe(De_Enfant) || pipe(Vers_Enfant) ) {
        perror("Je ne peux créer les pipes");
        exit(errno);
    }
}
```

```
/* fork() */
if ( (child_pid=fork() ) == -1 ) {
    perror("Fork() impossible !");
    exit(errno);
}

if ( child_pid ) {

    char m='p';
    printf("Je suis le processus Parent et mon Enfant est le PID = %d\n",
        child_pid);

    close(De_Enfant[1]);
    close(Vers_Enfant[0]);

    if ( write(Vers_Enfant[1],&m,1) != 1 ) {
        perror("Je n'arrive pas à écrire à mon Enfant");
        exit(errno);
    }

    if ( read(De_Enfant[0],&m,1) != 1 ) {
        perror("Je n'arrive pas à lire en provenance de mon Enfant");
        exit(errno);
    }

    printf("J'ai reçu le caractère '%c' de mon enfant.\n",m);
}

else {
    char m;

    close(De_Enfant[0]);
    close(Vers_Enfant[1]);

    if ( read(Vers_Enfant[0],&m,1) != 1 ) {
        perror("Je n'arrive pas à lire en provenance de mon Parent");
        exit(errno);
    }

    printf("J'ai reçu le caractère '%c' de mon Parent.\n",m);

    m = 'e';
    if ( write(De_Enfant[1],&m,1) != 1 ) {
        perror("Je n'arrive pas à écrire à mon Parent");
        exit(errno);
    }
}

return 0;
}
```

#### ./fork\_5

```
Je suis le processus Parent et mon Enfant est le PID = 14184
J'ai reçu le caractère 'p' de mon Parent.
J'ai reçu le caractère 'e' de mon enfant.
```

## Créons un daemon avec fork()

Poussons le raisonnement : si un processus père se termine après avoir effectué un fork, et que le processus enfant entre dans une boucle infinie, serions-nous en présence d'un daemon ? C'est en fait un peu plus compliqué... Un daemon est un processus dont le pid est 1 comme parent (c'est à dire init). Tout processus dont le parent meurt sans avoir lu l'état de son processus enfant est adopté par init. Si vous réalisez un fork et que vous arrêtez le parent alors que votre enfant ne termine pas vous êtes en présence théoriquement d'un daemon. En pratique vous devez le détacher du terminal avec setsid() et fermer tous les descripteurs for (i=getdtablesize();i>=0;-i) close(i);. Sans oublier qu'il faudra limiter les accès (mode 750 c'est-à-dire complètement de umask(027);) et changer le répertoire de travail (ici /tmp). Enfin pour déterminer que le daemon fonctionne, celui-ci interceptera certains signaux et journalisera cette interception dans un fichier log.

### fork\_6.c

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/file.h>
#include <unistd.h>
#include <fcntl.h>
#include <signal.h>
#include <sys/wait.h>
#include <errno.h>

#define RUNNING_DIR "/tmp"
#define LOCK_FILE "demonfork.lock"
#define LOG_FILE "demonfork.log"
#define F_TLOCK 2

void log_message(filename,message)
char *filename;
char *message;
{
    FILE *logfile;
    logfile=fopen(filename,"a");
    if (!logfile) return;
    fprintf(logfile,"%s\n",message);
    fclose(logfile);
}

void signal_handler(sig)
int sig;
{
    switch (sig)
    {
        case SIGHUP:
            log_message(LOG_FILE,"Signal hangup intercepté",sig);
            break;
        case SIGTERM:
            log_message(LOG_FILE,"Signal terminate intercepté",sig);
            exit(0);
            break;
    }
}
```

```
}
}

void daemonize()
{
    int i,lfp;
    char str[10];
    if (getppid()==1) return; /* Suis-je déjà un daemon ? Oui : retour*/
    i=fork();
    if (i<0) exit(1); /* erreur lors du fork*/
    if (i>0) exit(0); /* je me détache du parent*/
    /* Je suis l'enfant futur daemon */
    setsid(); /* je fais partie d'un nouveau groupe de processus */
    for (i=getdtablesize();i>=0;-i) close(i); /* je ferme tous les descripteurs*/
    i=open("/dev/null",O_RDWR);
    dup(i);
    dup(i);
    umask(027); /* Nouvelle permission et nouveau répertoire de travail*/
    mkdir(RUNNING_DIR);
    lfp=open(LOCK_FILE,O_RDWR|O_CREAT,0640);
    if (!lfp) exit(1); /* je ne peux ouvrir le fichier qui contiendra mon PID*/
    if (lockf(lfp,F_TLOCK,0)<0) exit(0);
    sprintf(str,"%d\n",getpid());
    write(lfp,str,strlen(str)); /* j'écris mon PID dans le fichier */
    signal(SIGCHLD,SIG_IGN); /* j'ignore certains signaux */
    signal(SIGTSTP,SIG_IGN);
    signal(SIGTTOU,SIG_IGN);
    signal(SIGTTIN,SIG_IGN);
    signal(SIGHUP,signal_handler); /* par contre j'intercepte */
    signal(SIGTERM,signal_handler); /* les signaux hangup et kill */
}

int main(void)
{
    daemonize();
    while (1) sleep(1); /* Itération principale : je suis un daemon */
    return(0);
}
```

Ce qui donne :

### ./fork\_6

```
ps fax | grep "7631" | grep "fork_6"
7631 ? Ss 0:00 /home/xl/fork_6/fork_6

cat /tmp/demonfork.lock
7631

kill -HUP `cat /tmp/demonfork.lock`

cat /tmp/demonfork.log
Signal hangup intercepté
```

Téléchargement de Code::Blocks :

[http://wiki.codeblocks.org/index.php?title=Compiled\\_packages\\_of\\_Code::Blocks](http://wiki.codeblocks.org/index.php?title=Compiled_packages_of_Code::Blocks)

■ **Xavier Leclercq** - [Xavier.Leclercq@programmez.com](mailto:Xavier.Leclercq@programmez.com)

# Premiers pas avec Mono sous Linux

*Fin 2006, la version 1.2 du projet Mono a été publiée. Celle-ci comporte de nombreuses évolutions, dont le support quasi complet du framework .NET 2.0. Une occasion s'offre désormais aux développeurs .NET de passer sous Linux.*



Il s'agit même d'une forte volonté de la part de Microsoft puisque la multinationale vient de conclure un accord avec Novell qui vise à rendre compatibles les applications Linux et Windows. L'initiateur du projet, Miguel de Icaza, est par ailleurs l'un des principaux acteurs de l'environnement de bureau GNOME. Cet article vous présente la solution Mono et vous en propose une première approche pratique.

## Les interfaces graphiques en Mono

On ne peut pas parler de Mono sans parler de GTK#. C'est le toolkit qui a été utilisé en premier pour la création d'interfaces graphiques en C#. En plus des widgets d'origine de GTK+, les développeurs ont ajouté des éléments. À titre d'exemple, on peut citer GtkGLAreaSharp qui permet d'embarquer du rendu 3D en utilisant la librairie OpenGL. Celle-ci est comparable à Direct3D, à la différence près qu'elle est portable et particulièrement bien adaptée à la plate-forme Linux.

Pour la conception d'interfaces GTK#, il est possible d'utiliser le design Glade. Celui-ci prend en charge toute la partie de gestion et génération de l'interface pour permettre aux développeurs de se concentrer sur le code utile. Stetic est un nouvel outil qui peut être utilisé seul ou au sein de l'EDI MonoDevelop. Il remplace avantageusement Glade pour la création d'interface en .NET. GTK# n'est pas la seule possibilité offerte aux développeurs pour concevoir leurs interfaces graphiques. L'API Windows.Forms est désormais disponible et propose l'ensemble des contrôles ainsi qu'une implémentation complète de System.Drawing.

## MonoDevelop

EDI libre, initialement conçu pour C# et les autres langages .NET, MonoDevelop est une solution parfaitement intégrée à GNOME. Initialement, cette application était portage de SharpDevelop sous GTK#, cependant il a évolué de bien des manières depuis ses débuts et constitue désormais un outil de qualité supérieure. MonoDevelop fait partie intégrante du projet Mono et peut être considéré comme le Visual Studio .NET de la plate-forme Linux.

Comme tout EDI, MonoDevelop propose de nombreuses fonctionnalités et il dispose d'un design conçu pour faciliter la vie des développeurs. On peut citer notamment la gestion intelligente de la complétion du code. Celle-ci permet de compléter les types, les méthodes et les champs que vous êtes en train de saisir en s'appuyant sur la documentation Mono. On retrouve également un navigateur de classe qui permet de lister les attributs et les méthodes de l'ensemble des classes de votre projet. La documentation du projet Mono est très importante. Vous pouvez accéder à la documentation .NET et Gtk# en quelques clics. MonoDevelop propose des modèles de projets. Ceux-ci vous permettront de démarrer rapidement une application console, GNOME# ou encore Gtk#. Toutes ces fonctionnalités ne sont pas sans rappeler celles de Visual Studio, la



migration vers l'environnement Linux se fera donc à moins mal. Enfin, un dernier point concernant MonoDevelop : son moteur d'extensions. Ce dernier est très puissant et permet de construire un environnement de développement sur mesure grâce à l'API modulaire qui est proposée. MonoDevelop offre également un gestionnaire qui permet d'installer et de publier les extensions depuis et vers les dépôts en ligne.

## Installer Mono et son environnement

Mono est d'ores et déjà intégré au système SuSE Linux 10. Plusieurs applications basées sur ce framework sont également incluses dans le système d'exploitation de Novell. Il est possible de télécharger une image VMWare d'OpenSUSE 10.2 avec Mono 1.2.3. Je vous la recommande, notamment pour réaliser des tests puisque des codes sources sont fournis et l'environnement est préinstallé. De nombreuses distributions proposent le paquetage d'installation, on peut citer Red Hat et Debian.

Une fois l'environnement Mono installé, vous pouvez procéder à l'installation de MonoDevelop. Pour cela, il vous faudra récupérer soit les paquetages, soit les sources. Je vous invite à vous reporter à l'explication qui est fournie sur le site de MonoDevelop afin de le compiler et de l'installer à partir des sources. Compiler MonoDevelop par vous-même vous offre la possibilité de disposer de la dernière version publiée sur le serveur Subversion, l'outil de gestion de versions utilisé par Ximian.

## Un peu de pratique...

Après avoir installé Mono et son environnement, il est intéressant d'effectuer des premiers tests. Voici un premier programme qui va nous permettre d'illustrer la compilation en lignes de commandes. Vous pouvez également créer un nouveau projet dans MonoDevelop à la manière de Visual Studio. Choisissez le modèle "Application console". L'avantage de réaliser des premiers tests avec un éditeur texte et un terminal est de mettre en œuvre le travail nécessaire à la compilation et à l'exécution. Créez le fichier `hello_linux.cs` avec votre éditeur préféré :

```
using System;

public class Hello_Linux {
    public static void Main(string[] args)
    {
        Console.WriteLine("Hello Linux !");
    }
}
```

En ligne de commande vous pourrez compiler de la manière suivante :

```
> mcs -out:hello_linux.exe hello_linux.cs
```

La commande man mcs vous décrira les différentes options disponibles pour le compilateur. Mcs vient de générer un exécutable, nous pouvons le vérifier à l'aide de la commande file :

```
> file hello_linux.exe
```

```
hello_linux.exe:
MS Windows PE 32-bit Intel 80386 console
executable
```

Il peut paraître surprenant que le fichier soit reconnu de type Windows, mais rappelez vous que le framework .NET a été conçu pour cette plateforme. Pour lancer l'application, il suffit de lancer la machine virtuelle avec ce fichier en paramètre :

```
> mono hello_linux.exe
```

En travaillant sous Linux, il est intéressant de profiter des logiciels de développements communément utilisés. Afin d'automatiser la compilation et de créer par exemple un script de lancement, il est judicieux d'utiliser les Makefiles. En voici un exemple basique :

```
all:
mcs -out:hello_linux.exe hello_linux.cs
echo "mono ./hello_linux.exe" > ./hello_linux
chmod +x ./hello_linux
```

Pour construire votre application, vous n'avez plus qu'à exécuter la commande suivante :

```
> make
```

L'exécution se limite désormais à une exécution classique :

```
> ./hello_linux
```

Réalisons à présent une application graphique utilisant GTK#. Il y a davantage de lignes de codes, cependant l'ensemble reste simple à appréhender :

```
using Gtk;
using System;

public class Hello_Linux {

    public static void Main()
    {
        Application.Init();

        Window window = new Window ("Hello Linux !");
        window.Show();

        Application.Run();
    }
}
```

Pour compiler, vous devez préciser un paramètre supplémentaire afin de définir le package à importer :

```
> mcs -out:hello_linux.exe hello_linux.cs -pkg:gtk-sharp
```

Pour l'exécution, la commande mono hello\_linux.exe suffira à lancer l'application. Cet exemple est un peu plus complexe et mérite quelques explications. Tout d'abord, l'appel à Application.Init() est nécessaire à l'initialisation de toute application Gtk#. Ensuite, nous créons la fenêtre avec pour titre "Hello Linux !" et nous demandons son affichage. Enfin, nous lançons la boucle de l'application par l'appel à Application.Run(). Pour quitter le programme, il sera nécessaire d'utiliser la combinaison ctrl+c puisqu'ici les événements ne sont pas gérés.

### Migration d'applications .NET vers Mono

Porter une application .NET vers Mono impose certaines étapes. La première est l'import du projet Visual Studio. Pour ce faire, on pourra utiliser l'outil en ligne de commande prj2make qui permet la conversion du projet Visual vers l'outil make, pierre angulaire de l'automatisation de compilation sous Linux. A ce stade, vous pouvez tenter une compilation du projet, cependant il y a un risque que cela se passe mal. Il y a à cela deux raisons principales. Tout d'abord, il y a fort à parier que celle-ci utilise des librairies disponibles uniquement sous Windows. Ensuite, Mono n'implémente pas totalement le framework .NET. Il vous faudra donc modifier le code.

Pour résoudre la problématique des librairies, la bonne pratique consiste à développer une surcouche sur une librairie disponible sous Linux offrant la même interface que l'ancienne. Cela se réalise simplement dans la plupart des cas grâce à la conception objet ainsi qu'aux nombreuses librairies disponibles. Miguel de Icaza l'a mis en œuvre pour porter Paint.NET et n'a modifié que 129 lignes sur les 70 000 du projet. Mono offre un outil qui permet de gérer la problématique de couverture du framework .NET. MoMA (Mono Migration Analyzer) est capable notamment de trouver les fonctionnalités non implémentées par Mono que votre application utilise. MoMA est disponible sur le site officiel du projet et fonctionne sous Windows comme sous Linux. Il permet d'analyser aussi bien des EXE que des DLL avec une simplicité remarquable. Il suffit de sélectionner le fichier à vérifier puis la version de Mono du système cible. Un rapport est généré, vous avez ainsi les informations nécessaires pour modifier votre code source afin de le rendre compatible Mono. Dans sa version 1.2 Mono couvre déjà la majorité des classes de .NET, vous ne devriez donc pas rencontrer de conflit majeur.

### Conclusion

Mono offre la possibilité étonnante de porter et développer des applications .NET sous Linux. Les développeurs habitués à Windows n'ont pas de difficultés à l'utiliser et peuvent rapidement mettre en œuvre des applications multi-plates-formes, tant au niveau applicatif qu'au niveau matériel. Novell propose également son support pour les besoins spécifiques des entreprises sur cette technologie et poursuit ses efforts autour de son système Suse Linux. Mono apporte un atout non négligeable au framework .NET qui lui permet d'affronter Java sur le terrain de la portabilité et Microsoft l'a bien compris.

#### ■ Lucio Guillosi

# WPF 3e partie : la 3D

Dans le numéro Hors Série spécial .Net 3.0, nous avons vu la 3D dans WPF. Dans cet article, nous allons essayer d'aller un peu plus loin et de voir, à partir d'un simple cube, les différentes possibilités qu'offre l'utilisation de la 3D en terme d'animations, de caméra ou de lumières.

Nous étudierons les caméras, les différents mouvements possibles, et les différentes lumières offertes par le Framework 3D. Je montrerai aussi l'utilisation de quelques outils pour faciliter le développement de la 3D. Pour la scène de départ, nous reprendrons celle vue dans le hors série .Net, dont le code est (résultats figure 1) :

```
<Window x:Class="WPF3DProgrammez.Window1"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="WPF3DProgrammez" Height="300" Width="300"
>
<Grid>
<Grid.Resources>
<ResourceDictionary>
<Storyboard x:Key="OnLoaded">
</Storyboard>
</ResourceDictionary>
</Grid.Resources>
<Viewport3D Name="mainViewport" ClipToBounds="True"
Width="Auto" Height="Auto" MinWidth="100" MinHeight="100">
<Viewport3D.Camera>
<PerspectiveCamera
FarPlaneDistance="100"
LookDirection="5,5,5"
UpDirection="0,1,0"
NearPlaneDistance="0"
Position="5,5,5"
FieldOfView="45" />
</Viewport3D.Camera>
<ModelVisual3D>
<ModelVisual3D.Content>
<Model3DGroup>
<DirectionalLight
Color="White"
Direction="-2,-3,-1" />
<GeometryModel3D x:Name="pCube1">
<GeometryModel3D.Geometry>
<MeshGeometry3D Positions="1,1,1,
1,1,0,0,1,0,0,1,1,1,1,1,1,1,1,0,0,1,0,0,1,1,0,1,1,1,
1,1,1,0,1,1,0,0,1,0,0,1,1,0,1,1,1,0,1,0,1,1,0,1,0,0,0,0,
1,0,0,0,1,0,0,1,0,1,1,0,1,0,0,1,0,0,0,0,0,0,1,0,1,1,0,1,0,1,0,
0,0,0" TextureCoordinates="0,1,1,1,1,0,1,0,0,0,1,0,1,1,1,0,1,0,1,0,
0,0,0,1,1,1,1,0,1,0,0,0,1,0,1,1,1,0,1,0,0,0,1,0,1,1,1,0,1,0,
1,0,0,0,1,0,1,1,1,0,1,0,0,0,1,1,1,1,0,1,0,0,0,1,0,1,1,1,0,1,0,
1,0,0,0,1,0,1,1,1,0,1,0,0,0,1,1" />
</GeometryModel3D.Geometry>
<GeometryModel3D.Material>
<MaterialGroup>
```

```
<DiffuseMaterial>
<DiffuseMaterial.Brush>
<ImageBrush ImageSource=
"C:\Users\xvanneste\Pictures\00.jpg"/>
</DiffuseMaterial.Brush>
</DiffuseMaterial>
<SpecularMaterial>
<SpecularMaterial.Brush>
<SolidColorBrush Color="#fff"
Opacity="1" />
</SpecularMaterial.Brush>
</SpecularMaterial>
</MaterialGroup>
</GeometryModel3D.Material>
</GeometryModel3D>
</Model3DGroup>
</ModelVisual3D.Content>
</ModelVisual3D>
</Viewport3D>
</Grid>
</Window>
```



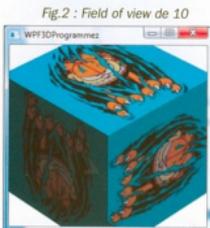
(Fig.1)

C'est à partir de cette scène que nous allons étudier les différents composants de l'environnement 3D

## Les caméras

### Les perspectives caméras

C'est celle qui est actuellement utilisée dans la scène. Ces caméras ressemblent beaucoup aux caméras traditionnelles utilisées dans le monde du cinéma. Comme les caméras réelles elles ont un FieldOfView qui peut donner un aspect très différent à une scène (Fig. 2 - 3 - 4) :



(Fig.2)

Fig.2 : Field of view de 10



(Fig.3)

Fig.3 : Field Of View de 45



Fig. 4 : Field of view de 100



Fig. 5

La caméra ne bouge pas, c'est l'angle d'ouverture qui change, on passe de 10°, à 45° puis à 100°. La perspective est cependant déformée dans les extrêmes. Par exemple, voici ce que cela donne avec un grand angle mais en rapprochant la caméra : Fig. 5.

```
<Viewport3D.Camera>
  <PerspectiveCamera
    FarPlaneDistance="100"
    LookDirection="-5,-5,-5"
    UpDirection="0,1,0"
    NearPlaneDistance="0"
    Position="1.5,1.5,1.5"
    FieldOfView="100" />
</Viewport3D.Camera>
```

La perspective est très exagérée. La même chose avec un petit angle mais éloigné.

```
<Viewport3D.Camera>
  <PerspectiveCamera
    FarPlaneDistance="100"
    LookDirection="-5,-5,-5"
    UpDirection="0,1,0"
    NearPlaneDistance="0"
    Position="10,10,10"
    FieldOfView="10" />
</Viewport3D.Camera>
```

Le cube semble alors aplati et la perspective est presque inexistante. On s'accorde à dire qu'un field of view de 45° correspond à peu près à ce que l'on voit à la télé. Un autre type de caméra est l'orthographique caméra. A la différence de la perspective caméra, ici on n'a pas d'impression de perspective. Ainsi, voici à quoi ressemble notre scène avec une orthographique caméra :

```
<Viewport3D.Camera>
  <OrthographicCamera
    FarPlaneDistance="100"
    LookDirection="-5,-5,-5"
    UpDirection="0,1,0"
    NearPlaneDistance="0"
    Position="10,10,10" />
</Viewport3D.Camera>
```

### Les lumières.

Comme les caméras, les lumières sont importantes dans une scène, ce sont elles qui lui donnent toute son ambiance.



Fig. 6



Fig. 7



Fig. 8



Fig. 9

**La DirectionalLight** : La lumière directionnelle est une lumière qui imite le soleil. En fait c'est une lumière qui envoie tous les rayons dans le même sens et de manière parallèle. C'est celle que nous avons dans notre scène et qui éclaire plus le sommet que les côtés de notre cube. Fig. 6.

```
<DirectionalLight
  Color="White"
  Direction="2,-3,-1" />
```

**La SpotLight** : La spot light ressemble à un spot conique. C'est-à-dire que la lumière forme un cône en partant d'un point et en s'éloignant de la source dans une direction en grandissant. On peut définir deux angles pour cette lumière l'inner qui correspond à l'angle de la lumière maximum envoyé et l'outer qui correspond à l'atténuation de la lumière. Bien entendu l'outer doit être supérieur à l'inner. Voici le résultat : Fig. 7.

```
<SpotLight
  Color="White"
  Direction="-3,-3,-3"
  Position="3,3,3"
  InnerConeAngle="5"
  OuterConeAngle="30" />
```

Comme vous pouvez le voir, le milieu de la lumière est plus clair que les extrémités du cône lumineux. Changeons un peu les valeurs : Fig. 8 - Fig. 9.

```
<SpotLight
  Color="White"
  Direction="-3,-3,-3"
  Position="3,3,3"
  InnerConeAngle="5"
  OuterConeAngle="6" />
```

```
<SpotLight
```



Fig.10



Fig.11

```

Color="White"
Direction="-3,-3,-3"
Position="3,3,3"
InnerConeAngle="20"
OuterConeAngle="30"
/>

```

**L'ambient light** : Comme son nom l'indique l'ambient light est une lumière d'ambiance. En fait toute la scène est éclairée de manière uniforme dans l'atténuation. Chose qui est impossible dans le monde réel. Voir le résultat en figure 10.

```
<AmbientLight/>
```

Bien entendu rien n'est moins réaliste que cette lumière.

**Le PointLight** : Comme son nom l'indique, le PointLight est un point lumineux qui se projette sur la forme. Ainsi on peut éclairer plus une face qu'une autre. Par exemple :

```
<PointLight Position="1.5,0.5,0.5"/>
```

**Mélange de lumière** : On peut mélanger les lumières pour obtenir des ambiances particulières :

```

<PointLight
    Position="1.5,0.5,0.5"
    Color="Red"
/>

    <PointLight
    Position="0.5,1.5,0.5"
    Color="Green"
/>

    <PointLight
    Position="0.5,0.5,1.5"
    Color="Blue"
/>

```

## Animations

Il y a trois axes d'animation en WPF et en 3D en général. La translation, la rotation, et la mise à l'échelle.

**La Translation** : La translation est la plus simple à comprendre. C'est le fait de prendre un objet et de le mettre un peu plus loin sur un des axes. Les translations peuvent s'appliquer à la géométrie ou au modèle 3D. La différence est que la géométrie, c'est le cube seul ; alors que le modèle 3D c'est le cube plus la lumière dans notre scène. Voici un exemple de translation (Résultat figure 11) :

```

</GeometryModel3D.Material>
<GeometryModel3D.Transform>
    <TranslateTransform3D OffsetX="1"/>
</GeometryModel3D.Transform>
</GeometryModel3D>

```

La translation peut bien entendu être combinée sur plusieurs axes :

```
<TranslateTransform3D OffsetX="1" OffsetY="0.5"/>
```

**La rotation** : Il y a deux types de rotations. La première, et aussi la plus simple à comprendre, est la rotation autour d'un axe. En fait, le cube va faire une rotation par rapport à son point d'origine sur un ou plusieurs axes.

```

<GeometryModel3D.Transform>
    <RotateTransform3D>
        <RotateTransform3D.Rotation>
            <AxisAngleRotation3D Axis="1 0 0" Angle="60"/>
        </RotateTransform3D.Rotation>
    </RotateTransform3D>
</GeometryModel3D.Transform>

```

La propriété axis prend un ratio X Y Z. Ainsi, le 1 indique que 100% de la rotation doit s'appliquer à l'axe X.

```

<GeometryModel3D.Transform>
    <RotateTransform3D>
        <RotateTransform3D.Rotation>
            <AxisAngleRotation3D Axis="1 0.5 0" Angle="60"/>
        </RotateTransform3D.Rotation>
    </RotateTransform3D>
</GeometryModel3D.Transform>

```

Le deuxième type de rotation est la rotation par Quaternion qui est très complexe et dépasse largement le cadre de cet article.

**La mise à l'échelle** : La dernière transformation est la mise à l'échelle qui permet de déformer une géométrie 3D. Une mise à l'échelle se fait en utilisant ScaleTransform de la manière suivante :

```

<GeometryModel3D.Transform>
    <ScaleTransform3D ScaleX="2" ScaleY="0.5" ScaleZ="1.5"/>
</GeometryModel3D.Transform>

```

## Conclusion

Au cours de nos articles, nous avons vu les bases du WPF, puis le déploiement et l'aspect Smart Client de celui-ci, notamment avec le WPF/E et les XBAP. Enfin, nous avons exploré les bases de la 3D. Le but de ces articles était de réellement mettre en place les techniques de base du WPF et voir comment débiter en WPF. Sur mon site <http://www.netfxfactory.org>, j'irais encore plus loin, notamment avec les Quaternions, les animations et les storyboard et la communication avec les bases de données.

■ **Xavier Vanneste** - [xvanneste@msn.com](mailto:xvanneste@msn.com)  
<http://www.netfxfactory.org>

# Générez vos documents PDF et RTF avec Java

Générer des documents au format PDF ou RTF est un besoin très courant. De nombreuses librairies existent pour le faire. Nous regardons aujourd'hui iText, une librairie pour Java.

**G**énérer des documents humainement lisibles depuis des applications est un besoin récurrent. Qui se risquerait à lire un ResultSet empaqueté dans du XML ? Un bon document au format HTML ou PDF ou RTF sera préférable :) Ceci posé, il y a plusieurs approches. Parmi celles-ci, nous pouvons choisir de générer un document à l'aide d'un outil travaillant sur des modèles de documents. C'est l'approche de FOP (<http://xmgraphics.apache.org/fop/>). Nous pouvons aussi incorporer à notre application du code qui génèrera le document en s'appuyant sur les facilités apportées par une librairie. C'est la démarche que nous suivons aujourd'hui, avec iText, une librairie Java qui permet de générer des documents HTML, PDF et RTF. Nous nous intéressons plus particulièrement à ces deux derniers.

## 1 Pour se faire la main

Sans doute existe-t-il de nombreuses librairies Java pour servir notre propos. Nous avons choisi iText pour plusieurs raisons. D'abord c'est une librairie gratuite, ensuite c'est une librairie très complète qui bénéficie maintenant d'une grande expérience, enfin elle est facile à appréhender, ce qui ne gâche rien. Vous pouvez télécharger iText et sa Javadoc à <http://www.lowagie.com/iText/>. Venant sous la forme d'un seul Jar, il suffit de pointer celui-ci par le classpath pour une utilisation autonome, ou de le recenser en tant que librairie sous un EDI tel que Eclipse ou Netbeans. Vous trouverez tous nos exemples complets sur le Cd-Rom accompagnant le magazine, ou sur notre site. Tradition oblige, voici notre premier exemple qui génère d'un seul coup un "Hello World" de quatre façons différentes: HTML, RTF, PDF, et ... une sortie sur la console :)

```
import com.lowagie.text.Rectangle;
import com.lowagie.text.Document;
import com.lowagie.text.PageSize;
import com.lowagie.text.Paragraph;
import com.lowagie.text.html.HtmlWriter;
import com.lowagie.text.pdf.PdfWriter;
import com.lowagie.text.rtf.RtfWriter2;
// d'autres import ...
```

```
public class HelloWorld {
    public static void main(String[] args) {
        final float marge = 100.0f;
        // Définir le rectangle englobant chaque page
        Rectangle rect = PageSize.A4;
        rect.setBackgroundColor(Color.LIGHT_GRAY);
        rect.setBorder(Rectangle.LEFT | Rectangle.RIGHT
            | Rectangle.TOP | Rectangle.BOTTOM);
        rect.setBorderColor(Color.BLACK);
        rect.setBorderWidth(10.0f);

        // Créer le document lui même,
        // en définissant une marge
        Document document = new Document(
```

```
rect, marge, marge, marge, marge);
try {
    // création des writers
    HtmlWriter.getInstance(document,
        new FileOutputStream("helloworld.html"));
    PdfWriter.getInstance(document,
        new FileOutputStream("helloworld.pdf"));
    PdfWriter.getInstance(document, System.out);
    RtfWriter2.getInstance(document,
        new FileOutputStream("helloworld.rtf"));
    // Quelques méta-données
    document.addAuthor("Fred");
    document.addCreator("Programmez!");
    document.addSubject(
        "Hello World généré avec iText");
    document.addTitle("HelloWorld - Programmez!");

    document.open();
    document.add(new Paragraph("Programmez!"));
    document.add(new Paragraph("Abonnez vous -:"));
} catch (Exception e) {
    System.err.println(e.getMessage());
}
finally {
    document.close();
}
}
```

Nous avons donné une partie des imports de ce premier exemple pour sensibiliser le lecteur aux risques de confusions qu'il est possible de faire en travaillant avec iText. Ainsi la classe Rectangle utilisée n'est pas celle du package java.awt. Cet exemple est bénin, mais attention, lorsque vous travaillerez avec les classes spécifiques à la génération de documents PDF et RTF. Une grande partie sont dépréciées, et résident dans des packages aux noms très voisins de ceux accueillant les classes valides. La démarche de travail apparaît clairement dans ce code. Nous commençons par définir la taille de la page, sa couleur de fond et un bord. Ensuite nous créons logiquement un document avec une marge. Arrivent maintenant les choses intéressantes. Les Writers sont instanciés à partir d'un document et d'un OutputStream. Les Writers sont en quelque sorte des flux Java automatiques. On ajoute quelque chose au document et cela se répercute sur le fichier généré via le flux. Dans notre exemple, nous remarquons qu'un même document peut être associé à plusieurs Writers, ce qui nous permet, avec un même code et des Writers dédiés, de générer plusieurs formats. Nous pouvons même sortir sur la console le contenu brut des documents formats en passant System.out à un Writer en guise de OutputStream :) Pouvons nous dire pour autant qu'il est possible de toujours générer plusieurs formats en même temps ? Pas vraiment. Si le document est très

simple, la réponse est oui. Mais dès qu'on y incorpore des fioritures cela ne marche plus très bien. Déjà dans notre exemple il y a un petit os. Nous avons défini un bord. Celui-ci, qui n'a rien à voir avec la marge, doit apparaître sur le bord de la feuille. Essayez notre code et vous verrez que cela ne fonctionne qu'avec PDF. Et à y réfléchir cela se comprend aisément. S'il vaut mieux abandonner l'idée de générer plusieurs formats avec un seul code, iText apporte malgré tout l'avantage que la démarche de codage est la même pour chaque format.



Notre premier document PDF complexe généré avec iText.

## 2 Un document PDF plus évolué

Nous voulons maintenant générer un document PDF qui comporte un logo, et du texte en italique et en couleur, comme illustré ci-contre. Autant d'éléments qui trouvent leurs correspondances dans des classes

iText. Le plus petit élément possible est le Chunk. Un chunk peut contenir un simple caractère, du texte associé à une couleur et une police, ou une image ou même ... d'autres Chunk. Vient ensuite la Phrase, classe dont le nom parle de lui-même, mais dont l'utilité dans la pratique est moins claire. Vient ensuite le Paragraph, dont le nom est suffisamment explicite. Il peut contenir des Chunk et des Phrases. Il existe bien sûr d'autres objets de haut niveau tels que les tableaux et leurs cellules, ce que nous verrons plus loin. Tous ces objets implémentent l'interface Element et comme tels peuvent être incorporés à un document. Il existe aussi des éléments électroniques libres si l'on peut dire, comme la classe Image qui normalement est conçue pour être intégrée à un Chunk, mais qui, puisqu'elle implémente aussi Element peut être incorporée à un document directement. Munis de toutes des informations, écrivons notre code:

```
// Programmez.java sur le Cd-Rom
// Voir les import sur le Cd-Rom
public class Programmez {
    public static void main(String[] args) {
        Document document = new Document(PageSize.A4,
            50.0f, 50.0f, 50.0f, 50.0f);

        try {
            PdfWriter.getInstance(document,
                new FileOutputStream("programmez.pdf"));
            PdfWriter2.getInstance(document,
                new FileOutputStream("programmez.rtf"));
            document.addAuthor("Fred");
            document.addCreator("Programmez!");
            document.addSubject("Magazine Programmez!");
            document.addTitle("Présentation");

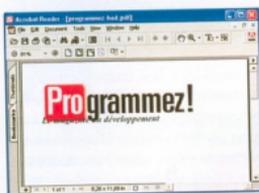
            document.open();

            Image image = Image.getInstance(
                "logo_programmez.jpg");
            image.setAlignment(Image.ALIGN_LEFT);
```

```
image.scaleAbsolute(400.0f, 112.0f);
document.add(image);
```

```
/*
 * Ici, mauvaise méthode, la phrase
 * chevaucherait l'image
```

```
Phrase phrase = new Phrase(
    "Le magazine du développement",
    FontFactory.getFont(FontFactory.TIMES, 24,
        Font.BOLDITALIC, new Color(0, 0, 0));
    document.add(phrase);
*/
Paragraph paragraph = new Paragraph(
    "Le magazine du développement",
    FontFactory.getFont(
        FontFactory.TIMES_ITALIC, 24,
        Font.BOLDITALIC, new Color(0, 0, 0));
    document.add(paragraph);
paragraph = new Paragraph();
paragraph.setSpacingBefore(10.0f);
Chunk chunk = new Chunk("Abonnez vous!",
    FontFactory.getFont(FontFactory.TIMES_ITALIC, 24,
        Font.BOLDITALIC, new Color(255, 0, 0));
    paragraph.add(chunk);
chunk = new Chunk(")",
    FontFactory.getFont(
        FontFactory.TIMES_ITALIC, 24,
        Font.BOLDITALIC, new Color(0, 0, 255));
    paragraph.add(chunk);
    document.add(paragraph);
} catch (Exception e) {
    System.err.println(e.getMessage());
}
finally {
    document.close();
}
}
```



Une mauvaise utilisation de la classe Phrase.

Le lecteur voudra bien veiller à ce que le fichier logo\_programmez.jpg puisse être localisé par la JVM et modifiera notre exemple si nécessaire. Le lecteur remarquera en outre que nous nous sommes amusés à générer quand même un document RTF en même temps. Vous constaterez par vous-même que si le PDF embarque bien une image étirée comme nous le voulions, le RTF se contente d'embarquer l'image à ses dimensions d'origine. Les lignes de codes en commentaire illustrent un piège tendu par la classe Phrase. Si nous incorporions une Phrase au document sans la mettre dans un paragraphe, nous aurions un problè-

me. Vous constaterez par vous-même que si le PDF embarque bien une image étirée comme nous le voulions, le RTF se contente d'embarquer l'image à ses dimensions d'origine. Les lignes de codes en commentaire illustrent un piège tendu par la classe Phrase. Si nous incorporions une Phrase au document sans la mettre dans un paragraphe, nous aurions un problè-

me de mise en page comme illustré sur la figure précédente. En outre, Paragraph permet de définir un espace en amont et en aval de lui, ce que Phrase ne permet pas du tout.

### 3 Construire un tableau

Nous voulons enrichir notre document PDF par un tableau des membres de l'équipe de Programmez! comme illustré ci-contre. Pour



Notre document est maintenant enrichi par un tableau.

cela iText vient avec une classe très puissante: Table. Nous utilisons plutôt PdfPTable qui est une décoration de la classe Table. Attention aux confusions. PdfPTable qui existe aussi, est une classe moins intéressante. Voici l'extrait concernant la table dans notre code :



Notre document au format RTF.

être ingrat en raison de la différence de la prise en charge du format par les logiciels de traitement de textes. Si vous n'obtenez pas le résultat escompté, avant toute autre chose, essayer de visionner le document avec plusieurs outils. Notre exemple génère un document semblable au précédent, hormis les bordures des cellules qui veulent illustrer les différences de salaires entre les membres de l'équipe :). Le code, très répétitif, parle de lui même et nous n'en donnons qu'un court extrait ici.

```
// ProgrammezPDF.java
PdfPTable table = new PdfPTable(2);
table.setSpacingBefore(20.0f);
PdfCell cell =
    new PdfCell(new Paragraph("L'équipe:"));
cell.setBorderWidth(3.0f);
cell.setPadding(8.0f);
cell.setColspan(2);
table.addCell(cell);

table.getDefaultCell().setPadding(4.0f);
table.getDefaultCell().setBorderWidth(3.0f);

table.addCell("Jean");
table.addCell("Kaminsky");
table.addCell("François");
table.addCell("Tonic");
table.addCell("Sébastien");
table.addCell("Develay");
table.addCell("Frédéric");
table.addCell("Mazué");

document.add(table);
```

```
Table table = new Table(2);
table.setBorderWidth(20.0f);
table.setPadding(5.0f);
table.setSpacing(5.0f);

RtfCell rtfcell = new RtfCell("L'équipe");
rtfcell.setBorders(new RtfBorderGroup(
    Rectangle.BOX, RtfBorder.BORDER_DOUBLE_WAVY,
    5.0f, new Color(0, 0, 0)));
rtfcell.setColspan(2);
rtfcell.setHeader(true);
table.addCell(rtfcell);
table.endHeaders();

rtfcell = new RtfCell("Jean");
rtfcell.setBorders(new RtfBorderGroup(
    Rectangle.BOX, RtfBorder.BORDER_STRIPED,
    1.0f, new Color(0, 0, 0)));
table.addCell(rtfcell);

rtfcell = new RtfCell("Kaminsky");
rtfcell.setBorders(new RtfBorderGroup(
    Rectangle.BOX, RtfBorder.BORDER_STRIPED,
    1.0f, new Color(0, 0, 0)));
table.addCell(rtfcell);

rtfcell = new RtfCell("François");
rtfcell.setBorders(new RtfBorderGroup(
    Rectangle.BOX, RtfBorder.BORDER_DOUBLE,
    1.0f, new Color(0, 0, 0)));
table.addCell(rtfcell);

// etc, etc.
document.add(table);
```

Pour travailler avec les tables sous iText, il faut toujours connaître à l'avance le nombre de colonnes. Ensuite, il faut avoir à l'esprit qu'une cellule (classes Cell et autres) ne correspond pas obligatoirement à une "case" du tableau. Une cellule peut ajuster sa largeur pour occuper une ou plusieurs colonnes et donc autant de cases. Notre code illustre ceci en étendant la première cellule sur toute la largeur du tableau. Enfin, nous montrons que s'il est possible de travailler sur chaque cellule individuellement, agir sur les propriétés de la cellule par défaut allège considérablement le travail.

### 4 Générer des documents RTF

Ce travail a ses propres particularités. D'abord le Writer ne travaille pas comme les autres qui écrivent le document dans un fichier au

■ Frédéric Mazué - [fmazue@programmez.com](mailto:fmazue@programmez.com)

# Surveillez vos fichiers sous Linux avec INotify

Les noyaux Linux les plus récents intègrent une nouvelle fonctionnalité de surveillance de fichiers très efficace. Nous faisons connaissance avec elle et nous apprenons à la mettre en œuvre avec C++.

Pouvoir être notifié d'actions effectuées dans un système de fichiers est très important. Des outils de recherche et d'indexation tels que Beagle (<http://beagle-project.org/>) ne pourraient fonctionner sans cela autrement qu'en parcourant totalement l'arborescence à intervalles réguliers pour se mettre à jour. Approche peu efficace s'il en est. Pouvoir être informé en cas de modification indésirable d'un fichier est également important pour la sécurité d'un système. Tous les systèmes d'exploitation offrent une solution à ce problème. Jusqu'à récemment, la solution Linux était dnotify, bien connue pour ses défauts. Ceux-ci sont notamment la limitation de la surveillance aux répertoires, la contrainte de gérer un descripteur de fichier par répertoire surveillé et l'impossibilité de démonter le périphérique accueillant le ou les répertoires sous surveillance. Le remède s'appelle INotify. Cette nouvelle fonctionnalité est intégrée au noyau et a été développée par Robert Love et John McCutchan, deux éminents codeurs de la communauté. Nous allons découvrir ensemble cette nouvelle fonctionnalité tout en écrivant un peu de code C++ que vous trouverez sur le site [www.programmez.com](http://www.programmez.com). Ce code est l'adaptation et la simplification du code d'une application réelle en cours de développement. Le rôle de cette application est d'accroître la sécurité d'un serveur en s'assurant qu'un hacker ne vient pas modifier des fichiers ou répertoires sensibles. Pour profiter au mieux de cet article, le lecteur doit avoir de bonnes notions de programmation système Linux. Il doit également connaître la librairie C++ Boost. Si ce n'est pas le cas, nous lui suggérons de se reporter à Programmez! 93 et 96 pour se familiariser avec Boost.

## 1 Un noyau 2.6.13 sans pépins

INotify est intégré au noyau à partir de la version 2.6.13. Votre serveur utilise un noyau 2.6.18. Avoir le noyau requis ne suffit pas, encore faut-il que celui-ci soit compilé avec le support d'INotify. Compiler avec ce support est à priori toujours une bonne idée, même si vous n'avez pas l'intention de programmer avec INotify. En effet, des outils tel que Beagle sus mentionné, s'ils peuvent se passer d'INotify, fonctionnent incomparablement mieux avec. Pour configurer la compilation de votre noyau, lancez le traditionnel `make menuconfig` puis allez dans "File systems" comme ceci :

```
File systems -->
[*] Inotify file change notification support
[*] Inotify support for userspace
```

Ensuite, compilez et installez votre noyau selon la procédure habituelle.

## 2 Les principes d'INotify

Voici ce que l'on pourrait appeler un Hello World INotify :

```
#include <iostream>
#include <sys/inotify.h>
```

```
int main() {
    int fd;
```

```
// Initialiser inotify
fd = inotify_init ();
if (fd < 0)
    std::cerr << "Erreur initialisation INotify"
              << std::endl;
else
    std::cout << "Hello INotify :)" << std::endl;
// annuler la notification inotify
close(fd);
}
```

Comme le montre ce code, INotify est initialisé par l'API `inotify_init` qui retourne un descripteur de fichier classique, que l'on ferme, également de façon classique, pour libérer toutes les ressources utilisées par INotify. Ce descripteur de fichier est le seul que l'on doit gérer même si on surveille des milliers de fichiers. L'astuce est que INotify ne surveille pas les fichiers eux-mêmes mais leurs inodes dans le système de fichiers inodes. On ajoute ou retire des points de surveillance par le biais des API `inotify_add_watch` et `inotify_rm_watch`. Outre le descripteur de fichier INotify, la première API reçoit le chemin du point à surveiller et un masque spécifiant les événements à notifier. La seconde API reçoit l'identificateur de point de surveillance retourné par la première. Lorsqu'INotify s'aperçoit qu'un inode est modifié, il remplit une ou plusieurs structures `inotify_event` et les écrit dans le fichier système identifié par le descripteur INotify. Pour être notifiée, l'application client lit simplement le descripteur de fichier. Une application client peut parfaitement lire simultanément d'autres descripteurs de fichiers, n'ayant rien à voir avec INotify, au moyen de l'API `select` pour prendre un exemple. Le seul point délicat dans la programmation avec INotify réside dans la structure `inotify_event` ainsi déclarée dans `sys/inotify.h`

```
struct inotify_event {
    int wd;
    uint32_t mask;
    uint32_t cookie;
    uint32_t len;
    char name[0];
};
```

Le membre `name` doit contenir le nom du point de surveillance quand il existe. Il est clair que zéro char sera très insuffisant la plupart du temps ! L'application client doit donc travailler avec une taille de tampon par exemple ainsi définie :

```
size_t buffer_size = sizeof(struct inotify_event) + 32;
```

Si ce tampon suffit INotify écrit, pour notifier un événement, le total de la structure dans le fichier système identifié par le descripteur, à la suite de quoi `read` peut lire côté client et l'application exploiter les données.

Si le tampon est trop petit, ce qui est a priori toujours le cas au début car `Inotify` notifie en général plusieurs événements en une seule fois, alors read signale 0 octet lui tandis que les événements restent en attente. L'application client ainsi informée peut alors augmenter la taille du tampon à la volée. Le membre len de la structure correspond à la longueur du nom contenu dans name éventuellement augmenté d'octets de rembourrages, nécessaires pour l'alignement des enregistrements. Puisque plusieurs événements sont notifiés simultanément la plupart du temps, il est à la charge de l'application client d'itérer dans le tampon de lecture autant que nécessaire. Compliqué ? Rien de tel qu'un peu de code pour nous éclaircir les idées :

### 3 Anatomie de notre application de démonstration.

`Inotify` est vraiment très bien, mais il faut apprendre à le manipuler. Sa nature fondamentalement asynchrone oblige à réfléchir à l'organisation des applications devant travailler avec. `Inotify` étant très souple, on peut l'aborder de nombreuses façons, toutes valables. Nous avons choisi d'écouter les notifications dans un thread dédié, nous appuyant sur le fait que read place ce thread en attente efficace. Deux choses sont susceptibles de débloquent read. Des événements qui arrivent, ou bien qu'il n'y ait plus de point de surveillance associé au descripteur `Inotify`, auquel cas read retourne -1, ce qui est pour nous une occasion d'arrêter notre thread. Notre application de démonstration comporte un deuxième thread qui simule le coeur de l'application. Nous établissons la communication entre les deux thread au moyen d'un très classique tube Unix. Le thread de l'application lit la sortie du tube en mode non bloquant et s'il n'y a rien à lire, imprime des points sur la console pour simuler un autre travail. Pour arrêter l'application, vous devez appuyer sur la touche `[return]` dans la console de lancement. Pour lancer l'application vous devez lui donner quelque chose à surveiller. Par exemple un répertoire :

```
./Demoinotify /home/fred
```

Mais cela fonctionnerait tout aussi bien sur un fichier :

```
./Demoinotify /home/fred/un_fichier
```

Puis dans une autre console, essayez :

```
cd /home/fred
touch dummy
echo "programmez" >> dummy
rm dummy
```

```
root@fred:~/development/cpp/recligne-un/DemoInotify/Debug
$ ./Demoinotify /home/fred
Application lancée -- appuyez sur [return] pour arrêter
Inotify: /home/fred dummy 100 -> IN_CREATE
Inotify: /home/fred dummy 20 -> IN_DELETE
Inotify: /home/fred dummy 4 -> IN_ATTRIB
Inotify: /home/fred dummy 8 -> IN_CLOSE_WRITE
...
Inotify: /home/fred dummy 20 -> IN_DELETE
Inotify: /home/fred dummy 10 -> IN_CLOSE_NOWRITE
.....
Inotify: /home/fred dummy 200 -> IN_DELETE
.....
root@fred:~/development/cpp/recligne-un/DemoInotify/Debug
```

Fig.2 - Notre programme de démonstration en action.

L'illustration ci-dessus, montre le résultat des opérations. Si vous essayez la surveillance d'un fichier, vous verrez que celle-ci se poursuit même si le

fichier est déplacé. Ce tour de magie est possible car `Inotify` surveille les inodes et non les fichiers eux-mêmes, comme nous l'avons dit.

### 4 Une encapsulation d'`Inotify`

Nous proposons une classe relativement simple pour les besoins de la démonstration. Son constructeur reçoit en argument le chemin à surveiller, l'extrémité d'entrée du tube de communication ainsi que le masque d'événements. Notre classe ne surveille donc qu'un seul point à la fois, mais il serait facile de l'étendre en lui ajoutant des méthodes enveloppant les API `inotify_add_watch` et `inotify_rm_watch`. Le code du constructeur mérite quelques commentaires :

```
atcher:InotifyWatcher(std::string inode,
int fd, unsigned int events) :
    inode(inode), fd_tube_in(fd), buffer_size(0),
    fd_inotify(-1), fd_watch(-1), fini(false) {
    fd_inotify = inotify_init ();
    if (fd_inotify < 0) {
        // Dans une application réelle on leverait probablement
        // une exception un peu plus évoluée ici ;}
        throw 1;
    }
    buffer.reset(new char[1]);
    // Définir le point de surveillance
    fd_watch = inotify_add_watch (fd_inotify,
    inode.c_str(), events);
    if (fd_watch < 0) {
        close(fd_inotify);
        throw 2; // Même remarque ;}
    }
    // Lancer le thread de surveillance
    thread.reset(
        new boost::thread(
            boost::bind(&InotifyWatcher::worker,
            boost::ref(*this)));
    }
```

Nous commençons par initialiser `Inotify` et installer le point de surveillance. Nous définissons aussi un tampon d'une taille de 1 octet afin de ne pas passer un pointeur nul à read. Nous nous repons sur le pointeur intelligent `boost::scoped_array` pour ce qui est de la gestion de ce tampon. Enfin nous lançons le thread par une ligne de code qui ressemble à une incantation ;) Nous utilisons la classe de `thread Boost` qui a la particularité de lancer le thread dès l'instanciation. C'est pourquoi nous n'avons pas un membre de type `boost::thread` dans notre classe mais un pointeur intelligent sur `boost::thread`. Nous instancions le thread à la fin du constructeur quand tout est initialisé et nous confions le pointeur sur l'instance à notre pointeur intelligent. Ce que nous passons au constructeur de `boost::thread` mérite également d'être expliqué.

À la base, le constructeur de `boost::thread` reçoit un pointeur sur une simple fonction. Or nous voulons que le corps de notre thread soit une méthode de notre classe, la méthode `worker` ci-dessous. Pour obtenir cela, nous passons par un adaptateur de liaison Boost qui permet d'associer la méthode avec la référence de l'instance de la classe. Le résultat est que lorsque le thread croira appeler une simple fonction il

invocera en fait this->worker comme par magie :) Regardons maintenant comment notre classe gère la libération des ressources :

```
ifyWatcher::~I NOTIFYWatcher() {
    destroy_fd();
    thread->join();
}

void I NOTIFYWatcher::destroy_fd() {
    boost::try_mutex::scoped_try_lock stl(tmtx);

    if(fd_inotify == -1)
        return;
    fini = true;
    inotify_rm_watch(fd_inotify, fd_watch);
    close(fd_inotify);
    fd_inotify = -1;
    fd_watch = -1;
}
```

Pour ne pas provoquer de conflit, nous utilisons un mécanisme de verrous. En effet fd\_inotify, descripteur du fichier I NOTIFY et fd\_watch identificateur du point de surveillance peuvent être libérés ici ou dans la routine de thread si une erreur y survient. Pour éviter tout dead lock, nous utilisons le très pratique scoped\_try\_lock de Boost qui permet, si on le souhaite, de n'acquiescer le verrou que s'il n'a pas déjà été acquis ailleurs. Dans la méthode destroy\_fd, l'appel de l'API inotify\_rm\_watch va débloquent le thread de son état d'attente efficace et finalement provoquer son arrêt. Enfin, on ferme le descripteur de fichier et toutes les ressources I NOTIFY internes telles que tampons ou autres sont alors libérées. Les essais avec mon noyau (2.16.18) montrent que le seul appel à close(fd\_inotify) ne suffit pas à débloquent l'API read du thread comme on aurait pu s'y attendre. C'est la raison pour laquelle nous prenons bien soin de tout fermer nous-mêmes.

## 5 Notre routine de surveillance

Voici le code :

```
void I NOTIFYWatcher::worker() {
    int len_read
    while(!fini) {
        len_read = read(fd_inotify,
            buffer.get(), buffer_size);
        // Si la méthode destroy_fd est appelée
        // pour arrêter, cela débloquent read
        // donc on arrive là :)
        if(!fini)
            break;
        // sinon, on bosse
        if(len_read < 0) {
            write(fd_tube_in, error_msg, strlen(error_msg));
            fini = true;
            continue;
        }
        if(len_read == 0) {
            // Ajouter un tampon élémentaire,
            // autant que nécessaire
```

```
try {
    buffer_size += one_buffer_size;
    buffer.reset(new char[buffer_size]);
    continue; // => retourner à read
}
catch(std::bad_alloc) {
    write(fd_tube_in, alloc_error_msg,
        strlen(alloc_error_msg));
    fini = true;
    continue; // => arrêter le thread
}
}
int index = 0;
while(index < len_read) {
    struct inotify_event *event;
    event = reinterpret_cast<struct inotify_event *>
        (&buffer[index]);
    try {
        std::ostringstream os;
        os << std::hex;
        os << "inotify: " << inode << " ";
        if(event->len)
            os << " " << event->name;
        os << " " << event->mask << " -> ";
        os << mask2text(event->mask);
        os << std::endl;
        write(fd_tube_in,
            os.str(), os.str().length());
        if(event->mask & IN_UNMOUNT
            || event->mask & IN_IGNORED
            || event->mask & IN_DELETE_SELF) {
            write(fd_tube_in, unmount_error_msg,
                strlen(unmount_error_msg));
            fini = true; // => arrêt
            break; // => quitter le while imbriqué
            // Sans le break, on pompe tous les
            // éventuels messages d'erreurs connexes
            // avant d'arrêter (question de choix...)
        }
        index += event_size + event->len;
    }
    catch(std::bad_alloc) {
        // on arrive ici si
        // std::ostringstream
        // lève std::bad_alloc
        write(fd_tube_in, alloc_error_msg,
            strlen(alloc_error_msg));
        fini = true;
        continue;
    }
} //while(index < len_read)
} // while(!fini)
boost::try_mutex::scoped_try_lock local_stl(tmtx, false);
if(local_stl.try_lock())
    destroy_fd();
}
```

Ce code comporte deux boucles while imbriquées. La première englobe la routine et place tout de suite le thread en attente efficace par l'appel à read. La deuxième boucle while parcourt le tampon lu par read de la manière expliquée plus haut. En outre, nous analysons le masque d'événements. Si nous y trouvons des événements signifiant que le point de surveillance est effacé ou démonté, nous arrêtons le thread. (Mais pas l'application. Vous devrez presser [return] dans le terminal de lancement pour cela). Sinon nous traduisons l'événement en texte que nous écrivons dans le tube. Le lecteur trouvera la routine de traduction sur le site.

## 6 Le corps principal de l'application

Ce code est le plus simple :

```
include <iostream>
#include <unistd.h>
#include <fcntl.h>
#include "notifywatcher.hpp"

int tube[2];
volatile bool app_worker_fini = false;

void app_worker() {
    // Passer en lecture non-bloquante
    // du côté 'sortie' du tube
    fcntl(tube[0], F_SETFL,
        fcntl(tube[0], F_GETFL) | O_NONBLOCK);
    char c;
    int len_read;
    // flag, uniquement pour soigner l'affichage
    bool flag = false;

    while(app_worker_fini) {
        len_read = read(tube[0], &c, 1);
        if(len_read == 1) {
            if(flag)
            {
                write(STDOUT_FILENO, "\n", 1);
                flag = true;
            }
            write(STDOUT_FILENO, &c, 1);
        }
        else {
            flag = false;
            write(STDOUT_FILENO, " ", 1);
            usleep(300000);
        }
    }
}

int main(int argc, char** argv)
{
    boost::scoped_ptr<INotifyWatcher> inw;
    std::cout.sync_with_stdio(false);
    if(argc != 2) {
        std::cout << "Donnez le nom d'un répertoire ou d'un "
            " fichier à surveiller" << std::endl;
        std::cout <<
```

```
        "Exemple: ./DemoNotify /home/fred" << std::endl;
        exit(EXIT_FAILURE);
    }
    if(pipe(tube) == -1) {
        std::cout << "Impossible d'ouvrir le tube de "
            "communication - arrêt" << std::endl;
    }
    try {
        inw.reset(new INotifyWatcher(argv[1],
            tube[1], IN_ALL_EVENTS));
    }
    catch(int e) {
        switch(e) {
            case 1:
                std::cout << "Impossible d'initialiser "
                    " INotify - Arrêt" << std::endl;
                break;
            case 2:
                std::cout << "Impossible d'ajouter le "
                    "point de surveillance "
                    " INotify - Arrêt" << std::endl;
                break;
            default:
                std::cout << "Problème inconnu - Arrêt"
                    << std::endl;
                break;
        }
    }
    close(tube[0]);
    close(tube[1]);
    exit(EXIT_FAILURE);
}
std::cout << "Application lancée - Appuyez sur "
    " [return] pour arrêter" << std::endl;
boost::thread t(app_worker);
std::cin.get();
inw.reset();
app_worker_fini = true;
t.join();
close(tube[0]);
close(tube[1]);
return EXIT_SUCCESS;
}
```

Deux points sont particulièrement intéressants. Le premier est la façon dont nous arrêtons l'application. Nous appelons indirectement le destructeur de la classe de surveillance (`inw.reset()`). Ce destructeur attend la fin du thread worker avant de rendre la main. Nous pouvons donc ensuite arrêter le thread principal de l'application en toute sécurité. Une fois son arrêt effectif, nous fermons les extrémités du tube. C'est terminé. Le second point intéressant se situe dans la routine de thread `app_worker`. Il s'agit de l'appel à l'API `fcntl` qui permet de passer en mode de lecture non bloquante du côté sortie du tube. A bientôt pour d'autres aventures dans les profondeurs de Linux.

### Lexique

**Inode** : bloc contenant les méta-données relatives à un fichier: date de création, d'accès, taille, attributs et liste des blocs de données.

■ Frédéric Mazué - [fmazue@programmez.com](mailto:fmazue@programmez.com)

# L'information à la carte

# 1

## 1 an ECO

Recevez le magazine  
chaque mois  
et économisez 20 €

**11 Numéros**

Prix au numéro : 65,45 €

**45 €**

(Prix France métropolitaine)

**-30%**



# 2

## 1 an TOUT NUMERIQUE

Lisez chaque mois le  
magazine seul  
**au format PDF**  
(téléchargement)

Inscription :  
[www.programmez.com](http://www.programmez.com)

**11 Numéros**

**35 €**

Tarif Monde entier

**-46%**



# 3

## 1 an ETUDIANT

Vous devez justifier  
de votre statut d'étudiant.  
**Economisez 26 €**

**11 Numéros**

Prix au numéro : 65,45 €

**39 €**

(offre réservée France métropolitaine)

**-40%**



# 4

## 2 ans

Abonnez-vous  
pour 2 ans et recevez  
le livre « Windows Vista »,  
Guide de l'Administration  
Microsoft Press  
William R. Stanek

**22 Numéros**

Prix au numéro : 130,90 €  
+ livre 39 €

**90 €**

(offre réservée France métropolitaine  
dans la limite des stocks disponibles)



**ECONOMISEZ 80€**

**OUI, je m'abonne !** ou abonnement en ligne : [www.programmez.com](http://www.programmez.com)

- ABONNEMENT 1 an ECO** au prix de 45 € TTC. Tarif France métropolitaine.  
Tarifs hors France métropolitaine : CEE et Suisse : 51,83 € - Algérie, Maroc, Tunisie : 55,95 € - Canada : 64,33 € - Tom : 79,61 € - Dom : 62,84 € - Autres : nous consulter
- ABONNEMENT 1 an ETUDIANT (11 numéros)** au prix de 39 € TTC. Offre limitée à la France métropolitaine.  
Photocopie de la carte d'étudiant obligatoire
- ABONNEMENT 2 ans + livre «Windows Vista» (22 numéros)** au prix de 90 € TTC. Offre limitée à la 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 :**  
Programmez ! - Service Abonnements - 22 rue René Boulanger - 75010 Paris.  
[abonnements.programmez@groupe-gli.com](mailto:abonnements.programmez@groupe-gli.com)

**Programmez!**  
LE MAGAZINE DES DEVIS PROFESSIONNELS

Offre limitée,  
valable jusqu'au  
31 mai 2007

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.

## Libéralisme virtuel et capitalisme réel : Zenghtu pulvérise Blizzard

*World of Warcraft a trouvé son maître... et il est bien sûr chinois ! Le nom du vainqueur est la compagnie Zenghtu, basée à Shanghai, avec son MMORPG Zenghtu Online.*

Zenghtu vient en effet de dépasser en nombre, sur le territoire chinois, les profits pourtant déjà spectaculaires de Blizzard. D'abord, en affichant un pic de connexions de 860.000 joueurs sur Zenghtu Online face à 680.000 sur Wow. Ensuite, et surtout, avec 21 millions de \$ de revenus et 15 millions de \$ de bénéfices par mois pour Zenghtu sur le dernier trimestre 2006, contre respectivement 12 et à peine 7 pour le " pauvre " Blizzard... Mais comment ?

### Entrez, c'est gratuit !

Non, pas en faisant payer un abonnement mensuel de connexion, à la mode de Blizzard :



Zenghtu est d'accès absolument gratuit ! Mais c'est une fois à l'intérieur qu'il faut ouvrir sa bourse, car ce sont les accessoires supplémentaires des joueurs qui s'achètent en ligne : une magie irrésistible, une épée vraiment dévastatrice, un pouvoir totalement décisif se payent cash en yuans auprès de l'éditeur. De plus, le jeu est extrêmement sauvage et autorise quasiment toutes les agressions entre personnages – bref, il est presque impossible de survivre pour les pauvres. De nombreux joueurs dépensent donc l'équivalent de 130 \$ par mois pour bien s'équiper et espérer survivre chez Zenghtu ! Certes, l'idée des équipements payants en MMORPG n'est pas à proprement parler nouvelle. Mais à ce niveau, elle est " révolutionnaire " ! Car jusqu'à présent, il s'agissait d'une pratique indépendante du jeu lui-même comme de l'éditeur, c'est-à-



dire de transactions privées entre joueurs, considérées comme illégales.

### Développement à la demande...

Il faut d'ailleurs savoir que pour le développement de son jeu, Zenghtu a fait appel à Shanda Entertainment, autre compagnie chinoise spécialisée dans les univers virtuels. Des spécialistes en effet : sur un MMORPG précédent, un des ex-managers de Shanda a été condamné à 5 ans de prison pour avoir vendu à son propre compte plus de 250.000 \$ d'équipements virtuels ! Facile, il suffisait de les développer à la demande... Pourtant, Zenghtu Online continue aujourd'hui son ascension, même si le gouvernement et diverses associations chinoises s'indignent contre cette addiction fort coûteuse.



L'histoire des jeux vidéo retiendra donc que c'est une entreprise du plus grand pays communiste qui a inventé le premier MMORPG hyper-capitaliste à deux vitesses : tu payes, tu gagnes et tu vis, tu radines, tu perds et tu meurs ! À côté, Blizzard ressemble à une association à but non lucratif.

## N-Gage : la plate-forme s'élargit

*Nokia mise désormais à fond sur sa plate-forme de jeu mobile N-Gage avec un panel d'éditeurs élargi, de nouveaux jeux, de nouveaux outils de développement, et des procédures améliorées de tests.*

Le dispositif de développement mis en place autour de la N-Gage a été dévoilé par Nokia à la dernière Mobile GDC (Game Developer Conference) de San Francisco. Il se manifeste d'abord par le ralliement à cette plate-forme d'éditeurs ludiques aussi importants que Glu, E. Arts, THQ Wireless, Indiagames, et l'incontournable Gameloft, soit une vingtaine de titres au total disponibles à courte échéance.

### Poker mobile

Gameloft se taille la part du lion avec 7 jeux annoncés d'ici la fin de l'année, des classiques pour la plupart, de Dogz (simulation de chien domestique) à Brain Challenge (entraînement cérébral comme sur Nintendo DS) en passant par Brother in Arms, de vrais combats en temps réel avec de la vraie 3D – quasiment comme sur un desktop ou une console – ou Asphalt 3D, du pilotage urbain sauvage aux bruitsages hallucinants. Mais on devra sans doute le titre le plus explosif à l'éditeur Glu Mobile, dont le World Series of Poker Texas Hold'em va exploiter au mieux la vogue actuelle et mondiale du poker : l'application offrira, depuis sa N-gage, des tables de 8 joueurs distants, des tournois en réseau, et on pourra même affronter des champions professionnels du WSOP comme Chris Fergusson ou Scott Fischman, convoqués lors de sessions spéciales. Impressionnant.

### Un SDK pour du C++ " familier "

Dans la foulée, Nokia vient aussi de rendre disponible son dernier kit de développement logiciel, le N-Gage SDK 1.0, qui offre aux développeurs une base de code unique, gérant



bien sûr le jeu réseau, pour tous les appareils Nokia de type S60 3ème édition. Pour Mark Ollila, des divisions Technology et Games Publishing, "c'est la pierre angulaire de notre nouveau système de développement. Nous apportons ainsi, pour le jeu mobile en réseau, un environnement C++ tout à fait familier. Soit, grâce à ce canal unique, un gain de temps et de budgétisation effectif." Un espace réservé aux développeurs vient d'être mis en place sur le net (<http://forum.nokia.com/games>) qui propose les outils, le support, les mises à jour et la documentation nécessaires. Enfin, Nokia n'oublie pas pour autant le Java et surtout sa solution de jeu SNAP Mobile, regroupant à la fois logiciel, hébergement, maintenance et gestion de communauté en ligne pour tous les téléphones compatibles Java MIDP 2.0. Le constructeur propose donc maintenant un test de conformité SNAP. L'idée, plutôt astucieuse est d'associer au programme de certification classique une procédure simultanée d'autotest du côté de l'équipe de développement et de l'éditeur, qui valide par la même occasion la compatibilité et la fiabilité du produit.

## ACTUS

## Vista &amp; Xbox-Live : l'unification

Une barrière ancestrale entre ordinateurs et consoles vient de tomber, et c'est le premier effet ludique Vista : dès le 18 mai, le réseau de jeu Xbox-Live s'étend désormais aux joueurs PC sur Vista à travers trois premiers titres spécialement adaptés. Halo 2 pour commencer, suivi dès juin de Shadowrun, shoot en 3D éminemment interactif particulièrement à même de réunir les utilisateurs des deux plates-formes. Enfin, vers septembre, la communion sera complète autour d'un jeu de cartes hyper grand-public, le UNO. C'est facile, mais efficace.



Dans la pratique, les joueurs sur PC vont pouvoir s'abonner au service Games for Windows-Live spécifique à Vista pour rejoindre les joueurs du réseau Xbox-Live. Le service est quasiment identique pour le coût comme pour la procédure d'accès ou les options – dont par exemple le très apprécié chat direct et vocal en cours de jeu. L'utilisateur n'aura d'ailleurs qu'un seul profil de joueur, peu importe depuis quelle plate-forme il se connecte. Les serveurs suivront-ils ? A tester...

## Jouer au volant ?

Micro-Application entre dans la cour des grands : l'éditeur vient de signer un contrat de développement de plusieurs de ses titres avec Nintendo, sur Nintendo DS pour être plus précis. Des jeux, bien sûr, mais la première adaptation concerne surtout le hit des hits de Micro-App : Code de la Route, programme d'apprentissage et d'examen, conforme à la toute dernière



formule du permis de conduire. Une annonce significative de l'évolution de la DS, et plus globalement des consoles, vers les applications pratiques et les services.

## L'OS pluridisciplinaire des smartphones

Symbian gagne encore du terrain sur le marché mobile. Le Symbian OS est en effet le système leader d'exploitation des smartphones, dont le marché est estimé proche du milliard d'appareils dans les 4 années à venir ! Blue Stark Technology vient donc de se rallier lui aussi à l'OS Symbian. Le développeur de la plate-forme MachBlue entend bien concourir ainsi à une certaine et profitable uniformisation, d'autant que les caractéristiques de l'OS correspondent typiquement à ses besoins – ouverture simultanée vers tous les médias, depuis la TV sur mobile jusqu'à la diffusion de contenus Rich Media et Flash en passant évidemment par les jeux...

## SDK Logitech pour accessoires 3D

3Dconnexion, filiale de Logitech, vient de mettre un kit de développement à disposition gratuite permettant d'implémenter les fonctions 3D en interfaçage de la plupart de ses accessoires, souris, joysticks, claviers, etc. Au-delà des outils, le nouveau SDK comprend des applications "prêtes à l'emploi" en code-source (zoom, rotation, effets panoramiques standard essentiellement) et des exemples de code aussi bien en C++ qu'en Java ou en Visual Basic. [www.3dconnexion.com/support](http://www.3dconnexion.com/support)

## Architecture de l'information pour le web

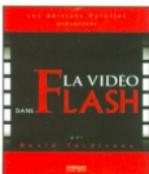


- » **Difficulté :** \*\*\*
- » **Editeur :** O'Reilly
- » **Auteur :** collectif
- » **Prix :** 42 €

L'information explose en diversité, en volume. Le Web 2.0 ne fait qu'accroître la tendance. Comment contrôler

une telle masse de données ? Comment les gérer ? Comment les exploiter ? Bref, comment trouver la bonne information ? Cet ouvrage a pour but de vous aider à définir l'architecture de vos informations qui mêle base de données, ergonomie, stratégie globale. Il faut prendre en compte 3 éléments de base : le contexte, l'utilisateur, le contenu. Et pour réussir son architecture, il faut notamment comprendre les besoins de l'utilisateur. Ensuite, il faut comprendre les différentes structures possibles. On y apprendra le rôle de l'étiquetage, la navigation, la recherche (et sa nécessité), l'interface, etc. La bonne idée est une partie plus pratique avec de l'éthique ainsi que l'offre et les technologies disponibles aujourd'hui. Les auteurs concluent par des cas pratiques. Un must pour tous les responsables et développeurs !

## La vidéo dans Flash



- » **Difficulté :** \*\*
- » **Editeur :** Eyrolles
- » **Auteur :** David Tardiveau
- » **Prix :** 19,90 €

La vidéo sur internet explose. Mais encore faut-il savoir comment la diffuser ?

Flash est devenu un vecteur de déploiement important notamment grâce à son format vidéo : FLV. L'auteur montre, étape par étape, la préparation de sa vidéo et l'encodage, avec le code nécessaire pour personnaliser l'affichage. Puis, on passe par la phase d'intégration dans un fichier SWF. On y trouve tous les événements possibles servant à la gestion de la vidéo. Pour aller encore plus loin, on apprend comment créer une vidéo interactive, avoir des effets spéciaux et surtout, au final, comment diffuser son travail ! Le livre se veut avant tout pratique, les nombreux exemples aident à bien comprendre chaque phase du projet.

## Apprenez à programmer avec Visual Basic 2005 Express

- » **Difficulté :** \*\*
- » **Editeur :** Dunod
- » **Auteur :** Dominique Maniez
- » **Prix :** 15 €

Destiné à ceux qui veulent découvrir .Net, cet ouvrage propose une approche progressive. On y découvre les fonctions de base du langage et l'outil de développement. Vous vous familiariserez avec sa syntaxe et avec les notions de mots-clés, de variables, ou d'opérateurs. Vous apprendrez à maîtriser les tests conditionnels, les boucles et les fonctions et à écrire rapidement votre premier logiciel. Les exemples sont téléchargeables sur le site de l'éditeur.

## Sécurité informatique : principes et méthode



- » **Difficulté :** \*\*\*
- » **Editeur :** Eyrolles
- » **Auteur :** collectif
- » **Prix :** 35 €

L'insécurité informatique demeure une menace. Comprendre les causes de cette insécurité, c'est comprendre les mécanismes et les contre mesures applicables. Les auteurs rappellent dans un premier temps les principales failles que l'on rencontre dans un système d'information. Puis on passe par les solutions techniques possibles. L'ouvrage rappelle, avec justesse, le rôle clé du chiffrement. Le réseau concentre une grande partie des techniques de protection. La bonne idée est d'avoir consacré une partie du livre à la charte des utilisateurs et de l'administrateur. Un

## Windows Vista Guide de l'administrateur



- » **Difficulté :** \*\*\*
- » **Editeur :** Dunod
- » **Auteur :** William Stanké
- » **Prix :** 39 €

Avec Vista, l'administration Vista change assez profondément. Dans cet ouvrage, l'auteur dis- séque les nouvelles fonctions, les réflexes à avoir, les nouvelles politiques de gestion à mettre en place. Pédagogique, les instructions sont montrées pas à pas. Et des tableaux permettent de se repérer rapidement. Le réseau se taille la part du lion mais aussi la configuration Web, les portables ou encore le dépannage et optimisation système. À avoir près de son PC...

ouvrage sans surprise, dommage que le poste de travail ne soit pas assez mis en valeur ainsi que le rôle du développeur. Mais il est toujours bon de rappeler les fondamentaux.



- » **Spip 19**
- » **Difficulté :** \*\*
- » **Editeur :** Eyrolles
- » **Auteur :** collectif
- » **Prix :** 19 €

Solution de gestion de contenu open source, Spip est aujourd'hui une référence incontournable du marché. Les auteurs proposent de créer de A à Z votre site web adossé au système de contenu Spip. On débute par les outils, l'hébergement, le nom de domaine et les fondamentaux des langages web. L'autre défi est de créer son premier site Spip en une après-midi... On peut aller plus loin avec quelques fonctions avancées, et notamment le référencement. Une bonne initiation, agréable à lire.

## Ajax : créez votre application Internet à interface riche



- » **Difficulté :** \*\*
- » **Editeur :** Eni éditions
- » **Auteur :** Florent Leroux
- » **Prix :** 39 €

Vous voulez créer vos applications Web 2 en Ajax ? Voici un ouvrage d'apprentissage sur Ajax pour démarrer en douceur. On y parle HTML, DHTML, manipuler les données en Ajax, créer des interfaces riches et dynamiques. *Chez le même éditeur : Java 6, collection les TP Informatiques, 27 euros.*



**Déploiement Windows Vista (Micro Application, 47,32 €) :** un des soucis en entreprise est le déploiement des systèmes sur les postes. Avec Vista, Microsoft a amélioré le processus de déploiement et les mécanismes. L'objectif est d'apprendre tous les secrets pour mener à bien le déploiement sur des centaines de machines. On y retrouve des études de cas concrets, la méthodologie à mettre en œuvre, comment bien utiliser Windows PE ou ImageX. Un ouvrage complet qui intéressera les utilisateurs avancés et les administrateurs !



## serveurs dédiés DUO /

**Vous n'avez pas à nous prier  
pour vous offrir deux fois plus  
de performance !**

### NOUVEAU

### Serveurs dédiés DUO



Pour les professionnels les plus exigeants, AMEN lance la nouvelle gamme de serveurs dédiés DUO basée sur des processeurs double coeur, disques durs en RAID, pour vous offrir 2 fois plus de puissance.

#### DUO 1000 ► 99 € ht/mois\* (118,40 € ttc/mois\*)

AMD Opteron 1210 - 2x1,8GHz - RAM 1GB  
Disque dur 2x160GB - Raid Soft  
2 adresses IP - Interface Plesk 8 jusqu'à 100 domaines - Trafic illimité

#### DUO 2000 ► 149 € ht/mois\* (178,20 € ttc/mois\*)

AMD Opteron 1212 - 2x2,0GHz - RAM 2GB  
Disque dur 2x200GB - Raid 1 matériel  
4 adresses IP - Interface Plesk 8 jusqu'à 300 domaines - Trafic illimité

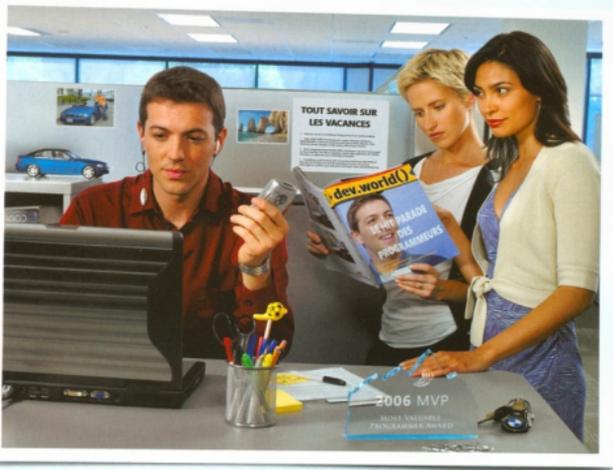
#### DUO 4000 ► 199 € ht/mois\* (238,00 € ttc/mois\*)

AMD Opteron 1214 - 2x2,2GHz - RAM 4GB  
Disque dur 2x250GB - Raid 1 matériel  
6 adresses IP - Interface Plesk 8 jusqu'à 300 domaines - Trafic illimité

Compatibles 

Nous avons foi en un idéal de services, surtout lorsqu'il vous permet de bénéficier des dernières avancées techniques : architecture réseau redondée, bande passante dédiée 2GB, haute disponibilité (99,9%), assistance technique par mail et téléphone 6j/7<sup>h</sup>). Quant à notre 'Garantie satisfait ou remboursé'<sup>(2)</sup>, elle vous permettra d'atteindre la sérénité absolue. Si vous croyez au web, vous croirez en nous.

► Pour plus de renseignements **0 892 55 66 77** (0,34€ / min) ou **www.amen.fr**



**Visual Studio® 2005.** La différence saute aux yeux.

Faites la différence. Migrer plus rapidement vers Visual Basic® 2005 vous permet de capitaliser sur vos compétences actuelles tout en gardant la main sur les projets ambitieux. Vous avez accès à plus de 400 fonctionnalités qui vous simplifient le code au quotidien. Vous pouvez ainsi vous concentrer sur l'essentiel. Retrouvez les 400 différences sur [www.microsoft.com/france/vs2005](http://www.microsoft.com/france/vs2005)

Microsoft   
**Visual Studio 2005**