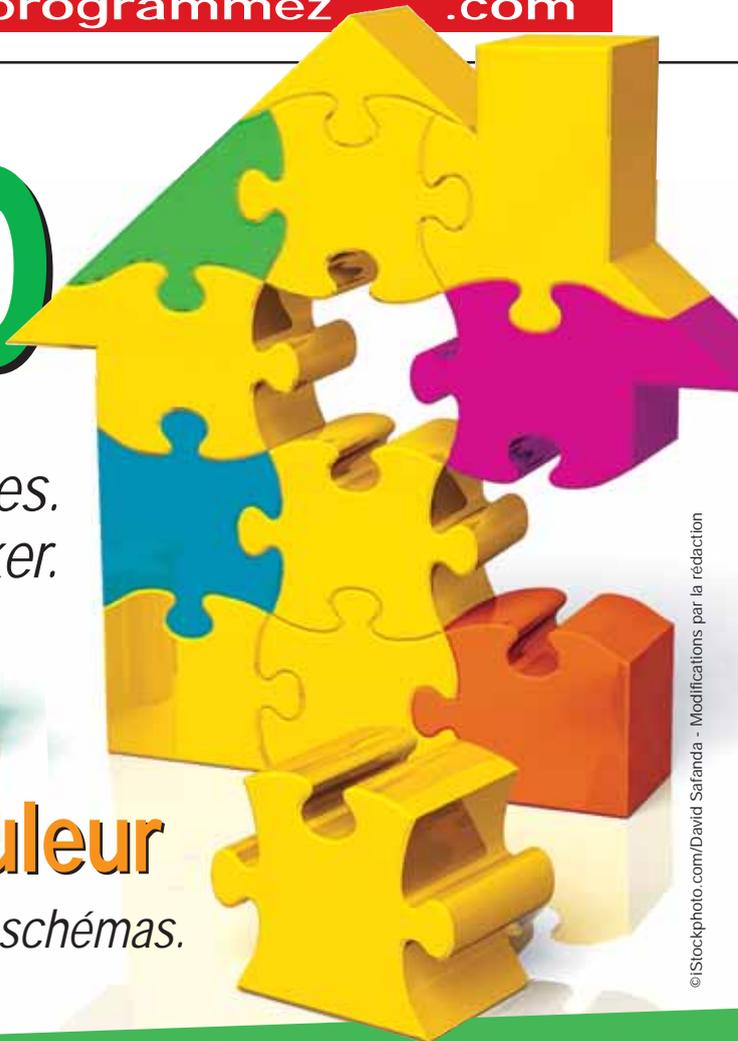


Mashup

*L'art de composer,
assembler et présenter les données.
Découvrez Popfly, Intel Mash Maker.*

XML sans douleur

*Maîtriser les styles XSLT. Bien utiliser les schémas.
XML 2.0 : le futur lointain ?*



©iStockphoto.com/David Safanda - Modifications par la rédaction

Mettez-vous aux TESTS!

*Les bonnes pratiques, les outils.
Testeur : un métier d'avenir ?*

OUTILS

Netbeans 6.1 : tout pour Java et Ruby !
Microsoft Expression 2.0 : les nouveautés en détail

JavaOne 2008

Les dernières annonces
du monde Java

INTERNET

La nouvelle guerre
des navigateurs

.NET

Manipuler les règles
métier dans la DLR

C++

Bien démarrer
la programmation Qt

WINDOWS

Manipuler
Windows Mail

M 04319 - 109 - F: 5,95 €



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

PYTHON
Optimiser votre
code avec Pyrex



**INCLUS EN STANDARD
DANS WINDEV 12 :**

WINDEV®

**DÉVELOPPEZ
10 FOIS PLUS VITE**



Développez de superbes IHM sans compétences graphiques grâce au générateur d'IHM (avec correcteur d'IHM) et aux gabarits fournis.



Environnement intégré (IDE & ALM)

- Tout est en **français**
- Hot Line personnalisée **gratuite**
- Déploiement libre et **gratuit**
- Crée des .EXE sécurisés, des Web Services, des applications .NET, des applications Java...
- 32 bits, **64 bits**
- **Compilation JIT**
- Code multi-plateforme compatible **Mobile et Internet**
- Génération d'application **Java** à la demande
- Fonctionne en **TSE** et Citrix
- Générateur automatique d'**IHM**, avec charte graphique automatique. Création d'IHM «Vista» & «Vista Like» par utilisation de gabarits fournis
- Tous les Windows : 98, 2000, NT, XP, Vista, 2008...
- Générateur d'**Etats et Requêtes gratuit**, création de **PDF**, code-barres, étiquettes. Fond de page PDF
- **Automatique** dans chaque application: menu d'export vers Word, Excel, OpenOffice, XML, PDF; Graphiques 3D; Historique de saisie,... Envoi de mail, Macros utilisateur
- Real-**RAD** : Générateur d'applications complètes
- AAA (Architecture Automatisée d'Application): créez votre propre RAD (Patterns)
- **HyperFileSQL**, base de données Client/Serveur, Locale et Mobile sous Windows et Linux (version illimitée, **libre et gratuite**); Gère 4 millions de Teras
- Accès à toutes les Bases de Données tierces: **Oracle, AS/400, SQL Server, DB2, MySQL, Access, xBase**, etc...
- **Réplication** multibases assistée
- XML natif
- Accès natif à **SAP R/3**, Lotus Notes, LDAP, Outlook,...
- Centre de suivi du **planning** d'équipes
- Centre de suivi des retours utilisateurs
- Tableau de Bord de **suivi de projet**
- Centre de Modélisation **UML**, Merise et Souple; code généré depuis l'analyse, reverse engineering
- **Dossier** automatique : analyse & programmation
- Création et utilisation de **composants** ; 3-tiers
- Règles métier; Gestion des exigences
- Langage de 5^e Génération **L5G**, qui élimine 90% du code
- **Ouverture** aux L4G et L3G: C++, C#, Java, VB, Cobol...
- Import d'applications Access et VB
- Fonctions Domotique (norme X10)
- Gestion liaison série RS 232, parallèle et **USB**
- Fonctions Bluetooth,
- Fonctions réseau **SNMP**
- Fonctions TAPI, OPC, FTP, HTTP, Socket, Twain, API, DLL,...
- Editeur de code intelligent, avec test immédiat sans recompilation
- **Tests unitaires** de code et d'IHM automatisés, Editeur visuel de tests de non-régression
- Refactoring
- Outil de **versionning** sophistiqué (gestion des sources)
- Débogueur puissant: threads, composants,... **Débogage à distance**
- **Profiler**, pour optimiser la vitesse du code
- Multilingue automatique: jusqu'à **20 langues**
- Générateur d'aide CHM
- Fonctions Multimédia (image, son, vidéo)
- Générateur d'**Installations en 1 clic**, et mises à jour automatiques (local, à distance, via Internet)
- Autoformation en 1 semaine (**manuel livré**)

www.pcsoft.fr

Demandez le dossier gratuit (200 pages + 1 DVD)
Inclut la **version Express** (gratuite) et 112 Témoignages
détaillés. Tél: 04.67.032.032 ou info@pcsoft.fr

Fournisseur Officiel de la
Préparation Olympique



XII



DVD-Rom 109 PROGRAMMEZ !

JAVA

NetBeans 6.1 - La toute dernière version de l'environnement multi-langage. Support de Java, JavaScript, Ruby. Edition Windows
Qt Open Source Edition for C++ Developers - Développez vos interfaces graphiques multi-plates-formes avec la dernière version de la librairie Qt. Edition Windows - utilisation non commerciale
QF-TEST - Outil d'automatisation de tests de non-régression et de charge des interfaces utilisateur - 30 jours

Base de données

MySQL Workbench 5.0.21 - Environnement graphique pour créer rapidement vos bases de données MySQL. Edition Windows
Firebird 2.1 - La toute dernière version stable de Firebird, base de données robuste et fiable en open source. Editions Windows 32 et 64 bits

XML

Altova XMLSpy 2008 enterprise edition - Editeur et environnement de développement XML. Edition Windows - 30 jours
API Stax - Gérer et manipuler XML dans vos applications Java avec la puissante API Stax

.Net

WCF Visualizers 1.5 bêta - Plugin Visual Studio
Moonlight 1.1 - Découvrez le Silverlight open source du projet Mono fonctionnant sous Linux !
Monodevelop 1.0 - La version finale 1.0 du très attendu IDE .Net open source. Idéal pour programmer ses applications Mono.

C++

Qt Open Source Edition for C++ Developers - Développez vos interfaces graphiques multi-plates-formes avec la dernière version de la librairie Qt. Edition Windows - utilisation non commerciale

Windows

Powershell 2.0 CTP SDK - Shell en ligne de commande surpuissant pour gérer et manipuler votre système ! Pré-version de la version 2.0. Edition Windows

Python

Pyrex 0.9.6.2 - Langage à la syntaxe voisine de Python et compilant en code natif
Psyco - Compilateur Just in Time pour Python
Python 2.5.2 - Dernière version du langage de programmation-Versions Windows et Linux.

> Actus

L'actualité en bref	8
Agenda	9

> Evénement

Intel Software Conference : l'avenir du multi-core.....	12
Java FX en vedette	14

> Test

NetBeans 6.1 : l'ultime IDE ?	18
-------------------------------------	----

> Gros Plan :

Mashup, l'art de composer, assembler et présenter les données

Mashup : où, quand, comment, pourquoi ?	22
Popfly : créer votre mashup sans une ligne de code !	27
Intel Mash Maker : l'autre outil de mashup !	30

> Dossier

Mettez-vous aux tests [1re partie]

Les meilleures pratiques du test logiciel	33
Les différents types de tests et les tests de régression.....	35
Les tests unitaires et PHP	38
Profession : testeur de logiciels.....	40
Les outils de tests : un panorama du marché	42

> Dossier :

XML sans douleur... ou presque [2e partie]

XML 2.0 : le futur du XML est-il en marche ?	45
XSLT sans douleur	47
Validation des documents XML	50
La sérialisation XML	53

> Développement Web

Microsoft Expression 2 : une évolution	56
Echangez vos données grâce à JSON	60
La prochaine génération de navigateurs au banc d'essai	62

> Code

Introduction à la programmation QT (1re partie)	66
Comment manipuler vos règles métier avec la DLR.....	70
Optimisez votre code Python avec Pyrex et Psyco.....	74
Constituer et expédier un e-mail depuis Windows Vista.....	77

> Temps libre

Les livres du mois	82
--------------------------	----





Information Management

_LE JOURNAL DE NOTRE INFRASTRUCTURE

_94^e JOUR : Impossible de retrouver les données clients dont nous avons besoin. Comment exploiter nos informations si elles ne sont pas facilement accessibles?

_Gilles a installé une machine à téléporter : il prétend qu'il peut transférer les gens instantanément dans leur espace de données... permettant au marketing de se libérer des données discordantes et hétéroclites.

_97^e JOUR : La réponse est ailleurs. Le logiciel IBM InfoSphere Master Data Management Server nous apporte une vue unifiée de nos informations clients en temps réel. Nous alimentons les utilisateurs, les processus et les applications avec des données fiables. Et nous obtenons de meilleurs résultats pour l'entreprise.

_Aïe! Je viens de téléporter Gilles en Espagne. ¡Lo siento, mi amigo!*

Pour accéder rapidement à une information fiable :
IBM.COM/TAKEBACKCONTROL/DATA/FR



XMLSpy – Outils pour le développement de Schémas XML

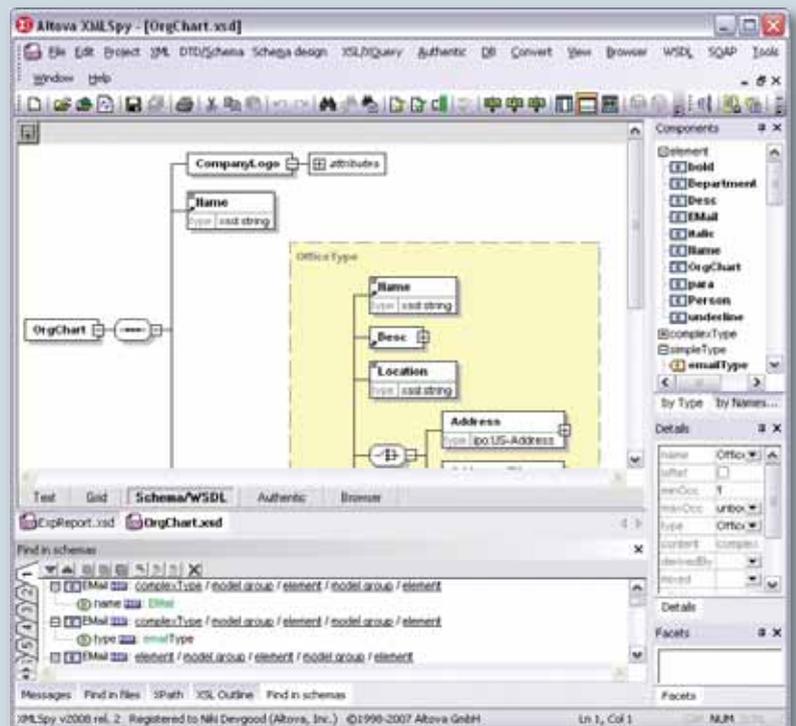
Créez des Schémas XML avancés, rapidement et facilement avec l'outil de représentation graphique que vous offre Altova XMLSpy, l'éditeur XML le plus populaire au monde.

- **Prise en charge des fichiers XML de grande taille**
- **Puissante fonction de recherche et de remplacement à base schématique:**
 - Recherche sur un ou plusieurs schémas
 - Options de recherche et de remplacement larges ou granulaires

**Nouveautés
dans la Version
2008 v2**

Grâce à ses représentations graphiques puissantes et faciles à utiliser, à ses aides à la saisie et à ses assistants, XMLSpy rend le développement XML accessible aux débutants comme aux développeurs confirmés. En travaillant dans un environnement graphique, vous pouvez vous concentrer sur les éléments sémantiques du schéma et laisser les détails syntaxiques du langage XML à XMLSpy.

- Editeur graphique de Schémas XML
- Conversion entre DTD et schéma des bases de données
- Conversion en UML
- Génération d'échantillons d'instances XML
- Création de schémas à partir d'une collection d'instances XML
- Validation et traitement d'erreurs avancés
- Fonction de recherche et de remplacement détaillée sur un ou plusieurs schémas
- Génération de code Java, C#, C++ à base d'un schéma
- Documentation complète sur les schémas en HTML ou Microsoft® Word
- Intégration d'un agent schématique (SchemaAgent®) puissant pour faciliter le développement de schémas complexes



Découvrez par vous-même pourquoi XMLSpy est le logiciel de choix pour plus de 3 millions de développeurs XML – Téléchargez gratuitement une version d'essai de 30 jours sur www.altova.com



Nuages à l'horizon

Il faut bien avouer que le printemps 2008 fut un cru informatique de haute volée mais pas forcément dans le sens du vent que l'on aurait imaginé. Le premier électrochoc fut l'annonce de la vente (définitive !) des environnements de développement CodeGear, filiale de Borland, à l'éditeur Embarcadero. Et pour une somme dont la faiblesse m'a surpris : environ 23 millions de dollars...

C'est réellement une page de l'histoire informatique et du développement qui se tourne. Borland, cofondée par le français Philippe Kahn, fut pour beaucoup d'entre nous le premier outil de développement. Je pense en particulier à Turbo Pascal ou Turbo C. L'histoire de l'éditeur débuta il y a 25 ans, en 1983... Il faut bien avouer qu'aujourd'hui il est difficile de vendre des IDE tellement ils sont devenus des outils de "commodités", surtout dans le monde Java avec Eclipse et Netbeans. Nous ne savons pas à quelle sauce sera mangé CodeGear et quel sera l'avenir de ses différentes gammes d'IDE mais il paraît difficile de tout conserver. Le repreneur va-t-il miser sur les IDE à fort potentiel de croissance pour PHP ou Ruby, voire .Net ou encore sur l'univers des applications riches de type RIA ? L'avenir nous le dira.

La seconde annonce qui nous a profondément attristés, au sein de la rédaction, est la fermeture aussi rapide que brutale d'O'Reilly France. Cet éditeur publiait des ouvrages de qualité qui devenaient la plupart du temps de facto des références pour le développeur. Bien entendu, la conjoncture de l'édition n'est pas facile mais nul ne s'attendait à une décision aussi radicale. Quid des traductions françaises ? A court terme, rien ne semble prévu.

Nous sommes tout de même à un tournant significatif dans l'outillage du développeur et des logiciels en général. Si durant des années, l'éditeur dictait ses volontés face aux utilisateurs, aujourd'hui, avec l'open source, le logiciel libre, le besoin de standards, le rapport de force tend à changer de camp. Aux éditeurs de s'adapter, de mieux satisfaire les utilisateurs et les développeurs. Cependant, on peut constater un nouveau mouvement plus qu'intéressant : le "nuage informatique", ou cloud computing.

Du On Demand au Cloud Computing

Vous me direz "c'est pas franchement neuf. C'est comme si on nous ressortait le Network Computer, le fameux client léger.". Il est vrai que les constructeurs, jamais à court d'idée recyclée, nous ont ressorti le client léger, mais cette fois habillé de virtualisation. Oh que c'est beau ! Mais revenons à notre sujet. On nous martèle depuis quelques années "l'informatique à la demande"...! Le mouvement est maintenant enclenché et toutes les annonces autour du S+S, du SaaS, des services en ligne et hébergés, de l'infrastructure dématérialisée, nous montrent clairement l'évolution logicielle dans les prochaines années (en fait dès 2009) : le "nuage informatique". Le développeur va jouer un rôle certain, même si, aujourd'hui, il est urgent de ne pas se précipiter : API et outils immatures ou absents, peu ou pas d'interopérabilité, offre en constante évolution. Attention au choix de la plate-forme !

Terminons sur ce qui peut apparaître comme les trois piliers du développeur : tests, mashup, XML, les trois dossiers de ce numéro. XML car il est partout. Mashup, car c'est plus simple pour agréger des données et trouver l'information qui nous intéresse. Tests car il faudrait s'y mettre sérieusement : partout, tout le temps !

<xsl:template match="/François Tonic">



Pro programmez!

LE MAGAZINE DU DÉVELOPPEMENT

Rédaction : redaction@programmez.com

Directeur de la Rédaction : Jean Kaminsky

Rédacteur en Chef : François Tonic

Ont collaboré : F. Mazué, L. Guillois, F. Dewasmes, V. Perdereau, G. Delamarre, J-B. Boisseau.

Experts : E. Hauchecorne, M. Briec, A. Angelo, D. Cohen-Zardi, F. Darphin, B. Homes, B. Delacretaz, A. Brillant, N. Biedermann, A. Peter, A. Vannieuwenhuze, A. Dupuis, O. Dessyn, E. Groise, M. Sperberg-McQueen.

Dessin : Michel Piedoue

Illustration couverture : ©iStockphoto.com/David Safanda - Modifications par la rédaction

Maquette : AJE Conseils

Publicité : Régie publicitaire, K-Now sarl

Pour la publicité uniquement : Tél. : 01 41 77 16 03
coordination@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
Ce numéro comporte 1 CD Rom

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 : 30 € (Monde Entier) souscription en ligne.

PROCHAIN NUMERO N°110 - Juillet/août - Parution : 30 juin 2008

Dossier Spécial été

Développez fun :

- Programmer votre Lego Mindstorm,
- Popfly Game Creator : votre prochain studio de création !

Et aussi : XML et bureautique
OOXML - ODF : leurs structures,
le modèle de développement

Lectures à la plage :

la sélection de Programmez !

Dossier Tests 2e partie

- Ruby et les Tests
- Mise en œuvre de PHPUnit

Brèves

O'Reilly France ferme !

Mi-mai, la nouvelle est tombée. L'éditeur de livres O'Reilly a fermé sa filiale française, suite à une baisse des ventes. Pour le moment, aucune nouvelle publication en français n'est prévue par l'éditeur. Il faudra se contenter des livres en anglais. C'est bien dommage, vu la qualité des ouvrages O'Reilly.

Microsoft compatible ODF

MS Office 2007 intégrant OpenDocument Format, le concurrent d'OpenXML, un rêve ? Non, la réalité ! Ce support sera effectif dans le Service Pack 2 d'Office prévu pour les premiers mois de 2009, en version 1.1. Il inclura aussi le support natif de XPS (le concurrent de PDF) et le format PDF 1.5, PDF/A (archivage). On pourra ouvrir et sauvegarder des documents ODF. Microsoft continuera par ailleurs à travailler au convertisseur OOXML - ODF. D'autre part, OpenXML normalisé sera supporté par Office 14, la prochaine version majeure de la suite. Et enfin, Microsoft rejoindra l'OASIS (comité technique) sur ODF et son évolution.

Concours Nvidia Challenge

CUDA Challenge est fait pour vous ! Exploitez au mieux la puissance des cartes graphiques Nvidia pour créer des applications étonnantes, novatrices. Les bibliothèques et outils de développement sont disponibles sur le site du constructeur. Vous avez jusqu'au 18 juillet, 20h, pour déposer votre projet. À vos claviers ! www.nvidia.fr/cuda

IDE

Borland vend CodeGear à Embarcadero !

Une nouvelle page de l'histoire informatique s'est tournée le 7 mai dernier. Après avoir filialisé les environnements de développement, faute d'avoir trouvé un repreneur, Borland quitte définitivement le monde des IDE en vendant CodeGear à l'éditeur Embarcadero. Et ce, pour une somme qui paraît bien faible : aux alentours de 23 millions de dollars. Reste maintenant à connaître la nouvelle stratégie que prendra Embarcadero. La marque CodeGear disparaîtra-t-elle à moyen terme ? Ce rachat propulse l'éditeur parmi les principaux fournisseurs d'outils de développements indépendants. Si les synergies entre les deux sociétés sont réelles, la nouvelle entité devrait aussi se poser des questions sur le maintien de tel ou tel IDE. Car en multipliant les IDE, CodeGear a aussi éparpillé une partie de ses ressources de développement. Même si à l'heure de l'annonce, Embarcadero n'a indiqué aucun plan d'arrêt pour tel ou tel produit. Du côté du distributeur français, OpenWay, on se montrait confiant sur la poursuite de la collaboration. A suivre.

WEB

Adobe dévoile Flash 10



Depuis quelques semaines, Adobe a lancé la pré-version de Flash 10 et en particulier son lecteur. Connue depuis plus d'un an sous le nom de code Astro, cette version doit apporter une sérieuse cure de nouveautés sur les performances, la vidéo, la collaboration développeur - designer. Parmi les grandes nouveautés, l'éditeur met en avant les fonctionnalités suivantes :

- **effets 3D** : il sera désormais possible de créer simplement des animations et espaces 3D avec une interactivité totale. Les API seront disponibles pour les développeurs.

- **Personnalisation des filtres et effets** : Flash 10 incorpore Pixel Bender, une technologie servant à After Effects pour les filtres et effets. Flash 10 permettra donc de mieux les exploiter.

- **Texte** : la partie texte a été elle aussi améliorée avec un nouveau moteur de texte (TextFiled demeure).

A cela s'ajoute une API de dessin en partie redessinée pour la rendre plus puissante et flexible. Sans oublier une amélioration des performances pour les applications et la vidéo.

D'autre part, l'éditeur prépare activement le second semestre. AIR 1.1, prévu cet été, est en cours de finalisation (avec la très attendue version Linux).

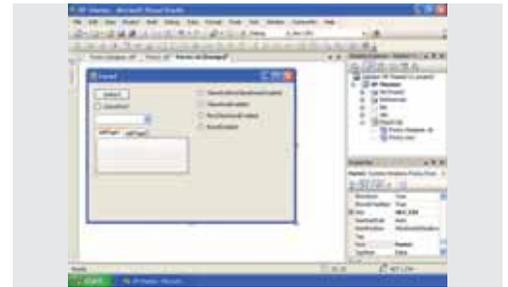
La nouvelle version d'Acrobat sera dévoilée courant juin et les premières informations sur la Creative Suite 4 commencent d'ores et déjà à circuler.

L'été sera chaud !

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

MISE A JOUR

Visual Studio 2008 et .Net 3.5 : service pack 1 en préparation



Début mai, Microsoft a mis en ligne une pré-version du SP1 pour VS 2008 et .Net 3.5.

Il doit apporter des améliorations notables pour la programmation MS Office, Windows et web, avec une optimisation du designer WPF, de nouveaux composants VB et C++. On aura même droit à un Ribbon MS Office 2008 pour MFC ! Parmi les nouveautés annoncées (pas toujours disponibles avec la bêta actuelle) : Intellisense sur le JavaScript, support de SQL Server 2008, une vélocité accrue pour Entity Framework. Concernant .Net 3.5 SP1, le développeur aura droit à une optimisation des applications WPF, sans changement de code, pareillement pour WCF. On bénéficiera du support de SQL Server 2008, d'une installation améliorée pour les applications clientes, ainsi que d'un Client Profile. Cette mise à jour apporte aussi des modifications, ajouts, améliorations sur les différentes bibliothèques : WPF, WCF et workflow. Ce SP1 rajoute donc de nouveaux éléments à .Net. Attention : ne pas installer sur un environnement de production.

Site : [http://msdn.microsoft.com/fr-fr/vstudio/products/cc533447\(en-us\).aspx](http://msdn.microsoft.com/fr-fr/vstudio/products/cc533447(en-us).aspx)

Brèves

HP rachète EDS

Hewlett-Packard a racheté EDS, géant des services. Ce rachat, pour 9 milliards d'euros, met HP en deuxième place du secteur des services informatiques, derrière IBM. Les services s'annoncent comme la prochaine bataille du marché informatique.

ILOG renforce son .NET

Le Français Ilog propose de nouveaux outils pour les développeurs .Net : Diagrammer pour .Net 1.5 et Gantt for .Net 3.5. Ils supportent WPF pour l'interface, ASP.Net Ajax pour les applications web. Disponible immédiatement.

MyEclipse 6.5 : pour juin

Genuitec annonce la disponibilité pour juin de la v6.5 de MyEclipse. Cette version apporte le support de Spring 2.5, l'amélioration du développement JSF, le support des portlets Java. À cela s'ajoutent des améliorations diverses sur les web services avec Jax-WS ou encore Facelet.

VMware : reprise après sinistre

L'éditeur VMware propose une solution de reprise d'activité après sinistre : Site Recovery Manager. Elle permet d'établir des plans de reprise, de les gérer, de les tester. Compatible avec les autres outils de l'éditeur : Infrastructure et Virtual Center.

Web Performance Suite en Français

Le distributeur français Kapitec a annoncé mi-mai la sortie de la version française de Web Performance Suite, environnement de tests de charge pour les applications web. C'est la première fois que cet outil est traduit dans notre langue.

Les applications sur MySpace

Après avoir dévoilé une première version des API pour développer les applications pour MySpace, on peut maintenant voir les applications dans une galerie publique d'applications. Plus de 1000 applications ont déjà été écrites depuis le lancement du projet. Site : <http://developer.myspace.com/>

CONCOURS

Résultat de la finale française d'ImagineCup



Le 15 mai se déroulait la finale française "conception logicielle" à Paris. Occasion pour Microsoft et les différents partenaires de rappeler l'idée forte de cette édition : le développement durable, l'écologie. Une dizaine de projets, que l'on pouvait voir en action, étaient présents sur les stands. Et le jury mit longtemps à se décider tellement il y en avait d'intéressants ! Les vainqueurs sont :

- 3e place : projet **Green Edge**. Il permet de modéliser le produit et ses composants et de mieux choisir sa composition en réduisant l'impact sur l'environnement.

- 2e place : **Hop Cube**. Il s'agit d'une monnaie virtuelle pour, par exemple, permettre à une autre personne d'utiliser un matériel acheté dont on se sert peu, et éviter l'achat du même appareil.

Et le winner is...Le projet **Smartcooking** ! Comme le précisent les créateurs du projet, un quart des ressources du monde est rejeté sans être consommé. Il s'agit donc d'optimiser la consommation du contenu de son réfrigérateur grâce à la technologie RFID. Une puce placée sur chaque élément possède des informations précises (date de consommation, impact de CO2, poids, etc.). Ces informations sont lues par une "centrale" possédant un écran tactile. La "centrale" tient constamment un état des aliments disponibles et prévient de la date limite de consommation ou de l'absence de tel ou tel élément. Et peut proposer des recettes avec des aliments expirant dans quelques jours. Pratique et original ! À noter un prix spécial du jury pour Green Edge.

IDE

Eclipse 3.4 : fin juin

Après la sortie fin avril de Netbeans 6.5, c'est au tour de la fondation Eclipse de préparer activement la sortie, annoncée pour le 27 juin, de la version 3.4 d'Eclipse. Les nouve-

tés sont nombreuses sur l'interface, les API, les éditeurs. Le module de mise à jour, d'installation a été revu de fond en comble et la découverte des plug-in se veut plus simple. On pourra gérer autre chose que des plug-in (ex. : le lancer, les licences, les fichiers de configuration, etc.). SWT possède une nouvelle API graphique permettant de déformer des images. Sous MacOS X, SWT améliore sa compatibilité avec Cocoa et sous Vista, on disposera d'une barre de progression native... On bénéficiera aussi d'un support des machines multi-cœur grâce à un compilateur Java adapté pour utiliser ce type de processeur. Cette version introduit une nouvelle API outil qui devient un standard d'Eclipse. On y trouve : un nouveau tag javadoc, le reporting de problèmes durant les build.

Agenda

JUIN

Le 3 juin, Paris 17e, Espace Champerret, La dixième édition parisienne du **Salon Emploi des Informaticiens Ingénieurs IT**, mise en contact de candidats avec les recruteurs dans l'informatique. www.kavesta.fr

Paris JUG animera une **soirée Spring** le 10 juin. Avec une 1ère présentation à 19h sur les tests unitaires avec Spring puis une 2e à 20h30 sur les dernières nouveautés Spring par Julien Dubois <http://www.parisjug.org/nextMeetings.html>

Le 17 juin 2008, Paris 16e Espace Hamelin, 17, rue Hamelin - Journée exceptionnelle Praxeme (payante) "**Extreme Architecture**" http://www.praxeme.org:80/Documents/XD2008/XD2008_invitation.html

Le 17 juin 2008, CNIT Paris La Défense, **Vocal Expo** - 3e édition de l'événement des technologies vocales avancées www.vocalexpo.com

Le 18 juin 2008, Paris 7e, chez Microsoft France, 148, rue de l'Université, salle Everest (niveau -1), **ALM par la pratique**, événement organisé par Microsoft, STR et Evaluant. <http://www.evaluant.com/alm/default.aspx>

Le 19 juin, Novotel Paris la Défense, 2 boulevard de Neuilly, Valtech organise les **Ateliers client riches Web GWT, Silverlight, Flex** (Prix 230 €) http://www.valtech.fr:80/fr/index/news/seminar/training/Atelier_RIA.html

JUILLET

Les 2 & 3 juillet 2008 à Paris, OCTO Technology organise la 1re édition de **l'Université du Système d'Information**. www.universite-du-si.com

Du 1er au 5 juillet 2008 à Mont de Marsan. 9e **Rencontres Mondiales du Logiciel Libre** www.rmll.info

Brèves

Infragistics met à jour NetAdvantage for JSF

NetAdvantage for JSF 2008 volume 1 est disponible. Cette nouvelle version supporte les portails WebSphere, JBoss et WebLogic. Cette édition met à jour de nombreux composants graphiques Ajax comme les DataTables ou encore les fenêtres de dialogues. www.infragistics.com

Coverity se lance dans Java

Editeur spécialisé dans les tests d'applications embarquées, Coverity propose désormais le support du langage Java avec Thread Analyzer for Java. Il s'agit d'un outil d'analyse dynamique des applications multithread afin de détecter les erreurs de programmation concurrente. Il peut s'associer à l'outil d'analyse statique : Coverity Prevent.

Afresco Enterprise arrive en 2.2

L'éditeur Afresco vient de lancer la v2.2 d'Afresco Enterprise avec un virage " web 2 ". Il prend en compte les améliorations suivantes : fonctions plus performantes sur les tests et la prévisualisation, création plus rapide de nouveaux sites sur un template existant, réutilisation des assets entre projets. Disponible aussi en édition communautaire.

Le W3C préconise WCAG 2.0

L'accessibilité des sites web, et notamment le contenu, pose un réel problème. Pour ce faire,

le W3C travaille sur le Web Content Accessibility Guidelines, dont la v2 est en version finale. Le consortium préconise d'ailleurs aux développeurs de le mettre en œuvre ! WCAG 2.0 fournit un socle stable pour l'accessibilité des applications et contenus Web, et les documents qui l'accompagnent facilitent l'utilisation d'une large palette de technologies Web dans différents environnements et situations. Il se veut plus simple que la v1.

Programmation par DDD

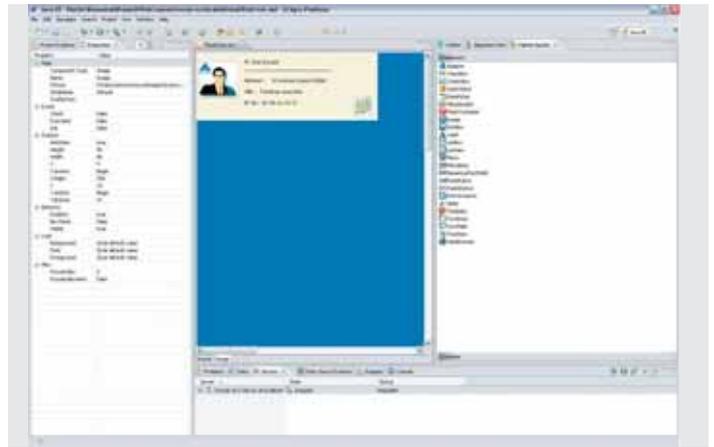
Le développement par modèle de domaine ou DDD est possible dans Visual Studio avec ECO for Visual Studio de CapableObjects AB. Il génère l'application à partir du modèle de domaine et génère la base de données, le code applicatif, le mapping de données. Il repose sur des processus de méthodes agiles comme le SCRUM. Compatible VS 2005 et 2008. Prix : 695 euros. Site : www.capableobjects.com

Iona prend en compte Biztalk 2006

L'éditeur supporte, comme précédemment annoncé, Windows Communication Foundation de Microsoft, avec l'outil Artix Connect for WCF qui s'ajoute dans l'environnement d'infrastructure SOA : Artix. Mais surtout, l'éditeur annonce une interopérabilité avec Biztalk 2006 et les applications JMS et Corba. Les développeurs .Net pourront se connecter à ces applications en transparence.

EDITEUR

Dessinez vos applications Java avec NatJet



L'éditeur Nat System lance son nouvel environnement Java : NatJet. Il permet de développer graphiquement ses applications Java EE et Ajax. Le code est généré automatiquement, on ne mélange plus la présentation et le code métier. On contrôle tout de suite l'ensemble des écrans, l'ordre, l'enchaînement. Pour les données, on dispose de DataObject, un éditeur permettant de faire le lien entre les champs de saisie, les beans métier et la base de données. En complément, NatOrb évite d'écrire du code SQL. L'objectif de l'outil est de réduire les coûts techniques du développement en automatisant un maximum la construction du projet, en évitant d'écrire du code. S'appuyant sur des langages et outils du marché, son intégration est facilitée. Il s'intègre facilement à Eclipse, NatJet prenant la forme d'un plug-in. D'autre part, comme Ajax est présent sur tous les navigateurs, les applications fonctionneront partout. Notons que NatJet s'occupe de générer le code nécessaire Ajax (en réalité du Java). Basiquement, on compose par glisser-déposer les objets d'interface de son écran dans l'éditeur graphique, en configurant le comportement des objets via les propriétés. Pour les données, on passe par le DataObject et son éditeur. Le tout (les écrans, l'accès aux données) est stocké sous forme XML. C'est ce fichier que le générateur utilise pour créer le code Java et les classes. NatJet génère deux classes : une classe abstraite technique (description des écrans servant aux développeurs), classe métier (code métier). Le développeur interviendra prioritairement sur la classe métier. La sortie est prévue pour le 16 juin. Le prix n'a pas encore été fixé. Site : www.natsystem.fr

NUAGE

Mesh : l'autre face du "nuage" de Microsoft

Microsoft a dévoilé une nouvelle brique de ses services en ligne : Mesh. Il fait partie de la plate-forme Live. Pour faire simple, Mesh permet de travailler, de retrouver, de synchroniser ses dossiers, fichiers, documents sur l'ensemble de ses terminaux à partir d'un seul point d'accès : Mesh. Basiquement, il permet : de protéger ses données, de rester informé, de faire un partage, d'accéder de n'importe où à son compte. Microsoft annonce sa disponibilité prochaine sur Mac et les téléphones portables. Pour le développeur, on dispose d'un SDK spécifique : Live Mesh SDK pour utiliser le service de synchronisation dans ses applications, créer un flux de partage, étendre l'accès à son Live Desktop et enfin pour implémenter les flux d'actualité de Mesh (via des notifications) dans son application !

Framework
.Net

Ajax

JSF

GWT

 **valtech**
training

Eclipse
RCP

Flex

AIR

1 journée
3 ateliers
230 €HT

90 formations
au développement logiciel

chez vous
ou à Paris, Toulouse, Lyon, Grenoble,
Genève, Bruxelles, Luxembourg

Atelier clients riches Web : GWT, Silverlight et Flex le 19 juin à Paris la Défense

Après une introduction rapide aux problématiques et solutions du client riche, vous participerez à des ateliers techniques successifs de 2 heures sur Flex, GWT puis Silverlight.

Vous écrirez votre première application dans chacune de ces technologies et serez ainsi amené à juger de leur facilité de mise en œuvre :

essayez-les, choisissez !

9h00 - 10h30 Théorie et démonstrations
par Xavier Paradon - Valtech Training (*)

Introduction

- Limites des applications Web
- Client riche : définition
 - Richesse graphique
 - Impact sur l'architecture
 - Déploiement et portabilité
- Rich Desktop Application (RDA)
et Rich Internet Application (RIA)
- Quelles différences et quel avenir ?

Tour d'horizon des technologies

- Flex, Adobe Integrated Runtime (AIR)
- Silverlight, Volta
- Ajax, Google Web Toolkit (GWT)

Démonstrations

10h45 - 12h45 atelier Flex
animé par Yann Chevalier - Baa0 (*)

14h15 - 16h15 atelier Silverlight
animé par Angelico Pacifico - Agile Partner (*)

16h30 - 18h30 atelier GWT
animé par Sami Jaber – DNG Consulting (*)

(*) Valtech Training se réserve la possibilité de changer d'intervenants.



+33 (0)1 41 88 23 00 - +33 (0)5 62 47 52 00
info@valtech-training.fr

www.valtech-training.fr

Intel Software Conference : l'avenir du multi-core

C'est mi-avril que s'est tenue la conférence sur les produits Intel à Prague. Elle a permis aux revendeurs ainsi qu'à la presse européenne de découvrir les nouveautés et l'avenir du multi-core. Cet article présente les enjeux et le futur des développeurs face aux évolutions à venir des processeurs et des outils associés.

La première présentation a permis d'éclaircir ce qu'est le multi-core : ses enjeux, son environnement ainsi que la stratégie mise en œuvre pour favoriser l'innovation. Intel souhaite renforcer la documentation autour du multithreading, notamment par la publication de livres blancs et de documentations techniques. Elle souhaite également créer une grande communauté autour de ses processeurs, de ses produits et plus globalement du multi-core. Cela passera par des blogs, compétitions de jeux, rencontres sur Second Life, le site web <http://whatif.intel.com>, l'open source ou encore des partenariats avec les universités. Un vaste programme est prévu pour supporter les développeurs. Intel met d'ores et déjà à leur disposition un " developer center " ainsi que des entraînements en ligne et des évaluations gratuites de ses outils. Intel a confirmé sa stratégie en termes de processeurs : c'est la fin de la course à la fréquence et le focus est maintenant ciblé sur les architectures multi-core. L'ensemble de la gamme de produits s'oriente donc vers ces nouvelles architectures et le logiciel Threading Building Blocks fait figure de proue. Celui-ci supporte de nombreux OS, dont Windows, Linux ou encore AIX. La liste des processeurs ne se limite pas non plus aux x86 puisque les processeurs Sun, IBM ou encore Sony-Cell sont pris en charge. L'outil a



également été traduit en japonais et d'autres processeurs seront supportés à l'avenir. D'ici 2009, de nouveaux produits seront mis sur le marché, notamment pour répondre aux besoins des solutions mobiles qui subissent actuellement une véritable explosion avec des produits tels que le iPhone ou l'Asus EEE. Enfin, il s'avère que les outils ciblent essentiellement les langages C et C++. Les langages .NET et Java n'ayant pas un grand besoin d'API, de par la nature de leurs utilisations et architectures. En effet, l'architecture n tiers utilisée par les applications d'entreprise permet de naturellement profiter des processeurs multi-core. La raison à cela ? Chaque requête client provoque l'exécution d'un thread sur le serveur. Ainsi, pour un serveur correctement dimensionné et moyennement chargé, l'utilisation de l'ensemble des cœurs sera optimale dans la plupart des cas. Cela dit, il est de toute façon difficile de fournir des API exploitant pleinement le concept de thread

pour du code managé. En effet, le code s'exécute au sein d'une machine virtuelle qui est elle-même exécutée par le système d'exploitation. Le champ d'action est donc relativement limité.

Intel : processeurs à venir et outils

Larrabee est le nom de code d'un nouveau processeur graphique qu'Intel développe et qui est annoncé comme le remplaçant de la gamme actuelle d'accélérateurs graphiques. L'architecture est révolutionnaire et s'éloigne fortement des processeurs graphiques que l'on connaît actuellement. La commercialisation est prévue pour 2009 et quelques prototypes devraient être présentés d'ici la fin de l'année. *Larrabee* est révolutionnaire dans le sens où il intègre un processeur x86 de 12 à 16 cœurs qui peuvent donc être programmés directement par une API x86 ouverte, avec retro-compatibilité DirectX et OpenGL. La carte ne se limite pas aux rendus 3D et pourrait être utilisée par les



moteurs physiques et les moteurs d'intelligence artificielle. Une utilisation dans le secteur du HPC n'est pas exclue et serait même fortement probable. Nehalem est un processeur plus conventionnel qui viendra remplacer les actuels Core 2 Duo. Il devrait également être commercialisé en 2009 et pourrait être équipé d'entre 6 et 8 cœurs. Il y a donc une réelle évolution vers le many core que les développeurs vont devoir appréhender. Afin de les aider à prendre en charge ces multiples cœurs, une gamme de produits est proposée à la communauté. On retrouve bien sûr le compilateur optimisé, mais l'outil qui permet de tirer toute la puissance des processeurs est sans aucun doute *Threading Building Block*. Cette API permet de s'affranchir des concepts de Threads au niveau du code pour apporter un niveau d'abstraction désormais nécessaire. Cela permet d'être plus productif et surtout de mieux gérer la répartition du travail. L'API permet d'optimiser le code grâce au concept de pipelines. La librairie IPP (Integrated Performance Primitives) est quant à elle plus spécifique et offre un ensemble de fonctions prêtes à l'usage. Ces fonctions sont orientées calculs scientifiques et traitement du signal. Les applications multimédias peuvent en profiter.

Gagner en performance avec le parallélisme

Dans sa session, Phil De La Zorda, a révélé quelques cas pratiques d'utilisation des produits Intel, avec des interviews de clients. Outils professionnels et vidéos ludiques sont impactés positive-

ment par les outils. Le cas du studio de développement de jeux vidéo Crytek est également analysé. Ce qui ressort de ces "cas clients" est un gain en productivité. Ce gain permet aux utilisateurs des produits Intel de réaliser des économies budgétaires et d'améliorer le "time to market", c'est-à-dire le temps entre l'initiation des développements d'un produit et sa commercialisation. Cet aspect est très critique dans les secteurs concurrentiels et notamment pour les éditeurs de logiciels.

Implication des développeurs

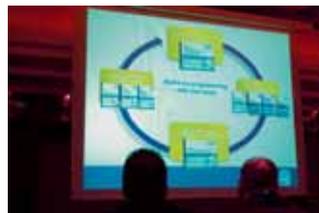
Pour faire suite aux atouts fonctionnels de l'approche multi-core, les intervenants de cette présentation ont abordé les problématiques que les développeurs rencontrent sur le terrain. La première difficulté est un manque de compétences et une nouvelle approche de développement au niveau de la méthodologie. Les difficultés purement techniques sont très variées et s'apparentent aussi bien à des problématiques de débogage que de synchronisation. Les deux consultants ont apporté également leurs retours d'expériences sur les applications types qui exploitent le parallélisme. On y retrouve des thèmes très variés de l'informatique qui regroupent quasiment toutes les branches: applications scientifiques, jeux vidéos, la virtualisation, la business intelligence ou encore les interfaces graphiques, dont Windows 7 sera probablement la démonstration la plus aboutie de Microsoft. En attendant, le monde Open Source apporte déjà quelques solutions avantageuses pour le grand public, notamment avec Ubuntu et son intégration de bureaux virtuels avec Compiz.

Le secteur du HPC et le multi-core

Le terme HPC (High Performance Computing) décrit l'usage de super

ordinateurs et de clusters qui forment un seul système logique utilisé par une ou plusieurs applications. Il s'agit donc d'environnements d'exécution massivement parallèles ayant recours à de nombreux processeurs interconnectés. De par leur flexibilité, leur consommation d'énergie et leur faible coût, les systèmes HPC dominent le monde des super ordinateurs.

L'utilisation d'outils tels que ceux proposés par Intel et notamment *Threading Building Block* permet des gains de 5 à 10% sur les architectures parallèles. Intel est donc considéré comme le leader technologique sur le secteur des processeurs multi-core. Cray le confirme en signant un accord avec Intel dont l'objectif est de faire progresser le calcul intensif sur processeurs Intel. La solution choisie sera dans un premier temps le Xeon, cependant les deux entreprises vont collaborer au cours des prochaines années afin de faire évoluer les technologies de calcul intensif.



Conclusion

Intel va pousser la technologie multi-core sur toutes les plateformes et la communauté autour de la technologie multi-core va se former rapidement. AMD peine à être innovant, en revanche VIA et les autres fondateurs, dont Samsung, peuvent tirer leur épingle du jeu en proposant des solutions novatrices, notamment en terme de faible consommation. De leur côté, les développeurs devront être en mesure de travailler avec des outils qui permettent l'abstraction de l'utilisation des threads que ce soit avec les outils Intel ou non.

■ **Loïc Guillois**

Web > PHP

Formation PHP Expert



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

Au Programme des 5 jours :

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



Prochaines sessions

Paris 2008

23 Juin, 7 et 21 Juillet
8 Septembre, 6 et 17 Octobre
24 Novembre, 1 et 15 Décembre

Lyon 2008

23 Juin, 7 Juillet
8 Septembre, 6 Octobre
24 Novembre, 15 Décembre

Tarif

1995 € HT

LE SPECIALISTE DE LA FORMATION POUR L'OPEN SOURCE



anaska
Conseil & formation en logiciels libres

Informations
01 45 28 09 82
www.anaska.com

Anaska adhère à la Fédération nationale des Industries du Logiciel Libre - FNILL

JavaOne 2008 JavaFX en vedette

Pour la 14e année, Sun tient sa grand' messe JavaOne, entièrement dédiée à Java. Les développeurs avaient de grandes attentes autour de l'événement. Et c'est toujours une occasion pour les éditeurs de dévoiler leurs nouveautés. Cette édition 2008 a-t-elle tenu ses promesses ?



Le slogan de cette année était Java+You, Java et vous, sous-entendu les utilisateurs et les développeurs. Cette tendance se retrouve dans la nouvelle mise à jour de Java 6 qui doit recentrer l'utilisateur au cœur de Java. L'autre tendance concerne le développement web et mobile. La concurrence étant vive, Sun et le monde Java se devaient de réagir, tout en minimisant Android et les plateformes Flex, AIR et Silverlight.

OpenSolaris

OpenSolaris (projet Indiana), piloté de façon intensive par Ian Murdock (fondateur de Debian) depuis son arrivée chez Sun, il y a environ un an, est désormais disponible sous forme de binaires notamment en Live CD, comme n'importe quelle distribution Linux. Ce Live CD vient avec le support ZFS (selon moi, le seul vrai aspect intéressant), le nouveau système de packaging baptisé IPS (Image Packaging System) et DTrace, l'outil de profilage et d'analyse dynamique de performance développé par Sun.

Update 10

Sun a bien entendu parlé de la 'Preview Release' de l'attendue Update 10 du JDK. Elle marque un tournant majeur par la modularité

du JRE (le consumer JRE) constitué d'un kernel léger qui téléchargera les modules complémentaires au fil des besoins, permettant un démarrage plus rapide, notamment des applets. Car c'est le grand retour des applets (si, si !). Sun a soigné le plug-in Java afin de permettre aux applets une exécution en tâche de fond et surtout, combiné à JavaFX, de pouvoir faire un drag and drop de l'application sur le bureau, puis la remettre dans le navigateur ou encore la conserver hors ligne sur son poste. La démo lors de la session a malheureusement crashé plusieurs fois (so-disant à cause de la taille des tuyaux au Moscone Center), mais cette fonctionnalité était vraiment intéressante et apportait quelque chose de nouveau par rapport à Flex et LiveCycle, que JavaFX (disponible dans l'update 10) cherche à concurrencer.

JavaFX

Sun a massivement misé sur cette plateforme, même si l'annonce date en réalité de JavaOne 2007. De nombreuses sessions étaient disponibles cette année. Neil Young (voudraient-ils copier Apple ?) fut invité sur scène pour montrer comment grâce au support BlueRay et JavaFX de la PlayStation 3 de Sony, il était

capable de réaliser une application JavaFX permettant de lire tous ses disques, ses vidéos en HD et même de recevoir les mises à jour des archives depuis internet. L'accent est mis d'ailleurs sur la phénoménale quantité de matériel qu'il est possible de mettre sur un disque BlueRay, Neil Young déroulant une liste de tous ces titres, longue d'une bonne dizaine de mètres ! L'application est bluffante par son ergonomie et son interactivité et constitue un bel exemple de ce qu'il est possible de faire. Sun nous dévoile son planning pour JavaFX :

- Juillet 2008 : JavaFX Desktop SDK Early Access Program
- Automne 2008 : JavaFX Desktop 1.0
- Printemps 2009 : JavaFX mobile et TV 1.0

Java et la mobilité

L'accent le plus fort a sans conteste été mis sur la disponibilité de Java sur les matériels portables. Java a cherché à montrer son adaptabilité sur les terminaux et sa pénétration sur ce marché. Sun a martelé plusieurs discours (Schwartz, Green et Gosling s'y sont mis !) sur le fait que l'information doit être partout, toujours à portée et que Java se doit de permettre de la diffuser. Nombre

de démos se sont donc enchaînées sur le sujet, à commencer par la démonstration de LiveConnect (celle-là même qui a servi pour la démonstration du plug-in Java) directement sur le JRE d'un GSM puis via Android. Cette dernière démo cherchait certainement à rassurer les développeurs Java concernant Android et sa compatibilité avec les standards de Sun, Android utilisant sa propre version de Java (Apache Harmony). Mais ces démonstrations étaient des "amuse-gueules" par rapport à celles présentées lors de la session de James Gosling :

Tout d'abord, **Sentilla** a démontré une petite carte électronique de la taille d'un timbre poste qui permet de connecter des capteurs et de réaliser des programmes Java embarqués. Cette puce permet de rendre intelligent des objets



simples (pervasive computing). Une première démo montre comment, à l'aide de capteurs infrarouges placés aux portes de la salle de session, il a été possible de suivre les personnes entrant et sortant au fil du temps et par session. Une autre démo montre comment rendre intelligentes de grosses balles qui sont lancées dans la salle. Chacune de ces balles embarque des capteurs et leurs mouvements sont traqués par d'autres puces Sentilla réparties dans la salle pour être envoyées vers une application Swing permettant de représenter leur position dans la salle.

Javacard 3.0 a également donné lieu à une démonstration à travers un concours de programmation de robots : chaque participant devait réaliser la stratégie de combat de son robot et la charger sur une carte à puce. Celle-ci était ensuite lue par le programme central qui consistait en une application Swing affichant les robots en train de combattre. Cette démonstration, amusante, permet toutefois de comprendre la puissance de ces cartes embarquant Java et pouvant contenir des programmes jusqu'à 1 Go ! Ceci est d'autant plus intéressant quand on sait que JavaCard 3.0 fournit un conteneur de servlets, le protocole HTTP et la pile TCP/IP...

Enfin, dernière démonstration, de loin la plus impressionnante, celle de **Pulse**, un stylo intelligent réalisé par la société LiveScribe. Il s'agit d'un stylo qui permet d'enregistrer et de synchroniser à la fois l'écriture et la voix. Le concept est difficile à expliquer tant c'est inimaginable, mais chaque mouvement du stylo et ce qui est écrit est enregistré dans le stylo via une mini caméra en même temps que le son. Une reconnaissance d'écriture est effectuée. Lors de la relecture des notes, il suffit de cliquer sur son cahier (vraiment !) et le stylo rejoue le son enregistré à ce



moment là. Le contenu peut être envoyé sur un PC pour traitement par une application desktop qui permet de faire de la recherche plein texte. De plus, le contenu peut être partagé sur internet via une application Flex. On peut voir la démo sur leur site (<http://www.livescribe.com>).

On a aussi vu le projet WebSynergy, résultant de la collaboration Sun et Liferay et visant à développer de façon conjointe une plateforme de présentation web incorporant à la fois le portail Liferay, GlassFish, OpenPortal, OpenSSO et d'autres projets. De même, Apple sort (presque) en silence la version 6 de Java pour OSX et rien sur le support de Java sur iPhone. JavaOne n'a également pour ainsi dire reçu aucun écho de l'annonce par SpringSour-

ce de la mise à disposition de Spring Application Platform, un serveur permettant de faire tourner à la fois des applications J2EE et des applications Spring et se basant sur Spring DM kernel (anciennement Spring OSGi). Enfin, JBoss a organisé un événement SpeakEasy (soirée privée) pour annoncer la sortie de JON (JBoss Operation Network) 2.0, une solution centrale de gestion d'un parc applicatif réparti sur n machines, basé sur Hyperic.

Coté pavillon

Du côté des exposants, on pouvait noter que le stand Nokia s'est réduit à sa plus simple expression, tout comme celui d'Adobe (qui s'est fait tout petit pour laisser de la place à JavaFX ?), le haut du pavé étant tenu par Moto-

rola, Intel, AMD et Oracle. Faut-il en déduire quelque chose sur le fait que les panneaux Oracle portaient également le logo de BEA ? Au niveau des démonstrations, Liferay présentait sa nouvelle release de Liferay (5.0), Spring-Source sa Spring Application Platform, et des démos GlassFish partout de la part de Sun. En dehors des sentiers battus j'ai pu voir une démonstration intéressante de EAF (Enterprise Application Fabric) de la société Appistry qui propose de rendre 'gridable' une application Spring sans rien toucher au code. Côté serveur, EAF joue le rôle d'un container Spring distribué permettant de jouer l'application sur plusieurs serveurs et côté client, l'application est re-'wiré' pour utiliser l'EAF. La démonstration montrait les temps de réponse mais aussi la capacité de tolérance aux pannes, puisqu'un des serveurs est brutalement coupé lors d'une démonstration de mise à jour massive de la base. Autre plateforme d'avenir, TerraCotta, qui n'en finit pas de m'étonner. Pour les personnes ne connaissant pas encore TerraCotta, il s'agit d'une solution Open Source de clustering mais au niveau VM. La solution est complètement trans-

Événements

parente pour les applications et permet d'avoir une granularité très fine sur ce qui est répliqué. Les utilisations sont multiples et enthousiasmantes, peut-être plus que JavaSpaces.

Les sessions techniques

Si de nombreuses sessions furent décevantes, quelques-unes méritaient une attention particulière. Côté Java 7, une chose est claire : c'est flou ! Seules certaines fonctionnalités sont à peu près sûres d'arriver dans la nouvelle version du JDK (multi-catch, safe re-throw). De plus, ceux qui avaient déjà une indigestion, à cause des annotations, vont être servis avec l'intégration de la JSR 308 (annotations étendues) dont le but est de réduire le nombre de bugs qui peuvent passer dans du code à cause de vérifications insuffisantes du compilateur. Les annotations aujourd'hui ne sont en effet possibles que sur les déclarations (de classes, de méthodes, d'attributs, de statements). Le but est d'ajouter des annotations à l'utilisation d'un type, à la déclaration d'une variable locale ou encore sur les casts (si si !). L'ajout de checker pluggable permettra de réduire considérablement le nombre de bugs, mais hélas au profit d'ajout de nombreuses annotations. Par ailleurs, mais cela a déjà été couvert lors du précédent dossier Java 7, on retrouvera également l'API Java modules qui permettra de modulariser les applications et résoudre certains problèmes de Classpath, ainsi que la toute nouvelle API NIO2 qui permettra de bénéficier de plus de fonctionnalités natives des systèmes de fichiers comme les permissions, les liens symboliques ou encore les notifications.

JMC

Java Media Component est une API dont le but est de remplacer JMF tout en le simplifiant. Cette API est

réellement des plus simples et se base sur quatre objets :

- JMediaPlayer : un JComponent qui permet de jouer un media et fournit les boutons de contrôle
- JMediaPane : un JComponent qui permet de jouer un media sans les contrôles
- MediaPlayer : un media player de plus bas niveau qui peut rendre des objets graphiques ou passer des données brutes à un autre système de rendu
- Media : une classe permettant de récupérer des informations sur le media en lui-même.

L'API est extrêmement simple à utiliser :

```
JMediaPlayer mp = new JMediaPlayer(new URI("movie.mov"));
mp.play();
```

Mais l'aspect le plus intéressant est qu'une version script de cette API est disponible pour utilisation dans FXScript. Ces classes sont disponibles dans le package `javafx.gui`. Leur implémentation est basée sur les classes de JMC et supportent les animations, la transparence et les effets. L'effet de reflet sur une vidéo à la Apple est du plus bel... effet !

Java EE

Côté Java EE, la grande nouveauté est évidemment l'API servlet 3 qui comblera des manques historiques (méthodes login et logout atteignables au niveau de la servlet, enfin me direz-vous, mais malheureusement encore en cours de discussion) et ajoutera des fonctionnalités utiles comme :

- le support des servlets asynchrones avec la possibilité de suspendre une requête et de la reprendre (utile surtout dans le cas d'utilisation de Comet),
- la facilité de développement avec l'introduction de nouvelles annotations (`@Servlet`, `@PUT`, `@GET`, `@ServletFilter`, `@ServletContextListener`, ...)
- la pluggabilité avec la possibi-



été pour les frameworks d'apporter un fragment de `web.xml` qui sera ajouté au `web.xml` de votre application

Et surtout, bien sûr, la fameuse capacité à déployer des EJB directement au sein des WAR, fonctionnalité d'ores et déjà supportée par GlassFish.

En ce qui concerne JSF 2.0, c'est encore très flou et on n'en connaît encore que les grands axes :

- faciliter le développement de composants (ce qui n'est pas un mal)
- support AJAX
- le Page Description Language (PDL)
- réduire la complexité de configuration
- permettre une meilleure compatibilité entre les fournisseurs de librairies tierces

Bilan

Finalement, cette édition 2008 est plutôt en demi-teinte et si l'excitation était quasi palpable le premier jour (longues files d'attente dans tous les sens pour les

sessions) tout est très vite retombé. Sun nous a resservi JavaFX, mais s'il était intéressant de voir comment la technologie a progressé, le manque de support au niveau des IDE et une syntaxe à la fois nouvelle et illisible (FXScript) n'encouragent pas à l'adoption. Ainsi, malgré le martèlement fait au niveau des sessions, une grande majorité du public reste sceptique. 2008 sera peut-être le grand retour de l'applet (ironique, non ?).

Côté JDK et J2EE, aucune grande nouvelle, mais les nombreuses démos de GlassFish V3 (ultra-rapide, léger, modulaire) laissent penser qu'il pourrait bien s'agir là du prochain serveur d'applications Open Source de prédilection, en remplacement de JBoss. OSGi devrait également être amené à s'imposer, prenant de court la spécification de Java Modules qui ne devrait paraître avec le JDK 7 que courant 2009.

■ Fabrice Dewasmes

Blog : <http://fdewasmes.free.fr>

Formation
Model Driven/UML
+ Générateur
d'applications

990HT €

Vous êtes
salarié(e) d'une SSII
d'un intégrateur
ou d'un VAR?

Obtenez 50%
sur le prix public

Avec BLU AGE™, valorisez enfin vos connaissances en UML ! Rejoignez la révolution du « Model Driven » !

Développeurs, Concepteurs, Chef de Projet, Architectes, Consultants Fonctionnels

N'avez-vous jamais rêvé de :

- o Livrer enfin en temps et en heure des applications spécifiques, 100% conformes aux exigences fonctionnelles des utilisateurs,
- o Transformer automatiquement et instantanément vos spécifications métiers, en applications prêtes à être déployées,
- o Pouvoir passer d'un framework à un autre en un clic de souris, sans remettre en cause vos investissements dans la connaissance métier,
- o Utiliser votre précieux temps pour concevoir et optimiser les applications, et non pas pour les coder et les débbugger,
- o Utiliser un langage et des méthodes de spécification, compréhensibles par tous les acteurs (techniques ou fonctionnels) impliqués dans un projet de développement.

Avec "BLU AGE™ Discovery", votre rêve devient enfin une réalité !

En trois jours et pour seulement 990 euros HT(*) par stagiaire :

- o Découvrez les fondamentaux du Model Driven Development™,
- o Initiez-vous aux techniques de modélisation UML et de génération automatique de code et d'applications,
- o Passez de la théorie à la pratique, en générant votre première application "école" avec notre atelier BLU AGE™,
- o Repartez avec une licence BLU AGE™ utilisable pendant 30 jours, pour un nombre illimité de générations.

(*) Montant déductible de votre prochain achat de licences BLU AGE™ ou de votre cursus de certification BLU AGE™.

Première journée animée par M. Frank BARBIER,
un des pères fondateurs du langage UML et auteur de l'ouvrage "UML 2 et MDE - Ingénierie des modèles
avec études de cas" aux éditions Dunod.

Inscriptions :

Par téléphone : 01 56 05 88 00
Par Mail : formations@bluage.com

Netfactive Technology, éditeur de Blu Age™, est membre de l'OMG et du SOA Consortium

Prochaines sessions :

- 3/4/5 Juin 2008 (Suresnes, 92)
- 18/19/20 Juin 2008 (Suresnes, 92)
- 1/2/3 Juillet 2008 (Suresnes, 92)
- 16/17/18 Juillet 2008 (Suresnes, 92)

BLU AGE™ est un atelier MDA™
de génération de logiciel, qui permet
une transformation automatisée à
100%, de vos diagrammes UML 2.x et
maquettes d'écran associées, en
applications métier Java EE ou .NET.



NetBeans 6.1 : l'ultime IDE ?

Hier réservé au développement Java, l'environnement proposé par Sun tente à travers cette dernière version de s'imposer comme une solution complète et ouverte aux technologies les plus récentes (mouvement déjà visible dans les précédentes versions).

Sorti fin avril, Netbeans 6.1 comble les lacunes constatées sur les précédentes versions et s'enrichit au passage de nouvelles fonctionnalités particulièrement novatrices. Découvrons à travers cet article les changements apportés à l'outil et profitons de l'occasion pour préciser les principales caractéristiques de celui-ci.

Un environnement plus véloce

Sensible aux critiques émises par la communauté des développeurs concernant les lourdeurs ressenties lors de l'utilisation de Netbeans, Sun a souhaité porter un effort particulier à optimiser son environnement de développement. Ainsi, l'éditeur californien se targue d'avoir grandement amélioré le temps de lancement de l'application, avec une réduction de 40% de celui-ci. Au delà de ce chiffre impressionnant, c'est tout l'environnement qui a fait l'objet d'une réécriture du code afin de rendre l'ensemble plus véloce. Avec Netbeans 6.1, Sun a su répondre à ces critiques jusque là réhébitoraires et fournir un logiciel performant utilisable par les développeurs dans leur travail au quotidien.

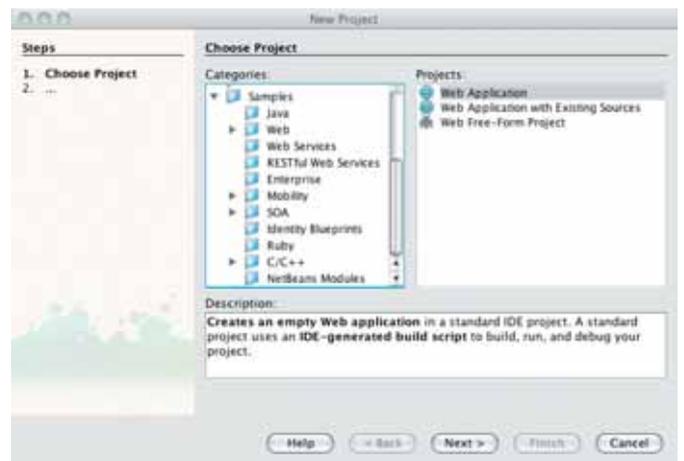
Les grandes nouveautés

Netbeans propose, à travers cette nouvelle version, l'intégration des principales technologies du moment. Ainsi, un effort particulier a été apporté à la gestion du Javascript. Ce langage a su

retrouver toutes ses lettres de noblesse puisqu'il est aujourd'hui de plus en plus utilisé pour des traitements côté client et se retrouve au cœur de la technologie Ajax. Dans un même souci de coller à l'actualité des technologies, l'environnement fait la part belle au langage Ruby. Ce dernier bénéficie aujourd'hui d'un fort engouement de la part des développeurs Web, et Sun ne pouvait dès lors se dispenser d'apporter un support complet à ce langage dans son EDI (Environnement de Développement Intégré).

Editeur Javascript

Avec l'avènement de la technologie Ajax, le langage Javascript est devenu un passage presque obligatoire pour le développement d'un site web moderne. Malgré cet usage intensif, ce langage souffrait d'un manque notable d'environnements adaptés. Nuancions toutefois notre propos puisque ce manque tend à être comblé avec l'arrivée récente d'outils comme Aptana, basé sur l'environnement Eclipse. Avec la version 6.1, Sun a su valoriser le langage Javascript au sein de son environnement. C'est ainsi que Netbeans propose notamment l'auto-complétion du code, une fonctionnalité qui s'avère fort utile au quotidien et qui est disponible depuis longtemps pour d'autres langages. Outre l'auto-complétion, Netbeans offre la coloration syntaxique et fournit un outil de debugging pour Javascript



bien pensé, rendant de fait obsolète l'utilisation d'outils externes jusque là indispensables.

Le plugin center

Autre amélioration, le plugin center a été repensé pour corriger les problèmes de lenteurs constatés auparavant sur le système, lorsque l'utilisateur procédait à la mise à jour de celui-ci. Avec Netbeans 6.1, le téléchargement de plugin s'effectue en tâche de fond afin de garantir une meilleure réactivité du système pendant ce temps là.

Le support MySQL

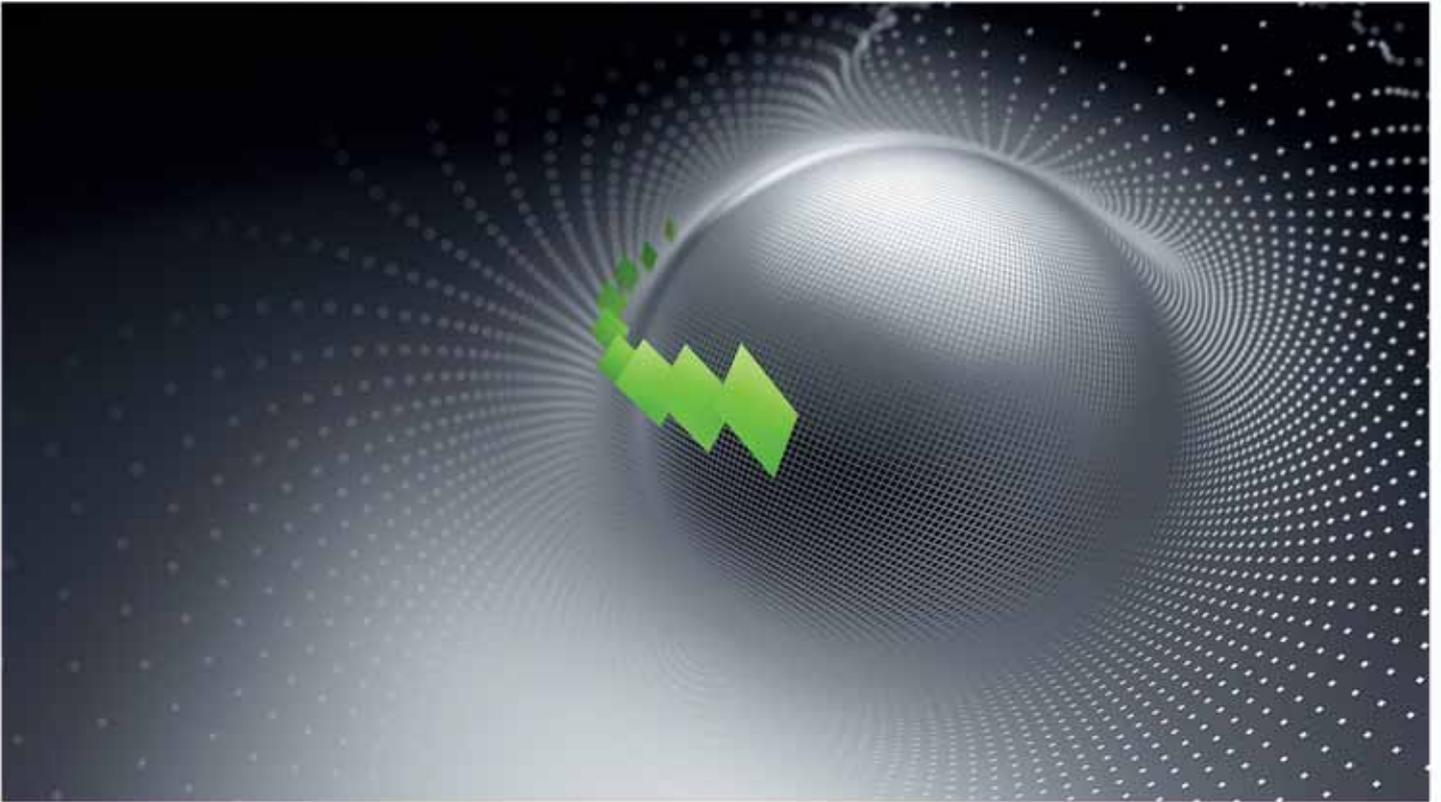
Récemment rachetée par Sun, la base de données MySQL se retrouve aujourd'hui au cœur de la politique de développement des outils de l'éditeur. Outre l'outil MySQL Workbench mis depuis peu à disposition de la communauté gratuitement, Sun a voulu apporter une touche particulière à

l'intégration de MySQL au sein de son IDE. L'explorateur de base de données a été refondu et permet à présent de gérer beaucoup plus facilement les bases MySQL : l'utilisateur a la possibilité de parcourir les différentes tables constituant la base, mais également de créer et supprimer des tables directement dans l'environnement. Il s'agit là d'une fonctionnalité particulièrement intéressante, au regard de l'utilisation toujours plus importante des bases MySQL dans la création actuelle des sites Internet.

L'intégration de Ruby

Ruby constitue aujourd'hui l'un des langages les plus en vogue dans le domaine du développement web. La v6.1 propose un support encore plus poussé de ce langage avec l'intégration notamment du framework Rails dans sa version 2. Au final, Netbeans offre au développeur un environnement

NVIDIA® CUDA™



NVIDIA CUDA est le seul environnement de développement en langage C au monde, permettant l'accès aux GPU NVIDIA. CUDA permet aux développeurs d'utiliser les GPU NVIDIA pour résoudre les challenges de calculs les plus complexes telles que les forages pétroliers et gaziers, la gestion des risques financiers, le design des produits, l'imagerie médicale et la recherche.



Démultipliez la puissance du GPU!

Avec le « CUDA Challenge », NVIDIA encourage la communauté Européenne des développeurs à créer des softwares qui exploiteront au mieux la puissance de traitement des GPU NVIDIA.

De très beaux cadeaux récompenseront les meilleurs projets !



Ne ratez pas l'opportunité de découvrir les dernières informations sur CUDA (communiqués de presse, documentation...)

Pour participer au CUDA Challenge

ou

Pour recevoir les dernières informations sur CUDA

Rendez-vous sur www.nvidia.fr/cuda

Test

bien adapté pour la création d'applications Ruby / RoR avec un confort d'utilisation certain. Précisons en outre qu'il est possible de choisir l'interpréteur Ruby natif ou bien d'opter pour l'interpréteur jRuby développé en Java, ce qui permet d'associer des modules développés dans ce langage pour disposer ainsi d'une architecture plus éprouvée et bénéficier d'une bibliothèque de composants très riches.

Support de PHP

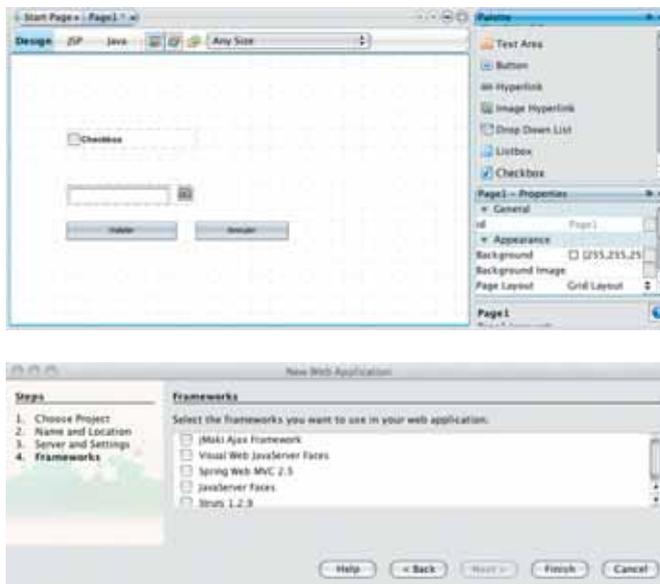
Poursuivant son initiative vers une plus grande ouverture, Netbeans propose un support accru du langage PHP. Le développeur bénéficie à présent de l'auto-complétion du code, de la coloration syntaxique, d'une meilleure navigation au sein des programmes mais également d'options de debugging particulièrement bien soignées.

Choisir ou non Netbeans ?

Comme nous venons de le voir, la nouvelle version de Netbeans apporte un nombre important de fonctionnalités novatrices. La question à présent est de savoir si, malgré toutes ces nouveautés aussi alléchantes soient-elles, l'environnement Netbeans constitue une solution pertinente pour le travail des développeurs dans leurs tâches quotidiennes. Finalement, Netbeans serait-il l'IDE de demain ?

Un outil polyvalent

Force est de constater que Sun a produit pour cette nouvelle version un effort conséquent sur l'intégration de nouveaux langages jusque là relégués au second plan par rapport à Java. Aujourd'hui, Netbeans offre un support complet de Javascript, Ruby, C/C++, et progressivement de PHP. Si l'on considère en outre l'arrivée des nouveaux modules MySQL et les outils de debugging multi langages, Netbeans se présente



comme un environnement tout en un, disposant de toutes les fonctions dont on a besoin le développeur dans le cadre de son travail. Gageons que cette polyvalence saura en séduire plus d'un.

L'intégration des dernières technologies

Grâce au soutien de Sun et de son positionnement technologique, l'environnement Netbeans ne pouvait que bénéficier de cette prédominance et met à disposition des utilisateurs les toutes dernières technologies comme par exemple Phobos, langage de script utilisé pour un développement web rapide et efficace, ou encore jMaki, une bibliothèque de composants graphiques basés sur Ajax. Soulignons ici le très beau travail réalisé par Sun pour offrir ces technologies dans un environnement de développement parfaitement packagé.

La mobilité

Netbeans met également à l'honneur le développement pour mobile en proposant un ensemble de bibliothèques ainsi qu'un module pour créer visuellement l'interface utilisateur. Associé à de nombreux exemples, tout est pensé ici pour fournir au développeur tous les éléments pour créer rapidement

des applications mobiles. Cerise sur le gâteau, l'environnement est livré avec un émulateur Java permettant de tester directement l'application mobile développée.

Le designer graphique efficace

Résolument tourné RAD (Rapid Application Development), Netbeans dispose d'un designer d'interface graphique (GUI Builder) riche en composants. Il s'appuie sur des frameworks éprouvés comme Swing pour les applications lourdes, jMaki ou JSF pour les applications Web.

De nombreux tutoriels

Associé à son environnement de développement, Sun propose à la communauté une importante base d'exemples traitant de sujets variés. Ainsi, il est possible de consulter des exemples relatifs au développement d'applications Ruby, au développement pour mobile, à la création et l'utilisation de web services, etc. Particulièrement appréciables, ces exemples offrent aux développeurs curieux, un moyen simple et efficace pour appréhender les dernières technologies. Précisons que ces tutoriels sont accompagnés de commentaires clairs, facilitant d'autant leur com-

préhension et que des rappels vers le site de Sun permettent d'approfondir certains sujets.

Le manque de support de la part de l'éditeur

Comme la plupart des logiciels gratuits, l'environnement Netbeans ne bénéficie pas du même niveau de support qu'un produit commercial propose habituellement à ses utilisateurs. Il n'existe pas de numéro de hotline ou de services après-vente à contacter en cas de problème. Comme souvent dans ces cas là, il convient à l'utilisateur d'aller lui-même au devant des informations dans les forums ou sur les sites dédiés pour trouver la solution à son problème.

La lourdeur de l'environnement

Riches de toutes les fonctionnalités précédemment évoquées, l'environnement Netbeans s'avère par conséquent très gourmand en ressources matérielles et il est indispensable de disposer d'une machine puissante pour garantir une utilisation de l'outil dans de bonnes conditions au quotidien. De plus, l'activation de fonctions comme l'auto complétion ou la documentation contextuelle peuvent parfois amener des lenteurs significatives et il est alors indispensable d'abandonner celles ci pour garder un environnement suffisamment réactif.

En anglais

Enfin, Netbeans 6.1 ne s'utilise pour le moment que dans la langue de Shakespeare, mais il ne faudra sans doute que peu de temps avant qu'un module de traduction ne soit proposé à la communauté. Après ce rapide tour d'horizon des différentes fonctionnalités proposées par Netbeans 6.1, nous pouvons retrouver l'environnement sur le CD-Rom du mois de Programmez. **Prochaine version** : 6.5, prévue pour début octobre 2008.

■ Vincent Perdereau
vincent.perdereau@ads-com.fr

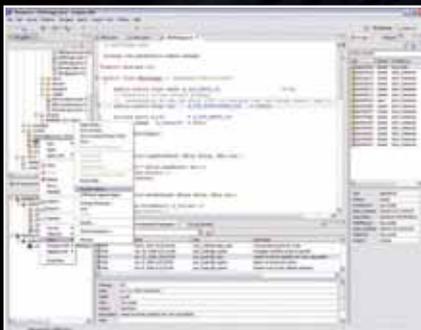
Présentation du module d'extension de Perforce pour Eclipse

Pour travailler avec Perforce dans une interface IDE Eclipse.

Le module d'extension de Perforce pour Eclipse permet aux développeurs d'accéder facilement au système de GCL Perforce depuis leur interface IDE Eclipse. Il propose les fonctionnalités suivantes :

- Accès rapide à l'historique complet des fichiers
- Prise en charge complète du développement collaboratif, avec possibilité de fusionner les fichiers
- Possibilité de travailler hors ligne lorsque la connexion avec le serveur Perforce est indisponible
- Outil de comparaison des fichiers et prise en charge du suivi des défauts intégrés
- Prise en charge de la fonction de refactoring de l'environnement Eclipse

Le module d'extension de Perforce pour Eclipse prend en charge les systèmes d'exploitation Windows et Linux. Et ce n'est que l'un des nombreux outils intégrés dans le système de GCL Perforce.



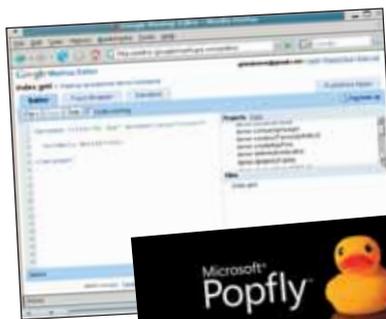
Module d'extension de Perforce pour Eclipse



Mashup

*l'art de composer,
assembler et présenter
les données !*

Le mashup, parfois, on en crée soi-même, très simplement en assemblant des briques entre elles, ou en codant quelques lignes de code, du javascript le plus souvent. Mais si nous devons en donner une définition, par quoi commencerions-nous ?



Grossièrement, le mashup est une agrégation, une composition de différents services entre eux pour afficher, visualiser et récupérer des informations, des données provenant de différentes sources. Ces services peuvent prendre de nombreuses formes : web services, services hébergés, services de type Google Maps, Live (Microsoft), Yahoo, MySpace, Facebook, etc.

On aurait tendance à penser : mashup = application composite. Il est vrai que le mashup est une catégorie, un type d'application composite. Mais ces dernières vont plus loin que le simple mashup. L'un des intérêts de notre assemblage est de pouvoir utiliser des briques hétérogènes pour ne voir que les informations

voulues dans une interface unifiée ou non. La technicité, la complexité du mashup dépendent du contexte et de l'utilisateur final, il existe des solutions pour créer sans code. Dans le cas d'interactions plus complexes, il faudra créer du code javascript, REST, Atom. Pour le mashup métier, là il faudra intégrer des API, des mashup éditeurs, prévoir un déploiement, etc.

Vous l'aurez compris avec la dernière remarque, le mashup n'est pas unique mais multiple. On répertorie trois grandes catégories : utilisateur (mashup consumer), données (data mashup) et métier (business mashup). Pour compliquer encore un peu les choses, les éditeurs ne sont pas d'accord sur les limites

de chaque catégorie et parfois, les uns disent que le mashup métier existe d'ores et déjà, alors que pour d'autres, c'est encore un concept virtuel.

D'autre part, comme on le verra, le mashup se consomme de différentes manières. Que ce soit sur une simple page de son navigateur ou dans un portail d'entreprise.

Dans ce présent dossier, nous allons aborder les aspects généraux des mashup ainsi que deux approches côté outillage : Popfly (Microsoft) et Intel Mash Maker. Nous regarderons aussi l'état du marché, les approches et les stratégies.

Bonne agrégation !

■ François Tonic

Mashup : où, quand, comment, pourquoi ?

Passons tout de suite aux choses sérieuses et pénétrons dans le monde merveilleux du mashup ! Comme expliqué précédemment, il existe plusieurs sortes de mashup qui sont fonction du contexte, de l'utilisateur visé, des données à chercher, à afficher. Même si les fondations restent identiques, le niveau de technicité évoluera très rapidement. Et c'est là que le développeur interviendra, pour créer la glue entre les briques, et que l'administrateur sera aussi de la partie pour assurer le déploiement, la disponibilité et les mises à jour.

On peut découper le mashup en trois couches : Présentation, logique, données. Et souvent, mashup rime avec widget / gadget. Ce qui est vrai, le gadget facilite l'agrégation et est surtout sans développement. Un des avantages du mashup est de s'appuyer sur des composants réutilisables, ce qui permet d'en créer de nouveaux très rapidement. Cependant, pour l'accès à des applications, à des données dont on ne possède pas de connecteur ou de services, il faudra créer l'exposition. Et d'une manière générale, si vous n'êtes pas habitué aux API, services, protocoles REST, Atom, RSS, etc., il faudra les apprendre et les manipuler.

Où ?

Basiquement, il s'utilise dans des sites web, une simple page web, ou le plus souvent dans le cas d'un portail d'entreprise, dans ce portail. Cette solution sera souvent la plus usitée dans un



mashup métier au sein d'une entreprise (portail intra ou extra-net). Le portail a l'avantage d'être un point d'entrée en entreprise. Et aujourd'hui, ces environnements supportent les gadgets et les " technologies web 2 ". L'exposition se fera alors souvent via un serveur d'applications.

Quand et pourquoi ?

Le mashup peut être vu comme un tableau de bord (dashboard) dans lequel on agrège, on

assemble des services, des widgets, des web services, etc. En entreprise, soit l'utilisateur peut assembler lui-même les briques, sans code pour créer des mashup simples, soit le développeur intervient pour créer de véritables applications mashup complexes en utilisant les API des éditeurs, en récupérant des briques tierces, etc. Il peut aussi servir à visualiser des informations provenant de différentes sources : suivi commercial (ex. : un commercial veut consulter les données d'un client, d'un partenaire), suivi de stocks, pour la géolocalisation, créer une place de marché, etc. L'avantage est que le mashup cache les applications, l'aspect hétérogène. Surtout, il est d'une souplesse incomparable car il est facile de mettre à jour les briques, de le modifier, de l'adapter à ses besoins, de supprimer un service, d'en rajouter, etc. Bref, il est flexible et évolue avec l'utilisateur et les services disponibles.

Le mashup évite de développer des applications spécifiques et donc permet de répondre plus rapidement à ses besoins (à la maison, au bureau). D'autre part, au lieu d'avoir plusieurs interfaces dans une seule page web, on affiche l'ensemble des briques et informations, bref l'interface s'unifie et utilise les concepts apparus avec le " web 2 ". Il n'a pas vocation, a priori, à devenir une application legacy. On pourrait presque le comparer à des applications jetables après usage.

Comment ?

Il s'appuie sur les langages, les technologies et les standards du monde du web. En vrac, on citera : Ajax, javascript, html, REST, Atom, RSS, XML, widget, etc. Ensuite, pour les outils, de nombreux éditeurs proposent des environnements complets pour créer ses mashup. Cela va du simple éditeur aux environnements de type RAD, avec construction graphique, fonctions d'import / export, etc. (Microsoft PopFly, Intel Mash Maker), en passant par les environnements de création mashup d'entreprise (Lotus, Serena Business Mashup et Mashup Composer, Twinsoft



Quelques règles à méditer

- **Penser l'interface** : la cohérence de l'interface utilisateur est un réel souci, surtout avec l'utilisation de services et de widgets ayant leur propre interface. Essayer de lui donner une cohérence d'ensemble, surtout dans un mashup métier
- **Définir les services**, composants, widgets dont vous aurez besoin
- **Définir les sources de données**, leurs origines, les liens d'accès
- **S'assurer de la bonne compatibilité** des composants du futur mashup
- **Le mashup doit correspondre à un besoin**, une demande, une attente
- **Définir le mode de déploiement** : surtout pour un mashup métier
- **Tester** les environnements de création avant toute conception
- **Séparer interface, données, code**

Convertigo Enterprise Mashup Server). On trouve des outils gratuits, commerciaux, open source. À cela se rajoutent des API, des briques prêtes à l'emploi que l'on intègre à son environnement pour accéder par exemple à un CRM, un ERP, une base de données, des web services, etc. Pour la consommation du mashup, on rajoutera un serveur comme un serveur portail. Des éditeurs cherchent à créer un véritable écosystème autour de leurs environnements de mashup. Par exemple, Serena incite les éditeurs, partenaires à créer des composants et des services que le développeur récupère pour les intégrer directement dans son mashup en facilitant ainsi l'accès à des données d'un progiciel.

Le mashup de plus en plus pro !

Si aujourd'hui le mashup consumer est le plus visible, en entreprise et pour une utilisation professionnelle, il y fait doucement sa place. Pour Serena, le mashup business est plus qu'un concept, c'est une réalité. L'éditeur s'y investit beaucoup. Pour preuve, fin avril dernier, il dévoilait une place de marché dédiée : Mashup Exchange ! On y trouve de tout : la gestion et la coordination des CRM, la gestion des ressources humaines, le lien avec le Help desk, le mashup pour le département informatique, la gestion de la méthodologie SCRUM ou encore le développement d'applications. Par exemple, on peut mettre le mashup ALT RM SCCM



Dashboard. Il permet à chaque intervenant d'un projet de voir les statuts et métriques, le tout connecté à Dimensions RM. Chaque mashup est payant ou

L'exemple de iGoogle / Google Mashup Editor

Nombreux sont ceux qui composent sur iGoogle leur page web. On crée simplement, par ajout de gadgets son mashup personnel (mashup consumer et mashup de données). Dans la galerie des gadgets, on choisit le gadget à rajouter. Puis on arrive sur la page d'ajout avec l'ensemble des options disponibles du composant. Le nouveau gadget s'ajoute dans sa page iGoogle. Bien entendu, on réorganise la page comme on le souhaite par simple glisser-déposer. On change le thème de l'interface.

Un thème iGoogle est un fichier XML contenant les éléments clés pour changer l'affichage et les attributs. Et chaque thème peut contenir un ou plusieurs skins. On peut ainsi changer de skin selon le jour, l'heure. Et chaque skin contient un fichier



configmap. Premièrement on crée un visual design puis le template et enfin on le soumet au répertoire iGoogle. La page iGoogle contient : un header, zone de contenus et le footer (optionnel).

Google Mashup Editor est un éditeur en ligne pour coder son mashup. On dispose d'une arborescence de projets et de fichiers et d'une zone d'éditeur. On y édite, compile et teste (le test s'effectue dans une sandbox) ses applications. Pour cela il faut disposer d'un compte

actif. On peut colorier son code html, javascript, css. Il dispose d'une vérification syntaxique. La publication se fait d'un clic, en indiquant le nom du projet. L'url est généré et ensuite on publie directement. Bien entendu, on peut rajouter à son projet des feeds (via gm:list et Feed Browser). On peut créer et utiliser des templates, via du CSS. On possède un contrôle total de l'interface. Comme tout mashup, on définit les sources de données et de services nécessaires.

gratuit, avec ou sans support, ou avec une offre SaaS.

Serena dispose de deux environnements : *Mashup Composer* et *Mashup Server*. Mashup Composer est l'outil de création qui se veut simple et ergonomique sans nécessiter une bonne maîtrise du code HTML, javascript, CSS. L'interface s'inspire d'Office 2007 et permet à des non développeurs de créer leur mashup, avec des flux métiers, des processus. La maîtrise d'un tel outil nécessite une formation afin d'acquérir les bons réflexes, surtout l'agrégation et la construction des flux. Serena s'attaque aussi au déploiement avec l'offre Mashup Server. Là encore, la volonté est de simplifier les mécanismes en évitant l'intervention de l'administration. On publie depuis Mashup Composer dans le serveur et le serveur expose au réseau le

mashup. Une édition hébergée (mode SaaS) est même disponible ! Côté Microsoft, par contre, on fait plutôt profil bas. Pour l'éditeur, le mashup métier reste un concept et il met plutôt en avant le mashup consumer / données, notamment grâce à l'environnement Popfly qui permet, en ligne, de créer rapidement des mashup à partir de blocks prédéfinis. Le développeur peut créer ses propres blocks pour ses projets et usages, même si officiellement Popfly n'est pas conçu pour des usages professionnels. Pour le côté pro du mashup, l'un des intérêts est de présenter, d'afficher, d'analyser, de composer des données opérationnelles.

A ces outils, on peut rajouter Convertigo Enterprise Mashup Server de Twinsoft. Là aussi orienté entreprise, on dispose d'un Composer pour créer et inté-

grer les widgets. Pour créer le mashup à proprement parler, on dispose d'un IDE Eclipse : Convertigo Studio. Composer repose sur un framework (provenant de Dreamface interactive).

Quand IBM s'y met aussi

IBM Lotus s'intéresse aussi à ce domaine avec Lotus Mashup. Orienté entreprise, il s'agit d'un outil fonctionnant dans un navigateur, compatible avec les mashup, permettant la création graphique et la connexion aux systèmes internes. La disponibilité est attendue pour mi-2008. Pour Lotus, il s'agit de proposer aux utilisateurs métiers de résoudre des problèmes ponctuels et de pouvoir s'insérer dans leur architecture SOA. Pour le déploiement, on passe par un serveur d'applications (IBM WebSphere 6.1). L'intégration avec le portail n'est

Software Technology Resources



Les auteurs les plus réputés nous font confiance ...



Microsoft®

Visual Studio® 2008

2 offres à ne pas manquer !

1

Les experts de Microsoft vous offrent leur DVD de ressources

pour l'achat de Visual Studio 2008 professional



280 heures d'auto-formation



Apprenez avec les Coachs MSDN



2

STR et Microsoft vous offrent en plus 100 € en chèque cadeau !

pour l'achat de Visual Studio 2008 avec MSDN Pro ou Premium

Avec MSDN, tous les mois, recevez en avant première les dernières technologies Microsoft sur votre bureau !

- ▶ Accès continu aux outils de développement VS 2008, mais aussi à VB6, VS.NET, VS2005.
- ▶ Tous les systèmes d'exploitation Windows dont Windows Vista et Windows Server 2008
- ▶ SQL Server 2005 avec l'arrivée de SQL Server 2008
- ▶ Les versions bêtas et les produits avant les autres
- ▶ 2 ou 4 appels par an au support et un site privé de téléchargement



Les chèques "universel" KADEOS que nous vous offrons sont utilisables sur la plupart des grandes enseignes (consultez la liste sur www.str.fr)

OFFRE exclusive STR

MSDN le meilleur choix pour acquérir Visual Studio

Visual Studio 2008 Pro avec MSDN Premium
abonnement 2 ans **2 758 € ht** Prix pour 2 ans

Visual Studio 2008 Pro avec MSDN Professional
abonnement 2 ans **1 216 € ht** Prix pour 2 ans

Avec chaque abonnement recevez le DVD de formation et 100 € de chèque KADEOS

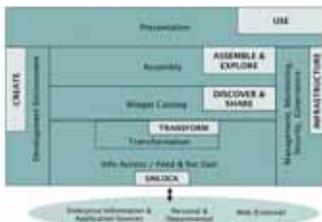
Validité de l'offre 27 juin 2008 - offre non cumulable avec d'autres offres ou remise supplémentaire accordée

ACHETEZ EN LIGNE sur www.str.fr

Informations et commandes Tél : 01 30 70 61 61 Fax : 01 39 46 38 64

Gros Plan

pas encore disponible. IBM Mashup Center constitue la plateforme de Big Blue pour assembler, créer, découvrir, partager les applications dynamiques web. Il est possible d'y découvrir les mashup, gadgets, services, feeds. On y trouve 3 composants centraux : Lotus Mashup, le catalogue et InfoSphere MashupHub. On dispose aussi d'un IDE pour créer des " Dynamic Widgets ". IBM utilise un terme intéressant : *Markecture*. Cela s'illustre par le schéma suivant :



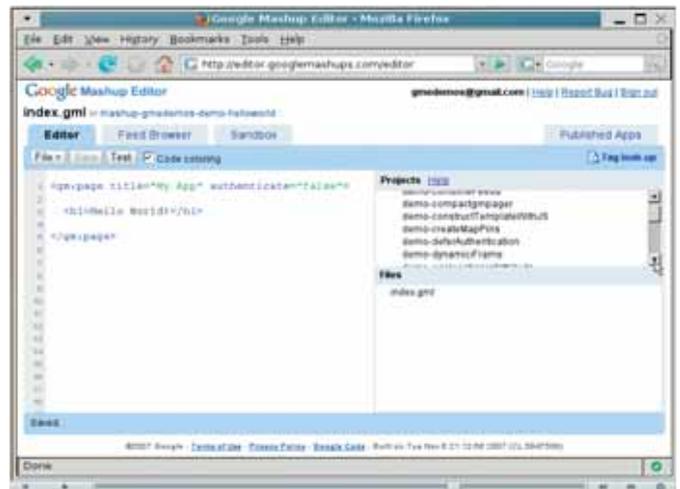
Lacunes et spécifications

Le mashup est pratique, souple, etc. Cependant, il faut aussi avoir conscience de ses limites. Tout d'abord, il n'existe pas de spécifications, de standards pour les mashup. Ainsi, chaque environnement devrait s'assurer de l'interopérabilité entre les différentes sortes de services, de gadgets par exemple. Assurez-vous que l'environnement choisi est capable d'interopérer avec tous les composants et services prévus dans vos mashup ! Faites des tests, du maquettage !

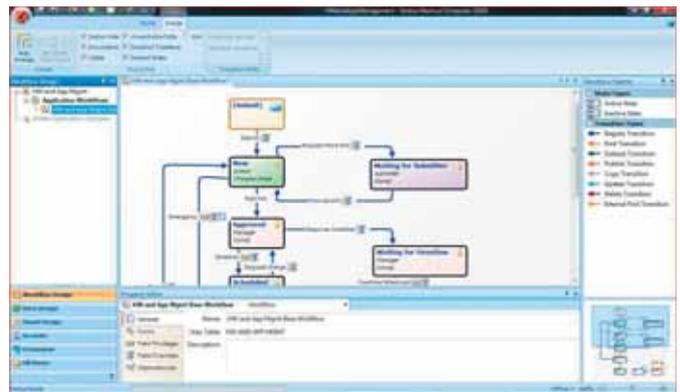
Sur l'interopérabilité, Tibco propose une brique de communication intéressante : PageBus. En quelque sorte, un bus Ajax pour connecter et échanger entre les composants Ajax. Le projet fait aussi partie d'OpenAjax. Sur ce terrain, récemment, la v1.0 d'OpenAjax Hub a été approuvée. Cette application spécifique doit fournir un standard autour de javascript / Ajax et assurer ainsi l'interopérabilité entre les nombreux frameworks Ajax disponibles. Il utilise le principe du " publier / abonnement ". La v1.1 doit apporter la sécurité mashup (notamment sur la sécuri-

sation de l'usage de données externes), et la communication client - serveur. La sécurité est aussi un élément souvent négligé dans le mashup. Utiliser des services, des données externes (dont on n'a pas la maîtrise) représente potentiellement une menace : code malicieux, code instable, service fantôme. Là aussi un standard de sécurité s'avère indispensable. Sur ce point sécuritaire, IBM a dévoilé IBM Smash qui permet de bien séparer code et données (prévenant ainsi toute manipulation de code, injection). L'éditeur a par ailleurs fourni Smash au futur OpenAjaxHub 1.1. Autre question, comment s'assurer de la qualité et de la disponibilité de chaque élément agrégé dans son mashup ? Là encore le mashup n'a aucun contrôle dessus. D'autre part, comment le tester efficacement avant publication ? Sur la partie infrastructure, plusieurs éditeurs comme BEA proposent des environnements serveurs pour déployer, administrer, contrôler les mashup. Ces outils devraient se multiplier rapidement. Vers quoi va aller le marché mashup ? S'agit-il d'une simple transition vers la SOA comme le présentait il y a quelques mois BEA ? Pas sûr, car le mashup est tout de même différent dans son utilisation mais complémentaire de la SOA. Ils peuvent parfaitement fonctionner ensemble et non s'opposer. Avec l'arrivée des nouveaux environnements et l'industrialisation de la conception au déploiement, il va sans aucun doute devenir une application comme une autre pour, comme précisé plus haut, résoudre des problèmes ponctuels, soulager une partie du département informatique. En plus, tout amateur du web et développeur peut s'amuser avec et compte tenu de l'explosion des services hébergés et des API, le mashup va pouvoir offrir de nouvelles possibilités !

■ François Tonic



L'outil Google



L'environnement Serena Composer

Quelques outils du marché

Google Mashup Editor	Editeur simplifié mais puissant pour le mashup. Pré-version.
Intel Mash Maker	Editeur s'intégrant dans les navigateurs pour créer et déployer. Pré-version.
Microsoft Popfly	Environnement graphique pour créer et déployer des mashup. Pré-version.
Lotus Mashup	Environnement utilisable dans une navigation pour créer, déployer, accéder aux systèmes à partir de mashup. Disponible mi-2008
Twinsoft Convertigo Studio	Editeur basé sur Eclipse pour créer les mashup.
Serena Mashup Composer	Environnement de création de mashup et de workflow. Orienté entreprise.
Tibco PageBus	Outil basé sur le publier et abonnement servant de bus Ajax pour assurer une communication / interopérabilité entre les composants Ajax
BEA AquaLogic Pages	Pour créer des applications web. Destiné avant tout aux utilisateurs. S'insère dans la gamme AquaLogic et notamment avec l'outil d'infrastructure AquaLogic Ensemble.
Kapow Mashup Server	Outil de création utilisant RSS, Atom, le contenu web, PDF, etc. Autorise le mashup de présentation (données), métier.
JackBe Presto 2.0	Solution mashup pour utilisateur avancé et développeur. Se connecte aux applications Flex, Java, Systinet, et aux principaux serveurs d'applications.
Yahoo Pipes	Editeur simplifié pour créer des mashup. On manipule les données via les métadonnées.

Popfly : créer votre mashup sans une ligne de code !

Lancé il y a quelques mois par les équipes Microsoft Labs, Popfly vous offre la possibilité de créer en moins de 5 minutes des applications web 2.0 / mashup. Grâce à son interface Silverlight, Popfly permet de créer ses propres applications de manière intuitive et dynamique, tout en restant accessible au grand public.

Aujourd'hui, de nombreux sites web communautaires proposent des API permettant d'exploiter leurs services sur des sites tiers. Ces derniers apportent une plus value en exploitant habilement ces API pour mettre en relation des données provenant de plusieurs sources différentes. La majorité de ces API requièrent de programmer en JavaScript, ce qui limite leur utilisation aux développeurs web. Popfly tente de repousser cette limite en donnant la possibilité à n'importe qui d'exploiter ces services web au travers de blocs que l'on manipule à l'aide de glisser-déposer. Le développement se fait alors de manière graphique et intuitive. C'est là tout l'intérêt de Popfly ! L'objectif de cet article est de vous présenter quelques-unes des capacités de cette technologie au travers d'un exemple simple. Nous verrons ensuite comment étendre les possibilités de création offertes par l'outil.

Vos photos Flickr sur Virtual Earth

L'un des exemples les plus probants sous Popfly est d'afficher des photos géo-localisées provenant de Flickr sur une carte de Virtual Earth.

Une fois loggé sur le site www.popfly.com avec votre live Id, allez dans la section create stuff puis mashup. Avant toute chose, il est nécessaire de connaître l'interface de développement mashup de Popfly. Sur la gauche se trouve une bibliothèque contenant tous les blocs disponibles pour votre mashup. Trois types de blocs sont présents :

- les blocs *source* (blocs d'accès aux données) situés en début de chaîne qui permettent d'obtenir les données.
- les blocs *de traitement* optionnels situés en milieu de chaîne adaptent les données pour les blocs suivants.
- Les blocs *d'affichage* (un seul par mashup) situés en bout de chaîne permettent d'obtenir un rendu graphique des données. Ils implémentent Silverlight et offrent donc toutes les possibilités de ce dernier au niveau graphique.

Au centre, se situe le *workspace* dans lequel le mashup sera créé. Et sur la droite, un onglet qui peut être masqué vous aide dans la création de vos mashup. Maintenant que vous avez pris connaissance de l'environnement de travail de Popfly, nous rentrons dans le vif du sujet ! Dans la bibliothèque, prenez le bloc Flickr, faites-le glisser sur le workspace et faites de même avec le bloc VirtualEarth. Connectez ensuite la sortie du bloc Flickr à l'entrée du bloc VirtualEarth et il ne restera plus qu'à configurer. Pour cela, cliquez sur la petite clé du bloc Flickr, ce qui aura pour effet d'afficher le menu de configuration du bloc. Dans le menu déroulant en haut sélectionnez *getGeotaggedPhotos*, puis dans la case text



mettez un mot-clé de votre choix (exemple : *sunset*). La configuration de ce bloc est finie. Dans le cas présent, il n'est pas nécessaire de configurer le bloc *VirtualEarth*, car on se contente des valeurs par défaut. Il ne reste donc plus qu'à apprécier le résultat : cliquez sur *preview* dans le menu du haut et admirez ! De jolies photos de Soleil couchant apparaîtront sur une carte *VirtualEarth*. Notons que l'exploitation de photos Flickr nécessitera une API KEY lorsque que vous voudrez exporter votre mashup. Cependant cette dernière n'est pas utile pour la *preview*.

Edition du bloc Flickr

Nous avons vu qu'il était très simple d'assembler un mashup en liant des blocs. Cela nous limite tout de même aux traitements déjà implémentés et un besoin spécifique nécessitera la création d'un bloc. Heureusement cela est prévu dans Popfly, mais il faut mettre les mains dans le cambouis et coder un peu de JavaScript. Une chose très pratique est qu'on a un accès libre au code source de tous les blocs publics. Nous allons d'ailleurs éditer le bloc source Flickr afin de l'adapter à nos besoins.

Un bloc Popfly se décompose en deux fichiers, un fichier XML qui décrit les entrées sorties et un fichier JavaScript avec le traitement à effectuer. Il est créé via l'interface en ligne Silverlight mais il peut être plus pratique de passer par un IDE (comme *Visual Studio 2008*) et de recopier ensuite le code. Les blocs d'affichage sont un peu spéciaux car en plus du JavaScript ils implémentent Silverlight. Il faut donc via différentes astuces intégrer un fichier XAML dans leur code JavaScript.



Le fichier XML

Allez dans la section *create stuff* puis *block*, vous tomberez sur un fichier XML quasi-vierge, puis sélectionnez dans le menu de gauche le bloc *Flickr*, et le fichier XML sera automatiquement remplacé par celui du bloc sélectionné. Les premières balises donnent une description générique du bloc et de ses auteurs. Ensuite la balise *keys* gère l'api Key requise pour certains blocs (comme celui-ci). La balise *operations* décrit les entrées-sorties des différentes fonctions disponibles dans le bloc, qui seront sélectionnables via un menu déroulant lors de la configuration du bloc dans un mashup. Enfin la balise *object* décrit des structures de données que le bloc utilise en sortie et qui n'existent pas par défaut. Dans notre exemple il s'agit de l'objet *photo* avec, entre autres, son url et ses coordonnées. Voyons plus en détail une des opérations *getUserPhotos* qui permet d'aller chercher les photos d'un utilisateur.

```
<operation name="getUserPhotos" callMode="sync">
<description>Get a Flickr users's photos.</description>
<inputs>
<input name="userName" required="true" type="userName">
<description>UserName</description>
<defaultValue>popflyteam</defaultValue>
<constraints />
</input>
<input name="number" required="false" type="integer">
<description>maximum number of photos to return</description>
<defaultValue>15</defaultValue>
<constraints />
</input>
</inputs>
<outputs>
<output isArray="true" type="custom" object="Photo" />
</outputs>
</operation>
```

On remarque qu'il y a deux entrées : le nom de l'utilisateur et le nombre minimum de photos à récupérer. Pour apprécier la facilité avec laquelle on peut modifier un bloc nous allons ajouter une entrée nommée *text* qui permettra de ne sélectionner que les photos portant certains mots clés parmi celles de l'utilisateur ciblé. Il suffit alors d'ajouter la balise suivante à la suite des balises *input*.

```
<input name="text" required="false" type="string">
<description>text in title, description, or tags</description>
<defaultValue>imagine cup</defaultValue>
<constraints />
</input>
```



Le fichier JavaScript

Passons maintenant à la partie JavaScript, sélectionnez l'onglet *Block Code* : Le code apparaît et l'on remarque que sont proprement séparées les différentes fonctions qui correspondent à celles décrites dans la partie XML. Prenons celle qui nous intéresse pour y ajouter la sélection de mots-clés : *getUserPhotos*. Commençons par changer la signature de la fonction pour lui ajouter un argument :

```
FlickrClass.prototype.getUserPhotos = function (username, number, text) ;
```

Puis il suffit d'éditer les lignes *var param* et *var photos* de la sorte :

```
var params = "&user_id=" + escape(encodeURIComponent(username)) + "&text=" + escape(text);
var photos = this._getFlickrPhotos("flickr.photos.search", number, params, apiKey);
```

Nous avons fini l'amélioration du bloc. Vous pouvez maintenant l'enregistrer en haut à droite sous le nom *flickr2*, il sera disponible dans la section *My Blocks* de la bibliothèque. Vous pouvez maintenant l'utiliser dans un mashup en reprenant les étapes vues au début de l'article.

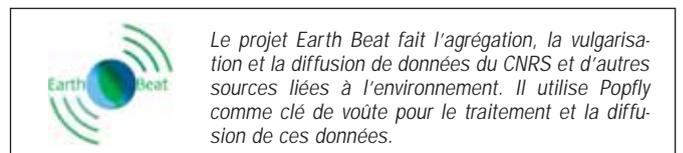
Simple, efficace mais pas très pro. Nous avons vu que *Popfly* permet de créer simplement une application *web 2.0*, sans langage de programmation. De plus, ses possibilités peuvent être étendues simplement en créant un bloc avec JavaScript et Silverlight. Cependant *Popfly* souffre de quelques défauts de jeunesse dus à son interface Silverlight, encore peu répandue, et au logo *Popfly* qui s'invite sur tous les blocs créés et qui limitera certainement son utilisation en milieu professionnel. Enfin, il faut garder en tête qu'il s'agit encore d'une version bêta en constante évolution. La dernière version de *Popfly* permet maintenant de créer facilement un site web ou un jeu vidéo !



■ Eric Hauchecorne, Martin Briec & Arnaud Angelo

Etudiants à L'ESIEE et participants au concours *Imagine Cup* de Microsoft dans la catégorie **Software Design** avec le projet **Earth Beat**.

<http://www.earthbeat.fr>



Le projet *Earth Beat* fait l'agrégation, la vulgarisation et la diffusion de données du CNRS et d'autres sources liées à l'environnement. Il utilise *Popfly* comme clé de voûte pour le traitement et la diffusion de ces données.



Infragistics NetAdvantage®

Un potentiel sans limite pour Windows Forms, ASP.NET et WPF

Les meilleurs composants pour interfaces utilisateur disponibles

Tous les composants essentiels pour la conception et le style d'applications corporatives irrésistibles

Une expérience utilisateur hors du commun

Concevez des IHM exceptionnelles et ergonomiques pour multi-plateformes

Un cycle de développement accéléré

Réduisez vos délais de développement et assurez une livraison rapide avec des composants prêts à l'emploi

Une utilisation facilitée

Concevez des IHM reproduisant fidèlement le look & feel des dernières technologies Microsoft® avec un minimum de code

Retour sur investissement

Focalisez vos attentions sur l'aspect métier et permettez à la boîte à outils Netadvantage d'assurer la cohérence, le style et l'utilisation simplifiée de l'interface de vos applications pour Windows Forms, WPF et WEB 2.0

Support complet pour l'IDE de Visual Studio®

Conçu pour compatibilité avec les plateformes de développement de Microsoft y compris Visual Studio 2008

Pour de plus amples informations:

infragistics.com

 0800 667 307

Intel Mash Maker : l'autre outil de mashup !

L'une des surprises principales de l'Intel Developer Forum 2008, non pas par son importance stratégique pour le fondeur, mais par le côté inattendu de la chose, fut l'annonce d'un logiciel de création de mashup : Intel Mash Maker. Apportant sa pierre à l'édifice, Intel propose une alternative assez novatrice aux outils existants.

L'une des innovations d'Intel Mash Maker est que l'éditeur a mis en avant l'utilisation des mashup plus que leur création. La plate-forme exploite pleinement le concept très à la mode du réseau social, car les utilisateurs sont invités à partager leurs widgets et mashup entre eux. Cette disposition est garantie par la licence d'utilisation de l'API utilisée pour la création des widgets et mashup. Alors qu'il est encore en phase bêta, le système jouit déjà d'un succès appréciable.

Installation

L'outil a pour autre particularité de se présenter sous la forme d'une extension de navigateur web. Initialement, seul Firefox était supporté, mais Intel a rapidement proposé une version propre à Internet Explorer 7. Au moment du téléchargement, spécifiez la version de votre choix et enregistrez-vous pour pouvoir vous créer un compte sur le réseau (le mot de passe vous est automatiquement envoyé par email). L'installation terminée, il vous faut rentrer ces informations de connexions dans la configuration de votre extension (ceci vous sera demandé à chaque lancement de votre navigateur tant que vous ne l'aurez pas fait).

Utilisation " client "

Pour tester votre outil de mashup, rendez-vous simplement sur la page de recherche d'Amazon US, et effectuez n'importe quelle recherche, cliquez sur le bouton " Price Comparison " de la barre d'Intel Mash Maker, puis observez ! Si vous ne pouvez pas faire le test, voici une démonstration (Figure 1, 2, 3) Le principe est

un clic pour activer les mashup publics...



Amazon sans Intel Mash Maker...

et avec !



simple : quelqu'un a créé ce mashup de comparaison de prix (ou plus exactement de génération de graphiques utilisant les prix des produits trouvés comme données), puis l'a publié. Pour l'utiliser, vous n'avez qu'à l'activer. Certaines options sont personnalisables, comme la position

des widgets. Rien de très compliqué en somme ! Le site officiel présente une large galerie.

Création de mashup

Si Intel Mash Maker est très pratique pour ce qui concerne l'utilisation des mashup, il ne faut pas oublier qu'il s'agit avant tout d'une plate-forme de création de mashup ! Là aussi, Intel a fait des choix hors des sentiers battus. Le plus significatif concerne les sources de données : il a été pris le parti de ne pas recourir aux web services, mais au contenu HTML directement. Tous les outils de création sont rassemblés dans un panneau latéral situé à gauche de l'écran, qui s'ouvre en cliquant sur la deuxième icône de la barre d'outils. Les quatre onglets qui le composent contiennent respectivement les widgets, les mashup

disponibles pour la page en cours, les données identifiées dans la page et les extractors (instructions appliquées pour extraire les données du code HTML).

Cette dernière fonctionnalité constitue le coeur du principe d'agrégation de l'environnement : à chaque page correspondent un ou plusieurs extracteurs de données, qui sont créés pour l'essentiel en sélectionnant visuellement des éléments de la page (div, span, paragraphes, éléments de liste, etc.), Ceux-ci sont ensuite utilisés par les widgets pour produire de nouveaux affichages, comme dans le cas du " Price Comparison " sur Amazon, ou encore de la carte vous indiquant la situation géographique de vos contacts Facebook avec " Face map all ". Chaque page d'un site Internet peut ainsi être consultée comme le serait un service web, et exploitée sous forme de champs d'information identifiés. Ces données, nous l'avons dit, peuvent être utilisées pour alimenter des widgets prêts-à-l'emploi. L'étape suivante, bien plus intéressante, consiste à créer ses propres widgets. Ceux-ci peuvent respecter la norme Google Widgets, ou bien être réalisés à partir de l'API maison d'Intel. N'hésitez pas à consulter la documentation sur le site officiel, mais également les sources des widgets standard (en obtenant l'URL de leur définition XML grâce aux " experts tools ") pour apprendre rapidement comment créer tout type de widget personnalisé pour créer vos propres mashup ! Site officiel :

<http://mashmaker.intel.com>

■ Gauthier Delamarre

LEONARDI

...LYRIA REJOINT LE GROUPE W4 !
L'UNION DU BPM ET DE L'IHM MD...
A SUIVRE...



ihm

EN TOUTE SIMPLICITÉ !

Votre application de gestion 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 ? ”

NOUVELLE
VERSION!
LEONARDI
V4.0
OPEN SOURCE

Si votre réponse est moins d'une semaine, inutile de vous rendre sur notre site, ni de télécharger la version gratuite de LEONARDI, sinon il est temps de passer à la vitesse "Model-Driven"...



LYRIA

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

Mettez-vous aux TESTS!

1^{re} partie



Véritable serpent de mer du développement, les tests sont plus que jamais d'actualité. Souvent encore perçu comme une variable d'ajustement par les entreprises pour tenir les délais, pas toujours très considéré par le développeur, le test est pourtant au cœur de tout projet, de toute programmation. Il est vital pour la sécurité des applications et des données et il est au cœur des cellules qualité dans les entreprises.

Les tests recouvrent en réalité une multitude de formes, d'outils, de pratiques. Surtout, ils interviennent à tout instant de la programmation, à la phase de conception, on peut piloter le développement par les cas de tests (approche MDT, Model Driven Testing, notamment). Et chaque module, chaque classe peut recevoir un test spécifique. Les tests unitaires sont aujourd'hui les plus utilisés par les développeurs, essentiellement grâce aux divers frameworks comme Junit, PHPUnit.

Il y a autant de types de tests que de scénarios. Ainsi au sein d'un même projet, on aura différents types de tests : au plus près du

code jusqu'aux tests de déploiement, de stress, de montée en charge, sans oublier les tests d'interface, de non régression, etc. Et là à chaque catégorie, correspondent des bonnes pratiques, des outils spécifiques.

On ne fait pas un test de charge comme on ferait un test d'interface.

Dans ce dossier spécial tests, en deux parties, nous allons aborder aussi bien les bonnes pratiques que la mise en œuvre avec différents outils, différents langages. Il est important de mêler les deux aspects. Car on ne peut pas réaliser de bons tests si on ne comprend pas la finalité de chacun et les réflexes à avoir. Et si vous avez peur de man-

quer d'outils, ne vous inquiétez pas, c'est même la profusion quels que soient les tests. Et le défi des tests s'accroît avec les évolutions des environnements, des langages, des applications, des architectures. Avec des plates-formes comme RIA, RDA, les mashup, javascript, etc. le test reste un réel souci car l'outillage n'est pas performant, voire inexistant. Pour les éditeurs, ce manque d'outils est d'ailleurs un problème. Peut être qu'à la lecture de ce dossier, vous aurez envie de devenir testeur et d'en faire votre métier. Encore peu répandu en France, le testeur est pourtant un métier d'avenir. Alors pourquoi pas vous ?

■ François Tonic

Les meilleures pratiques du test logiciel

Souvent sous-évalué et jugé peu digne d'intérêt par les développeurs, le test n'en est pas moins un maillon crucial de l'ingénierie logicielle, et un pilier fondamental sans lequel aucun logiciel ne peut être mis sur le marché avec la qualité nécessaire. L'objectif de cet article est de vulgariser les dimensions majeures du test logiciel et de donner un retour d'expérience des meilleures pratiques telles que nous les percevons chez SoftFluent et utilisées dans nos projets.

Un investissement souvent insuffisant dans la qualité

Le test est une vaste discipline qui permet de s'assurer de la qualité des logiciels. Cette qualité comprend différentes dimensions que l'on peut regrouper dans les trois catégories suivantes : les fonctionnalités, l'ingénierie et l'adaptabilité.

Pourtant, le test fait souvent l'objet d'un investissement insuffisant de la part des éditeurs de logiciels ou des équipes informatiques. Comme pour la sécurité, le sentiment de dépenser du temps perçu comme improductif prime sur les bonnes pratiques qui recommandent de prévoir au minimum **un testeur pour deux développeurs**. La réalité observée sur le terrain peut être jusqu'à deux fois moindre.

En France, ce phénomène est probablement accru par le poids important des sociétés de services et des développements spécifiques. Leur importance relative est plus forte qu'ailleurs, en partie pour des raisons culturelles, et par conséquent, la culture des logiciels "packagés" est plus faible. Les sociétés de services ont en effet une sensibilité moindre que les éditeurs de logiciels à la qualité, car l'impact d'une correction est souvent limité. A l'inverse, un éditeur de logiciel doit investir sur la qualification de son logiciel en amont, sous peine de payer très cher les multiples déploiements de son produit.

Dans des sociétés comme Microsoft, où les logiciels sont déployés par dizaines de millions d'exemplaires, certaines équipes comportent jusqu'à deux testeurs par développeur !

Les différents niveaux de tests

Tout d'abord, notons que les tests logiciels se font en général à plusieurs niveaux. On distingue potentiellement quatre principaux niveaux de tests :

- **Les tests unitaires**, qui sont menés par le développeur lui-même, et qui consistent à vérifier la bonne exécution des fonctions dont il a la charge. Ces fonctions sont tes-



tées de manière indépendante, souvent avec des données réduites. Il est souhaitable de définir très tôt des jeux de tests représentatifs, car le développeur pourra réaliser les tests unitaires sur cette base et la qualité en sera améliorée.

- **Les tests d'intégration**, qui sont menés par un testeur dédié, au sein de l'équipe de développement ou d'une équipe qualité travaillant en coopération étroite. La particularité de ces tests est que ceux-ci visent à tester la mise en commun de plusieurs composants et l'enchaînement de processus complets au-delà des simples fonctions unitaires. On parle parfois de tests en boîte blanche, dans la mesure où l'on est amené à regarder les relations entre les composants pour vérifier leur bonne coopération. Là encore, la disponibilité de jeux de tests tôt dans le cycle est un facteur important d'efficacité.
- **Les tests du système** complet ou la **qualification**, qui consistent à dérouler des scénarios complets, représentant les cas d'utilisation du logiciel, sans se préoccuper de l'implémentation ou des composants sous-jacents. On est alors dans une logique

de boîte noire, et il est fortement recommandé que l'équipe qui réalise ces tests soit distincte de l'équipe de développement. Elle peut même parfois être mise en "off-shore", ce qui a le mérite de nécessiter un formalisme assez détaillé, et pousse à une bonne description de l'ensemble des scénarios d'utilisation. Lors de l'évolution du système, on rejouera ces scénarios afin de valider la **non-régression**.

- **Les tests d'acceptation**, menés avec des utilisateurs pilotes, afin de valider l'adéquation du logiciel au métier et sa facilité d'adoption par ceux qui devront l'utiliser régulièrement. De la même manière, associer très tôt les utilisateurs à une première portion du logiciel permettra de connaître les critères d'acceptation par les utilisateurs, et au besoin d'ajuster certains choix, par exemple ergonomiques.

Notre conviction sur les tests unitaires, d'intégration et du système, est que pour des logiciels métiers, il convient d'automatiser au maximum ces tests. Un test déroulé manuellement est une charge. Un test consolidé dans un script rejouable à l'infini est un investissement. L'automatisation des tests unitaires et

de certains tests d'intégration permet d'en extraire des sous-ensembles re-jouables lors de la génération elle-même automatisée du système.

Par ailleurs, les artefacts de tests comme le code des tests unitaires, leurs fichiers de configuration ou les jeux de données, participent à la documentation du logiciel nécessaire pour sa maintenance et comme exemples d'utilisation des interfaces publiques (API). Ceci est particulièrement important dans les systèmes exposant certaines fonctionnalités sous formes de services, ou s'intégrant dans d'autres systèmes.

Les différentes natures de tests

Par ailleurs, au-delà de ces différents niveaux de tests, on distingue des tests différents par leur nature :

- **Les tests fonctionnels** valident le bon comportement du logiciel en matière de service rendu. Ces tests sont souvent ceux décrits en détail dans les scénarios et sur lesquels l'essentiel de l'investissement est fait, par analogie aux spécifications qui décrivent souvent en détail la fonction à remplir. Notre recommandation sur le sujet rejoint les points évoqués ci-dessus. Ecrire les scénarios très tôt est fortement souhaitable, car cela valide la bonne analyse fonctionnelle, tout en donnant de la matière au développeur pour ses propres tests unitaires. Mais mettre l'exclusivité de l'effort sur les tests purement fonctionnels est certainement une erreur fréquente.
- Les tests de **compatibilité de plate-forme** comprennent au minimum des tests d'installation et de bon fonctionnement sur des environnements ciblés (système d'exploitation, base de données...), voire pour les éditeurs des tests plus poussés détectant les problèmes potentiels avant certification pour une plate-forme donnée. Ils permettent de détecter le respect des bonnes pratiques telles que l'utilisation de répertoires virtuels, l'exécution sans droits d'administration, la désinstallation, la localisation des ressources etc. Les postes des développeurs ne doivent pas être confondus avec l'environnement d'exécution cible.
- Les **tests de robustesse** visent à tester le logiciel dans des conditions dégradées ou aux limites. Ils sont rarement menés de manière systématique, alors qu'ils ne sont pas compliqués à mettre en œuvre : débranchement d'un câble, arrêt brutal d'une

10 conseils à retenir

1. Le test est un vrai métier, il faut de réels rôles de testeurs distincts des développeurs.
2. Le test ne doit pas être sous-évalué, de l'ordre de 25 à 30% de l'effort d'un projet de développement.
3. Les jeux de tests doivent être réalisés tôt dans le cycle pour lever les erreurs de conception et faciliter le travail des équipes de développement.
4. Une application doit toujours être conçue de manière à être testable, et un bon découpage en composants est une aide précieuse sur ce point.
5. Il convient d'investir sur toutes les dimensions du test y compris l'acceptation par les utilisateurs et l'ergonomie.
6. Il faut définir les conditions d'exploitation du système et tester les configurations, la montée en charge ou le comportement en cas de coupure réseau.
7. C'est toujours une bonne idée d'investir sur l'automatisation pour l'ensemble des tests et en particulier pour valider la non-régression.
8. Les tests sont des développements qui font partie du cycle de gestion des versions.
9. Comme pour le développement, un travail partiel bien ciblé permet de couvrir l'essentiel selon une règle de 80/20. Il vaut mieux commencer par cela et améliorer par itération, que chercher à tout prix une exhaustivité utopique.
10. Une approche statique fortement typée et pilotée par les modèles facilite les tests, à l'inverse d'une approche dynamique basée sur des métadonnées, qui est très souple pour le développeur, mais aboutit à une découverte tardive des difficultés.

machine... Il est important de savoir comment le logiciel réagit en ces circonstances et de définir le scénario de reprise. Faire l'impasse permet de gagner (à peine) un peu de temps à court terme, mais laisse planer un risque dont les conséquences financières sont souvent impossibles à évaluer. Il est évident que l'utilisateur sera tôt ou tard confronté à une panne réseau, alors autant prévoir de suite le comportement du logiciel et documenter le symptôme.

- Les **tests de performance** sont souvent menés en fin de projet. Les facteurs influant sur la performance étant nombreux, il est risqué de tirer des conclusions hâtives sur des tests partiels en l'absence de volumes représentatifs et de présence de l'ensemble des conditions d'exécution. Cependant, les développeurs peuvent très bien tester le comportement de certains services unitaires tôt dans le cycle lorsque ceux-ci présentent un risque de performance (il s'agit d'une sorte de test unitaire de performance). De plus, l'expérience prouve que les résultats de ces tests doivent surtout être précisément documentés afin de formaliser les conditions dans lesquelles les mesures ont été prises et les conclusions tirées doivent être analysées au vu de compteurs significatifs : consommation CPU, évolution de l'utilisation

mémoire, nombre de requêtes par seconde, flux d'entrée/sortie...

- Les **tests de montée en charge** doivent être distingués des tests de performance. Une application peut être performante mais limitée dans sa capacité à soutenir un nombre croissant d'utilisateurs (ou tout autre paramètre). Un système logiciel, particulièrement s'il est destiné à l'Internet, doit être conçu pour pouvoir monter en charge de manière illimitée par simple ajout de matériel. Des tests doivent être menés en ce sens, avec des outils permettant de simuler de nombreux clients simultanés.
- Les **tests d'ergonomie** sont rarement menés car ils nécessitent d'associer des utilisateurs finaux. Pourtant, ils peuvent être un facteur très important de qualité réelle et perçue, en améliorant le confort d'utilisation des logiciels. Ces tests incluent tout autant la facilité d'utilisation (le " Feel ") que l'apparence visuelle (le " Look ").

Favoriser la détection précoce des anomalies

Le lecteur l'aura compris au travers des différents commentaires des paragraphes qui précèdent, chez SoftFluent, nous sommes plus proches des méthodes agiles que cascadiées, et nous privilégions tout ce qui permet de dis-

poser de cycles courts, pour le test comme pour la bonne définition des orientations fonctionnelles d'ailleurs. La raison est avant tout économique.

Le coût d'une anomalie est très différent suivant la phase lors de laquelle l'anomalie est introduite et la phase lors de laquelle celle-ci est détectée. Ainsi, si une erreur de conception, détectée dès la phase de spécifications coûte 1 à corriger, la même erreur révélée lors des tests de qualification coûtera 10, et potentiellement jusqu'à 100 si le logiciel est déjà déployé. Le tableau ci-contre estime les rapports pour différents types d'anomalies en fonction de ces phases. C'est pourquoi nous cherchons à valider un maximum de points en amont, avant qu'un effort important de développement, qui est par nature coûteux, ait été consommé. De plus, les anomalies sont plus faciles à comprendre et corriger sur un périmètre limité qu'au sein d'une application com-

Introduction/Détection	Spécifications	Architecture	Réalisation	Qualification
Spécifications	1	3	5	10
Architecture	-	1	10	15
Réalisation	-	-	1	10

plète obligatoirement porteuse d'une certaine complexité.

L'évolution vers les méthodes agiles

Enfin, d'un point de vue humain, les méthodes agiles imposent une collaboration permanente, et cette collaboration entre les experts fonctionnels et les développeurs est le secret d'un logiciel réussi. Dans les méthodes plus traditionnelles, un formalisme poussé est souvent un moyen de se rassurer sur le plan théorique, mais c'est aussi un moyen pour les développeurs de s'abriter derrière des écrits pour ne plus effectuer de modifications, même si celles-ci répondent à un réel besoin. Le phénomène se produit également parfois

avec les testeurs, alors que le testeur est l'ami du développeur, puisqu'il lui révèle ses erreurs ! En guise de conclusion, retenons que le test est un métier noble, qui comprend de multiples facettes, mais reste insuffisamment reconnu et valorisé, alors qu'il joue un rôle fondamental dans la qualité logicielle. Chacun pouvant par ailleurs évaluer son expérience quotidienne avec l'informatique, on conviendra aisément que des progrès restent à faire dans le domaine ! Alors que tous les développeurs et informaticiens se mobilisent pour redonner au test la place qu'il mérite.



■ Daniel COHEN-ZARDI
Président de SoftFluent

Les différents types de tests et les tests de régression

Les applications sont de plus en plus nombreuses et complexes. Elles évoluent rapidement au sein de technologies et d'environnements hétérogènes. Réussite et rentabilité d'un projet passent maintenant par un suivi, tout au long du cycle de développement : de la qualité et de la réalisation des tests logiciels. Le choix parmi les différents types de tests tient une dimension incontournable de la gestion de projet. Encore faut-il en comprendre les enjeux et les types de tests.

Bien que le terme " types de tests " soit très largement usité, son sens mérite d'être précisé, tant il est empreint des us et coutumes des différentes organisations. Notons au passage que cette imprécision de vocabulaire est souvent à l'origine de confusions qui tendent les relations entre interlocuteurs (développeurs, testeurs, chef de projets, etc.) impliqués dans les tests.

Nous pouvons retenir trois axes de classifications des " types de tests ", selon :

A. Les phases du processus de développement : c'est dans cette catégorie qu'il faut ranger: les tests unitaires, les tests d'intégration, les tests de recette, les tests de VABF (vérification d'aptitude au bon fonctionnement), les tests de VSR (vérification de service régulier).

B. La caractéristique qualité vérifiée : c'est

dans cette catégorie qu'il faut classer les types suivants : fonctionnalité (règles de gestion, ou bout en bout (processus)), performance (regroupant temps de réponses, tenue en charge et points de ruptures), intégrité (et traçabilité), sécurité (d'accès et de fonctionnement), disponibilité, interopérabilité, exploitabilité, maintenabilité, portabilité, ergonomie, efficacité (procédures), ...

C. La nature technique des " objets " testés : tests d'interfaces, tests de batch/transactionnel, tests techniques (installation, sauvegarde, plans de reprises). Ces désignations sont en fait des cas particuliers de certains critères qualité listés en B : par exemple, les tests d'interfaces ne sont qu'une partie des tests permettant de vérifier l'interopérabilité d'un système.

Inclassable, la notion de " tests de non régres-

sion " s'affiche comme un sur-ensemble de tout ou partie des trois classes précédentes appliqué aux parties non modifiées du produit ! Cet ensemble décrit un éventail large des tests réalisables sur un produit.

Pourquoi, quand et comment une société met-elle en place ces types de tests ?

C'est l'objet principal de la stratégie de test (éventuellement décomposée en stratégie générale et détaillée si la composition du logiciel le nécessite) :

- Etablir la couverture a priori des types de tests souhaités ; en utilisant une classification B (aujourd'hui normalisée). Le choix des " caractéristiques qualité " vérifiées pour chaque exigence du produit est déterminant. C'est lui qui permet de régler la profondeur

Dossier

de test en fonction par exemple des risques en cours.

- Répartir les différentes activités de tests sur le cycle de vie du produit ; ce qui revient à utiliser la classification A.
- Répartir les différentes activités de tests entre les acteurs et les environnements à partir du contexte organisationnel de l'entreprise et des architectures existantes.

Le " comment " définit : quelle technique de test utiliser ; une technique de tests est une méthode formelle de fabrication des scénarii et des cas de tests attachés à la vérification d'un critère qualité donné. Il y a autant de techniques de tests que d'éléments de la classification C.

Tous les acteurs du cycle de vie d'une application doivent avoir un référentiel commun qu'ils partageront pendant tout le projet. La traçabilité (de plus en plus importante) est une des clés de la qualité. C'est pourquoi, la valorisation du test représente un enjeu majeur, motivé par une réalité économique. L'activité du test est complexe. Il est donc nécessaire de professionnaliser la discipline avec des spécialistes capables de manager les tests. La tendance va vers les centres de qualification logiciels dont le rôle est de définir la politique

et la stratégie de tests, coordonner tous les acteurs, de gérer les moyens, et de prendre en responsabilité le plan d'évolution.

Il est également indispensable de s'inscrire dans un processus d'amélioration continue et de reproduire régulièrement cette démarche, faisant ainsi du test une activité permanente, tout en accompagnant le changement tout au long de la démarche.

La stratégie de test vise à répartir au mieux les objectifs, focaliser les tests sur les exigences critiques et permettre de capitaliser les savoirs.

Le test de régression : incontournable et vital !

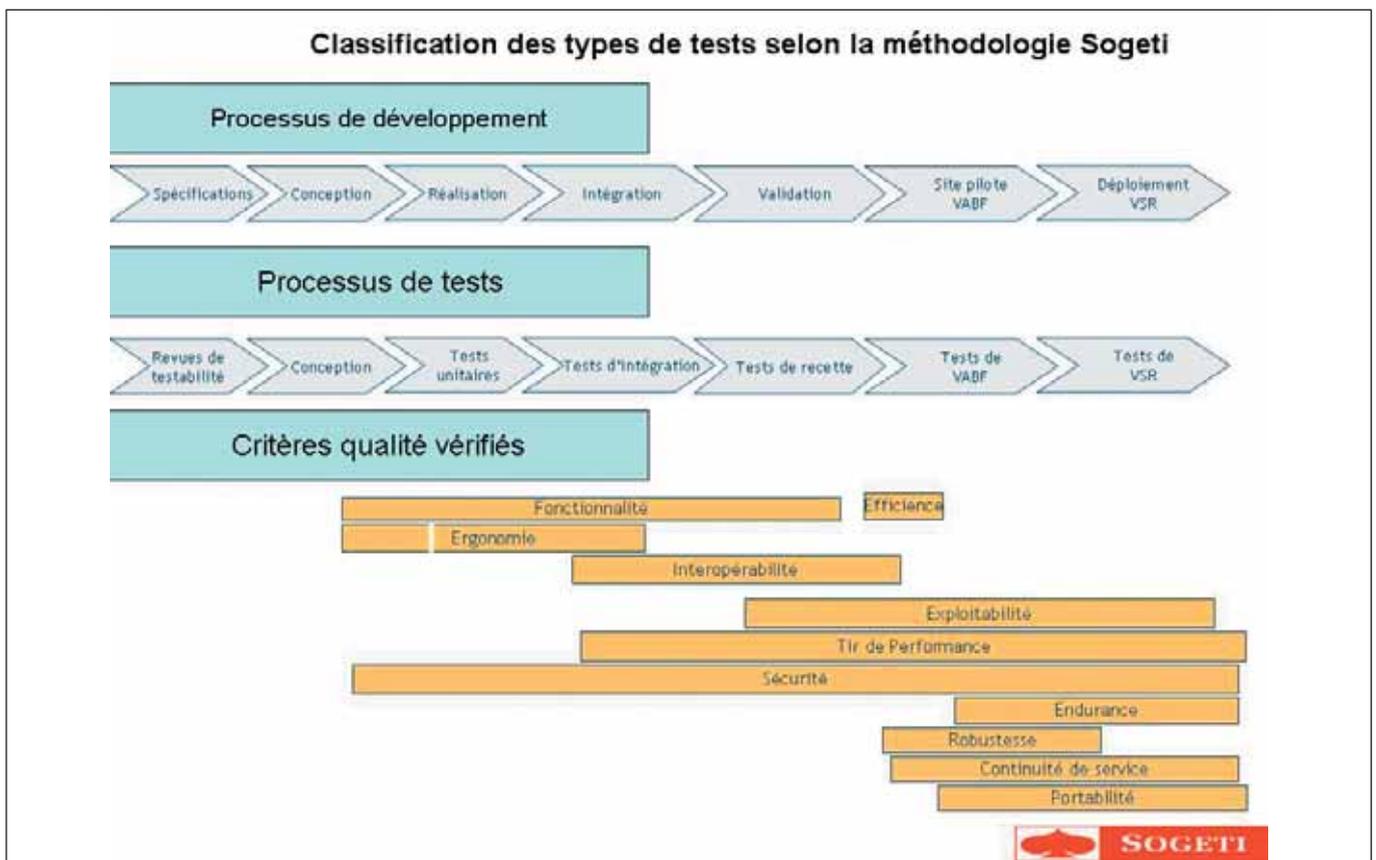
Ce test fait suite à une modification de l'application (du fait d'une mise à jour ou de l'aboutissement du test de recette) : il faut alors vérifier que l'application n'a pas perdu de fonctionnalités lors de l'ajout de nouvelles fonctionnalités ou la modification de fonctionnalités existantes.

Pour spécifier des tests, les testeurs ou les développeurs ont à leur disposition un ensemble de techniques aux caractéristiques multiples : approche " boîte blanche " ou " boîte noire ", test statique ou dynamique,

revue, check-list, analyse des valeurs limites, classe d'équivalence, table de décision, ... Bien que de nombreux ouvrages existent sur le sujet, peu de professionnels utilisent explicitement ces techniques. La qualité d'un cas de tests repose alors plus sur l'expérience de son auteur que sur une approche structurée. Pourtant les atouts des techniques de spécification des tests sont nombreux : maîtrise de la couverture des tests, mise en application d'une stratégie de tests définie, capitalisation et réutilisation des cas de tests. Bien évidemment la formation des testeurs et des développeurs n'est pas la seule condition à l'emploi des techniques de spécification des tests, il est nécessaire également de disposer d'une documentation de référence complète et de qualité afin de produire des cas de tests sans ambiguïté.

Ces techniques de spécifications s'appuient sur différents points. Une technique de test est une combinaison d'actions afin de produire un produit de test universel.

Différentes techniques peuvent être abordées :
- L'évaluation d'essai : des évaluations peuvent être effectuées à différents niveaux (estimations par niveau de test, par phase de test, par activité de test)

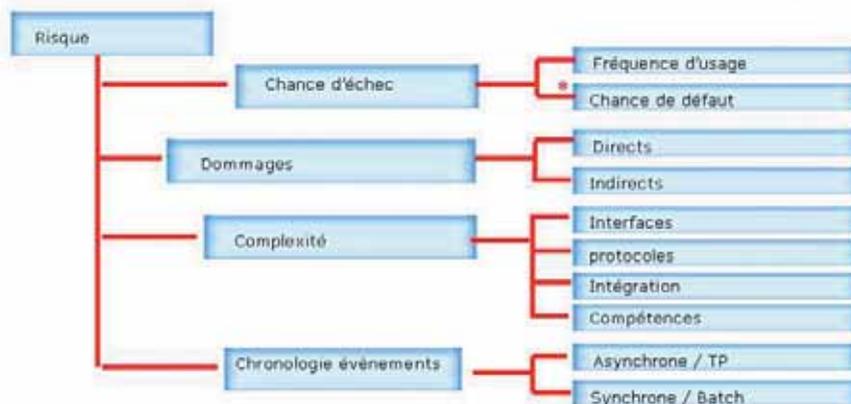


Suite du Dossier TESTS :
prochain numéro Juillet/Août

- Ruby et les tests
- Mettre en œuvre PHPUnit

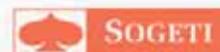


Risques: facteurs, échelles de mesures et pondérations



Module	Compl.	Fréq.	Taille	UTIL.	Réutil.	R/O/L	Risque	Risque relatif
Com-mande	1.00	1.00	380	3.00	3.00	3.00	3180	40%
Facture	0.87	0.83	480	2.00	0.00	3.00	1281	16%
Bases de données	1.00	1.20	100	1.00	0.00	2.00	360	5%
Blocks	1.80	0.83	200	3.00	0.00	3.00	1348	18%
Acquisition	1.00	1.20	178	3.00	3.00	3.00	1000	14%
							7888	100%

$$\text{Risque} = (\text{Complexité} * \text{Fréquence} * \text{Taille}) * (\text{Importance utilisateur} + \text{Réutilisation} + \text{Batch/On-line})$$



Le choix de ces techniques d'évaluation requiert de l'expérience. Le choix se fait selon plusieurs techniques d'évaluation basées sur des ratios, la taille de l'objet du test etc.

- La création de variables métriques : la définition, gestion et utilisation de variables métriques sont essentielles lors du processus de test car il permet au chargé de test une réponse assortie de fait aux questions portant sur la qualité de l'objet du test et la progression du processus de test.

- Une analyse du risque produit : elle analyse le produit à tester avec l'objectif d'atteindre une vue partagée par le chargé de test et les autres parties prenantes. Elle permet de répondre à différentes questions : quelle est le risque encouru par l'organisation si le produit n'atteint pas la qualité attendue ?

En conclusion, la notion de " types de tests " est fondamentale dans la maîtrise de la couverture et de la profondeur des tests. Leur répartition entre les phases de tests, les acteurs et les environnements est une condition majeure d'efficacité et de productivité des

tests. Les normalisations et les techniques comme celles de TMap® (Test Management Approach)* sont conçues pour permettre à des professionnels spécialisés la production d'un Testware** de qualité.

Conduire un processus de test structuré est le meilleur moyen de maintenir une qualité maximale en optimisant les délais de développement des logiciels. Cette formalisation permet aux entreprises une réelle gestion des risques et un gain de temps sur l'élaboration du logiciel en découvrant au plus tôt les anomalies les plus importantes. Elles misent sur la capitalisation de leurs investissements lors de la création du processus en sauvegardant, pour des utilisations ultérieures, certaines phases du projet. La structure de ces normes repose sur quatre questions : **quoi** et **quand?** **comment?** **où?** **qui?** Celles-ci correspondent elles-mêmes à quatre thèmes: **cycle de vie**, **organisation**, **infrastructures** et **outils** et enfin **techniques**. Le cycle de vie des tests détaille chaque étape du projet. L'organisation concerne autant la structure de l'équipe de test que

sa place au sein du projet. L'infrastructure définit la stabilité de l'environnement et les outils à employer. Aux techniques applicables lors des tests, elles permettent de structurer et de reproduire les tests ultérieurement et concernent la gestion des plannings, la définition des méthodes de revue de tests, la spécification des cas de tests et le reporting. Applicable à tous types d'entreprises, le cycle de vie de ces normes (planification et suivi, préparation, spécification, exécution et clôture) est néanmoins suffisamment flexible pour s'adapter aux besoins spécifiques de chaque client. Un bon usage des types de tests est l'un des éléments qui permet de passer du test au testing. Le testing étant le meilleur moyen de maintenir une qualité maximale en optimisant le délai de développement des logiciels (et accessoirement de maintenance des tests dans le temps, ndlr).

■ François Darphin

Directeur Technique en charge du Testing chez Sogeti

* TMap® : Cette méthodologie, créée et enrichie continuellement par Sogeti, permet une consolidation des gains en termes de réduction des coûts et d'amélioration de la qualité du patrimoine logiciel et technique. Mise à disposition du centre de service de tests, elle garantit une démarche de gestion structurée des tests logiciels avec un " pilotage des tests orienté métier " et la qualité des projets grâce à une boîte à outils complète.

** Testware : Ensemble des données et documents de tests

Les tests unitaires et PHP



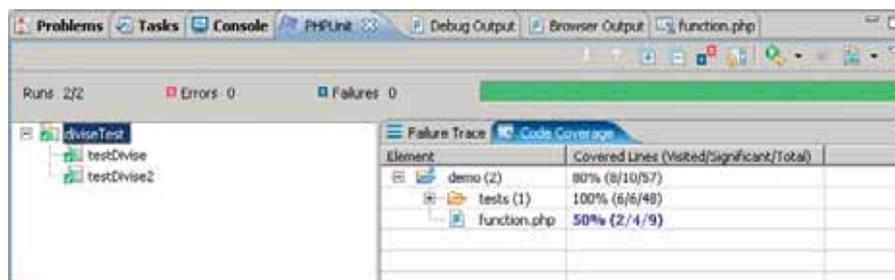
1^{re} partie

La notion même de test unitaire est très longtemps restée obscure pour les développeurs PHP. Et avouons-le, elle le demeure encore pour beaucoup d'entre eux. Cela s'explique probablement par le fait que l'écriture d'applications critiques en PHP est encore assez récente, et que c'est à l'origine principalement pour ce type de projets que les tests unitaires ont été pensés. Après plusieurs années de maturation, la communauté PHP s'approprie désormais les concepts et bonnes pratiques éprouvées de l'ingénierie logicielle traditionnelle.

Les enjeux

L'intérêt soudain que porte la communauté PHP à cette pratique tient essentiellement à deux facteurs : premièrement, le nombre d'applications majeures (y compris des sites constituant le cœur du modèle économique de certaines entreprises) a augmenté considérablement ces dernières années, et d'autre part l'apparition des méthodes de développement agiles, comme le célèbre eXtreme Programming (XP), si cher désormais aux développeurs web. Le premier phénomène a entraîné l'adoption de pratiques jusqu'alors relativement absentes du champ d'intérêt des développeurs PHP, qui préféraient la souplesse offerte par le langage à la rigueur (pour ne pas dire la rigidité) qu'imposaient les méthodes inhérentes aux autres plates-formes. Mais pour se voir attribuer la responsabilité d'applications d'envergure, PHP et son écosystème ont dû " rentrer dans le rang ". Il n'était plus question de prôner la souplesse en tant qu'argument majeur de la plate-forme PHP. Celle-ci devait être assortie des mêmes garanties de robustesse et de fiabilité qu'offraient les plates-formes concurrentes, pour lui conserver tous ses avantages en termes de rapidité de développement (time to market*) sans compromettre la qualité de l'application produite.

La méthode de développement XP a, quant à elle, induit une nouvelle problématique dans le processus de développement. Par définition, cette méthode de gestion de projet amène le développeur à régulièrement revenir sur un code précédemment validé, ou tout au moins sur du code très proche d'éléments validés. Si cette approche dite " par itération " permet une prise en charge souvent plus pragmatique, et donc plus rapide, des aspects fonctionnels de l'application, il apparaît un risque important de ne plus être en mesure de s'assurer de la cohérence de l'ensemble. A la façon d'un jeu de dominos-cascade, une modification sur un



bout de code peut entraîner des répercussions sur de nombreuses autres parties de l'application : on parle alors de régression.

Les tests unitaires ont été conçus pour faire face à ce type de problématiques. En imposant de maintenir opérationnelles des unités de code indépendamment les unes des autres, le recours aux tests unitaires ajoute aux langages impératifs une notion proche du paradigme des langages fonctionnels, garantissant la cohérence et la non-régression des éléments du programme, et ce conformément aux charges définies lors de conception de l'application. Avec l'apparition de la méthode de développement pilotée par les tests (test driven programming ou TDD), l'exploitation des tests unitaires a été poussée à son extrémité (jeu de mot facile, la méthode de test driven programming ayant été conçue par Kent Beck, également co-inventeur de la méthode XP). Le concept de cette méthode est très simple : il s'agit de définir l'application d'abord en écrivant le jeu de tests qu'elle devra respecter, et de n'en produire le code qu'ensuite. Le développement ne prend fin que lorsque tous les jeux de tests sont passés avec succès par le code. Avant d'étendre les fonctionnalités de l'application, on procédera d'abord à une nouvelle phase d'écriture de tests, et cætera. En appliquant cette manière de faire, on se rend compte combien les tests unitaires représentent les rails, et surtout les aiguillages, qui seront nécessaires à l'application pour que

l'ensemble de ses constituants puissent fonctionner de concert, en bonne intelligence, et prévenir tout ... déraillement. Si cette méthode peut, dans la pratique, sembler un peu jusqu'au-boutiste, elle est en revanche la parfaite illustration de l'idée qu'il faut se faire des tests unitaires.

* time to market = temps écoulé entre l'identification d'un besoin et sa réalisation, puis sa distribution

A quel niveau utiliser les tests unitaires ?

Comme nous venons de le voir, des méthodes proposent de commencer par écrire les tests avant l'application elle-même. Sachez qu'il est possible en faisant ainsi d'aller très loin. La plupart des frameworks de tests unitaires permettant la génération d'objets factices (appelés *mock objects*) à l'intérieur des scripts de tests pour simuler le comportement attendu par un composant quelconque de l'application (lui-même étant de fait testé unitairement par ailleurs, cela va de soi), afin notamment de ne pas compromettre un test à cause d'erreurs pouvant survenir dans un autre composant que celui auquel se rapporte le test en cours. L'utilisation abondante de ces *mocks objects* dans les tests permet de tester grandeur nature, ou presque, la corrélation entre les composants de son application avant même qu'elle n'ait été écrite. Conclusion, il est possible de valider l'architecture de son application sans

avoir écrit la moindre ligne de code fonctionnel. Les tests unitaires auront alors contribué à améliorer la structure même de l'application. Cependant, cette possibilité n'est pas celle la plus souvent mise en œuvre. Dans bien des cas, l'architecture de l'application est décidée et/ou validée par d'autres moyens, en amont (modélisation, conformation à un framework, etc.). Les tests unitaires sont donc le plus souvent implémentés au fur et à mesure de la conception des composants, dont l'unité la plus naturelle est la classe en programmation orientée objet. Ainsi, chaque définition de classe au sein d'un projet devrait se voir assortie de la création d'un test unitaire correspondant. D'un point de vue pratique, dans ce type d'utilisation, il convient d'automatiser l'exécution des jeux de tests au minimum lors de la réversion de code dans un dépôt commun dans le cadre de l'utilisation d'outils de gestion de versions comme SVN.

Le fonctionnement

Un framework de test unitaire est composé, outre son runner (le code permettant l'exécution des tests eux-mêmes), de diverses entités dont trois sont indispensables à maîtriser :

– le Test Case

Il s'agit du composant principal d'un framework de test unitaire (du point de vue de l'utilisateur), portant sur une fonction ou une classe. Concrètement, il s'agit d'une classe dérivant d'un modèle de Test Case, fourni par le framework, et qui comporte un certain nombre de méthodes nécessaires à l'exécution des tests, parmi lesquelles setUp()^{**}, tearDown()^{**} et chaque test proprement dit, matérialisé par une méthode préfixée de la chaîne test^{**}. Autrement dit, un test case peut comporter autant de tests qu'on le souhaite, avec au minimum un test par méthode de la classe testée. Les méthodes spéciales setUp() et tearDown() permettent respectivement de mettre en place puis nettoyer l'environnement de test entre l'exécution de chacun des tests. Cet environnement est constitué de fixtures dans le jargon des tests unitaires, qui sont des données figées, repères à l'aune desquels on pourra déterminer si les tests sont validés ou non (il peut s'agir de l'établissement d'une connexion à une base de données, par exemple).

– la Test Suite

Ensemble de Test Cases permettant de les

tester tous simultanément, établissant ainsi des ensembles de type scénario.

– La couverture de code

Cette troisième notion est indissociable du concept de tests unitaires, bien qu'il s'agisse le plus souvent d'un outil tiers, ou du moins n'intervenant pas dans le processus de test lui-même. Appelé "code coverage" en anglais, il s'agit d'un outil de mesure quantitatif permettant d'émettre un avis qualitatif sur les tests écrits pour une application. Dans sa version la plus simple, il va permettre de déterminer la quantité d'instructions réellement exécutées lors de l'exécution des tests unitaires. Ce faisant, on va pouvoir évaluer si oui ou non les tests couvrent une partie représentative de l'application. Dans une moindre mesure, le fait qu'une partie du code n'est jamais exécutée alors que l'on est sûr de la pertinence de ses jeux de tests peut permettre de juger de l'obsolescence de certaines portions de code.

^{**} Bien que n'étant pas universelle, cette nomenclature s'applique aux deux frameworks de test unitaire les plus utilisés en PHP : SimpleTest et PHPUnit (voir références).

Conclusion

Le sujet des tests unitaires est vaste, et souvent abstrait pour les développeurs qui ne sont pas habitués aux méthodes très "théorisées" de développement. Pourtant, après une brève période d'adaptation, il devient évident que leur emploi systématique permet d'accélérer les développements, en évitant de nombreux bugs, et en facilitant largement la résolution de ceux qui surviennent cependant. Pour rendre leur intérêt et leur emploi plus tangible, nous consacrerons le mois prochain deux autres articles sur le sujet, l'un sur le framework PHPUnit, et l'autre sur l'intégration de ce même framework au sein de Zend Studio 6.0 (Zend Studio for Eclipse).



Les liens

SimpleTest => <http://www.simpletest.org/fr>
PHPUnit => <http://www.phpunit.de>

■ Gauthier Delamarre

NOUVEAU

Une nouvelle
rubrique
a démarré sur
www.programmez.com

les **TUTORIELS** !

Les pas à pas
pour vous guider

www.programmez.com/tutoriels.php

Profession : testeur de logiciels

Nous avons tous des exigences de qualité pour les logiciels que nous utilisons. Elle est nécessaire, que ce soit pour des applications critiques (monde spatial, aéronautique, médical, ...) ou de grande taille (banques, assurance, ...), mais aussi dans le cas de logiciels embarqués (ex. automobile, télécom) ou d'applications grand public / entreprise (jeux, logiciels de gestion, ...). Chacun des membres des équipes de développement, du concepteur au chef de projet, de l'architecte à la maîtrise d'œuvre, est impliqué dans la qualité des applications fournies.

Les anomalies dans les logiciels devraient être supprimées avant leur découverte par les utilisateurs. Les testeurs cherchent, préalablement à la livraison, les " bugs " de façon à permettre leur suppression. Ainsi, les testeurs participent, au même titre que les développeurs, à l'amélioration de la qualité des logiciels. Ils font partie du cycle de vie du développement et de l'application.

Problématique

Les tests ne se limitent pas à des aspects d'exécution (automatisée ou non) et de vérification de bon fonctionnement. Vu qu'ils sont sur le chemin critique de la mise en production, la planification des activités de tests, leur efficacité et leur rentabilité doivent être assurées. Les testeurs contribuent donc, à des niveaux de responsabilité importants, à la production d'applications logicielles de qualité. Pour assurer la qualité des logiciels développés, il est important que les testeurs possèdent un niveau de connaissance adéquat.

Les formations supérieures en informatique ne manquent pas et, malgré l'importance grandissante des tests, leur part dans le cursus est plus que limitée. Les développeurs valident leurs connaissances par un diplôme, par contre aucune formation supérieure pour testeurs de logiciels n'existe en France.

Comment un employeur ou un client potentiel peut-il s'assurer du niveau de connaissances de ses testeurs sachant que les tests représentent plus de 30% de la charge des projets ?

Techniques

Depuis le début des années 1980, le ratio des tests dans les projets s'accroît, en raison de l'augmentation de la complexité et des interactions des applications entre elles.

Les tests évoluent avec les techniques de modélisation et de conception. Les techniques

de tests actuelles couvrent tous les niveaux de développement (tests unitaires ou de composants, tests d'intégration, tests système et tests d'acceptation), s'adaptent aux méthodes de conception utilisées (cycle en V, techniques agiles, UML, eXtreme Programming) et aux techniques actuelles (Java, Ajax, Eclipse, etc.). Parmi les techniques couramment utilisées nous avons :

- les partitions d'équivalence et leur extension aux tests aux limites,
- les oracles de tests et la définition des données attendues préalablement à l'exécution des tests,
- les tables de décision et cas d'utilisation,
- les transitions d'états et flux de données,
- les techniques statiques (sans exécution du logiciel) et dynamiques.

Les techniques évoluant, il est nécessaire d'avoir un organisme indépendant qui tienne à jour un référentiel cohérent, et puisse évaluer – par le biais d'un examen – le niveau de connaissance des testeurs.

Solution

Au regard de ce constat, des spécialistes en tests de logiciels de plusieurs pays européens se sont regroupés et ont mis en place un schéma de certification (ISTQB) dont l'objectif est de fournir une évaluation indépendante et reconnue des testeurs de logiciels. Depuis 2002, cette association a mis en place des cahiers des charges de cours (appelés syllabus) définissant le niveau de connaissance minimal attendu de testeurs débutants (syllabus niveau Fondation) ou expérimentés (syllabus niveau Avancé). Parallèlement, des examens ont été mis en place pour valider l'acquisition du niveau des connaissances. Des sociétés de formations peuvent développer des formations qui respectent ces cahiers des charges. Le CFTL (Comité Français des Tests Logiciels) représente la France dans ce sché-

ma de certification. La pertinence de cette approche, incluant une reconnaissance croisée des certifications, est validée par le nombre important de testeurs certifiés (plus de 80.000. à ce jour dans le monde, près de 200 en France) et par les sociétés exigeant une certification CFTL-ISTQB à l'heure actuelle.

Un parcours professionnel

Conscients du besoin de professionnalisation de la profession, le SYNTEC et plusieurs grands comptes mettent en œuvre, pour les testeurs de logiciels, un parcours basé sur les métiers définis dans les syllabus CFTL-ISTQB : Testeur (niveau Fondation) puis Analyste de test (focalisation sur les aspects fonctionnels) ou Analyste Technique (focalisation sur l'utilisation de techniques ou outils particuliers) et enfin Test Manager (gestion des campagnes de tests). Ce parcours permet d'offrir des perspectives de carrières intéressantes aux testeurs. Le FAFIEC a mis en place pour l'année 2008 une action collective de formation en tests de logiciels. Un organisme de formation accrédité par le CFTL a été retenu dans le cadre de cette action. Ses formations, ouvertes à toutes les SSII et prises en charge à 100% par le FAFIEC, comprennent les examens de certification CFTL niveau Fondation et niveau Avancé.

Cursus

Le premier niveau de connaissances (Fondation) comprend des notions utiles aussi aux développeurs, chefs de projets et AMOE/AMOA pour déterminer ce qu'ils sont en droit d'attendre de la part de leurs équipes de tests. Il couvre :

1. Les fondamentaux du test : nécessité, principes généraux et processus de tests, la psychologie des tests
2. Tests et cycle de vie logiciel : modèles de développement, niveaux de tests, types de tests, tests et maintenance.

3. Techniques statiques : processus de test, processus de revue, analyse statique outillée.
4. Techniques de conception de tests : conditions de test, techniques de conception de tests, techniques basées sur les spécifications, sur la structure ou sur l'expérience, sélection des techniques appropriées.
5. Gestion des tests : organisation et indépendance des tests, tâches du responsable des tests et des testeurs, estimation, planification et suivi de l'avancement des tests, reporting et contrôle des tests. Gestion de configuration, tests liés aux risques, gestion des incidents.
6. Outils de support des tests : outils de gestion des tests, outils d'aide aux tests statiques; outils d'aide à la spécification, à

l'exécution et au suivi des tests, outils de support des performances et de surveillance; rentabilité des outils de tests, bénéfices potentiels et risques, méthodes d'introduction d'outil dans une organisation.

Le niveau Avancé étend les acquis du niveau Fondation en y ajoutant, entre autres, les aspects de mesures et de métriques, d'amélioration continue des processus dans le cadre de SPICE et CMMI, et l'impact spécifique des applications critiques et des systèmes de systèmes. Les tests impliquent, outre une connaissance métier, la connaissance de techniques spécifiques et de méthodes complexes qu'il n'est pas possible de résumer correctement dans une lettre de motivation. L'obtention d'une certification CFTL-ISTQB permet de valider l'acquisition de ces connaissances, et ainsi rassurer un client ou un employeur potentiel.

connaissance en développement de logiciels est un atout non négligeable pour les testeurs de logiciels : d'une part cela facilite la compréhension des besoins, d'autre part l'utilisation d'outils automatisés requiert des connaissances en programmation.

Les tests de logiciels mettent en œuvre des connaissances plus étendues que les seules compétences de développeurs : il est nécessaire d'avoir un bon niveau de synthèse et de communication, une compréhension rapide des aspects matériels, logiciels, organisationnels et humains impliqués dans le développement des applications, de façon à améliorer la qualité sans impacter les coûts ou les délais. C'est un challenge de taille que certains développeurs peuvent être en mesure de relever.

Conclusion

L'amélioration de la rentabilité des développements des logiciels passe par une amélioration de la qualité des produits logiciels et des processus. Les testeurs de logiciels contribuent à ces améliorations de par leurs activités et leur expérience. La professionnalisation du métier de testeur de logiciel est en marche, par la réalisation de filières professionnelles et de certifications internationales reconnues. La valorisation de l'expérience des testeurs et la mise en place de filières professionnelles prend de l'ampleur en France et dans le monde, entre autres, par le biais de certifications internationales reconnues, ouvrant la voie à une réelle internationalisation des ressources.

Quelques chiffres

Selon une étude récente du cabinet Pierre Audoin Conseil, le marché français des tests de logiciels était de 380 millions d'euros en 2006, 424 millions d'euros en 2007 et devrait subir une croissance annuelle de 12,4% jusqu'en 2010. Ces chiffres sont certainement en dessous de la réalité car ne prenant pas en compte les tests effectués au sein des sociétés utilisatrices. Une telle croissance est une opportunité pour les testeurs dont les fourchettes de salaire vont pour les débutants (selon le niveau d'études) de 20 à 30 K euros, pour les testeurs confirmés de 30 à 37 K euros, pour les Test Manager de 40 à 60K euros, voire plus de 80 K euros pour des directeurs de projets de tests.

Internationalisation

L'internationalisation du marché (tant européen que mondial) et le recours à la sous-traitance et à l'offshore nécessite un mode d'évaluation qui soit réellement international. Les certifications définies ci-dessus sont reconnues actuellement comme équivalentes à celles fournies dans plus de 36 pays, de tous les continents. Ainsi, un testeur ayant obtenu une certification CFTL-ISTQB en France pourra sans problème la valoriser en Angleterre ou en Allemagne. Dans ces pays, comme dans d'autres, une certification ISTQB est une condition sine qua non à un emploi de testeur.

Un challenge pour les développeurs

Une bonne intégration des testeurs et des développeurs permet de garantir un bon échange d'informations et un niveau de qualité accru. L'expérience a démontré qu'une



■ **Bernard Homès** ISTQB CTAL
Président du Comité Français
des Tests Logiciels
www.cftl.fr

Programmez!
LE MAGAZINE DU DEVELOPPEMENT

Prochain numéro
Hors Série .Net :
14 juin 2008

• Visual Basic 9 :

Les meilleurs trucs et astuces pour maîtriser votre langage préféré et exploiter au mieux VB dans Visual Studio 2008.

• Silverlight :

Développer et déployer vos applications Silverlight. Les outils, les bonnes pratiques.

Outils de tests : panorama du marché

Il est difficile de réaliser un panorama des outils de tests. Il en existe plusieurs centaines. Cependant, quelques dizaines, open source et commerciaux, se distinguent du lot. Nous allons les passer en revue.



Les outils open source

OpenSTA : outil d'injection de charge http / HTTPS, idéal pour tester les performances

PHPUnit : framework de tests unitaires dédié à PHP

JUnit : framework de tests unitaires dédié à Java. Une déclinaison Java EE existe : J2EJUnit, plus connu désormais sous le nom de Cactus. Il existe aussi ServletUnit pour les servlets

Nunit : framework de tests unitaire dédié à .Net

Httpunit : framework de tests unitaires pour tester l'authentification http, la connexion. On peut le combiner à Junit.

C Unit Testing Framework : framework de tests unitaires pour les codes C

LogiTests : suite d'outils pour test des sites et applications web. Il comprend LogiTest pour enregistrer et rejouer des tests. Y compris les

tests fonctionnels et de régression. Il inclut aussi LogiMonitor.

Jmeter : outil de tests de charge pour application Java, originellement conçu pour le web. Il teste les ressources dynamiques et statiques.

Clif : outil bien connu, il s'agit d'un framework de tests de charge pour Java.

Pour en savoir plus sur les outils de tests open source : <http://www.opensourcetesting.org/>

Les outils commerciaux

Editeur	L'offre
Parasoft	Jtest permet d'automatiser les tests unitaires Java et génère automatiquement les tests Junit. Permet aussi de vérifier la validité du code en incluant 500 règles de développement pour détecter les violations de codage. C++ Test et .TEST sont les pendants de Jtest en C++ et .Net. Pour les tests de performances et de charge, on dispose de WebKing pour les applications web. Pour le service web et les architectures services, l'éditeur propose SOA Test.
Verifysoft	L'éditeur propose Conformiq Qtronic, un générateur automatique de cas de test. CTC++ et CTC++ for Java et C# pour couverture de test pour C, C++ et Java et C#. Il existe aussi CMT++ et CMTJava, pour la mesure de complexité pour C/C++ et Java.
Coverity	Editeur spécialisé dans l'embarqué, on dispose de Coverity Present pour vérifier et détecter les défauts dans son code (Windows, COM). Depuis peu, l'éditeur propose aussi Coverity SQS pour automatiser l'identification et la résolution des défauts en C, C++ et Java. Enfin, toujours pour Java, Thread Analyzer for Java.
Webperformance	Web Performance Suite permet de mettre en œuvre rapidement des tests de montée en charge et de performance pour les applications web. Il comprend Analyser pour mesurer, analyser, suivre les performances de pages web et Load Tester pour les capacités de charge de ses pages web.
Quest Software	Jprobe permet de tester facilement et rapidement son projet Java afin d'en améliorer la qualité. Il s'utilise aussi bien avec un serveur d'applications qu'avec un IDE.
IBM Rational	<p>Functional Tester : Outil de tests fonctionnels. Java et .Net. Existe aussi l'édition Plus.</p> <p>Manual Tester : outil de création et d'exécution de tests manuels qui permet de réutiliser des phases de test, réduisant ainsi l'impact des modifications logicielles pour les testeurs et les analystes métier. Résout les problèmes liés aux tests manuels, souvent fastidieux et sources d'erreurs, surtout si l'application subit de nombreuses modifications.</p> <p>PurifyPlus : Outil de qualité du code. Permet de détecter les problèmes de codes, les fuites mémoires, etc. Disponible pour Windows, Linux, Unix.</p> <p>Robot : Outil d'automatisation des fonctions de tests des applications client-serveur en environnement .NET, Java et Web.</p> <p>Test RealTime : Outil pour tester les composants et analyser les logiciels multi-plates-formes en temps réel. Pour code C, C++, Java, Ada.</p> <p>Performance Tester : Outil de test de charge et de performance.</p>
Quality First Software	Qf-test est un outil pour tester (automatiquement) les interfaces Java. Il fonctionne à base de scénarios de tests.
HP	HP propose une gamme de tests et de qualité étendue. L'un des plus connus est LoadRunner, un outil de performance et de charge. TestDirector est un outil de gestion des tests (planning, exécution, gestion des défauts, etc.).
Compuware	L'éditeur propose plusieurs solutions de tests. On citera tout particulièrement TestPartner permettant de construire les tests fonctionnels à différents niveaux (développeurs, experts, utilisateurs). Test Management vise plutôt la gestion des tests, la planification, l'exécution ; idéal pour les équipes de qualité et les cellules de tests. La gamme DevPartner permet aussi des tests, de faire de la couverture de code, d'analyser les ressources, les threads, etc. La gamme couvre les principaux langages.
Leirios	L'environnement Smart Testing permet d'implémenter le test fondé sur les exigences métiers et d'automatiser la conception du patrimoine de tests (cas de test, scripts exécutables, couverture du plan de test) en garantissant la qualité de la couverture fonctionnelle. Cette solution opérationnelle s'inscrit dans une démarche de mise en place d'usine à tests permettant d'améliorer la conformité, la traçabilité et la gestion du risque.

Quelques autres éditeurs :

Borland : SilkCentral Test Manager, Gauntlet, Silk Test et Silk Performer - **Empirix** : e-TEST suite, OneSight, e-LoadExpert

Microsoft : Web Application Stress, Visual Studio Team System Software Tester

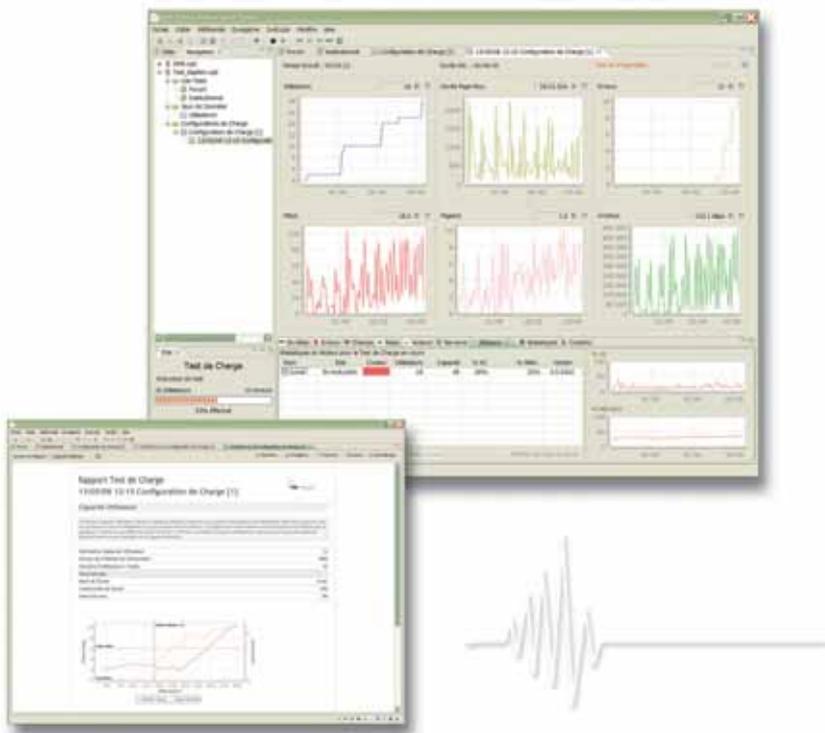
■ François Tonic

Suite du Dossier TESTS :
prochain numéro Juillet/Août

- Ruby et les tests
- Mettre en œuvre PHPUnit



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



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

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

Evaluation gratuite de Web Performance Suite sur <http://www.kapitec.com/Pub/WP?id=109>

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



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

Power Your Web Projects

XML sans douleur... ou presque

Dans le numéro précédent de votre magazine préféré (n°108), nous avons entamé le dossier XML. Nous avons évoqué les fondamentaux, les bonnes pratiques, avec quelques mises en œuvre d'outils. Vous l'aurez compris, XML est aujourd'hui un élément inséparable des développeurs et même, sans qu'ils le sachent, des utilisateurs. De plus en plus de logiciels supportent, consomment et communiquent en XML. Ce qui ne va pas toujours sans problème, notamment sur le XHTML comme on le verra dans les pages qui suivent. Dans cette 2e partie, nous avons privilégié l'aspect pratique avec la validation par schéma (après avoir abordé celle par DTD) et le XSLT. Le XSLT est parfois négligé mais à y regarder de plus près, il ouvre une souplesse d'utilisation et de manipulation sans pareille ! Nous parlerons aussi d'évolution du XML avec une approche très réaliste et terrain de XML 2. Pour cela, un des "spécificateurs" de XML 1.0, Michael Sperberg-McQueen, nous dévoilera son approche. Mais comme XML le vaut bien, nous terminerons notre plongée au cœur d'XML le mois prochain, dans une 3e partie encore plus orientée code et mise en œuvre. Bonne validation.

■ François Tonic



Comparaison et fusion de fichiers XML

XML s'est tranquillement imposé comme un citoyen de première classe dans la panoplie du développeur : les fichiers de paramétrage et de configuration sont désormais nombreux à utiliser ce mode de sérialisation et de nombreux formats "métier" sont sans cesse créés pour stocker des informations et les réutiliser facilement.

Une des difficultés que peut rencontrer le développeur est alors liée à la manipulation de ces fichiers au sein de son environne-

ment et, bien que le format soit textuel, son côté verbeux rend délicate la comparaison, voire la fusion de différentes versions de fichiers.

Eclipse, grâce au projet modeling et en particulier EMF Compare, répond à ce besoin de manipulation des informations du fichier XML. Il est en effet possible, moyennant la description XSD d'un format XML, de charger ce fichier en tant que modèle où l'on ne manipule alors plus l'arbre XML, mais bien les entités représentées. Toute la panoplie

d'outils fournis par Eclipse, dont EMF Compare, peut alors exploiter ces informations et la comparaison n'est plus syntaxique mais sémantique. Il en découle une meilleure maîtrise de l'évolution des données et des paramétrages et moins de risque d'erreurs avec un fichier XML fusionné bien formé et des références correctes.

Pour plus d'informations, consultez le tutoriel sur le wiki Eclipse : http://wiki.eclipse.org/Comparing_XML_files_with_EMF_Compare



XML 2.0 : le futur du XML est-il en marche ?

Il existe de bonnes raisons pour envisager une révision majeure du XML, et d'autres, tout aussi plausibles, pour écarter cette hypothèse. Personnellement, je ne saurais dire lesquelles seront retenues en fin de compte, et je pencherais en faveur d'une approche conservatrice.

Toutefois, je suis conscient que ce conservatisme peut refléter un parti-Pris, car je suis l'un des auteurs et éditeurs de la première spécification XML.

Toute spécification, dont l'utilisation est effective, suscite des comptes-rendus sur les erreurs ou problèmes qu'elle présente. L'expérience montre que telle ou telle décision de conception ne donne pas forcément les résultats attendus, le résultat étant parfois meilleur mais hélas souvent pire. Force est de constater que les seules spécifications qui n'ont pas à être modifiées ni reconsidérées sont celles qui ne sont plus utilisées. L'usage largement répandu du XML, tant dans les documents en langage naturel que dans les messages de données et inter programmes, a donné lieu à des demandes répétées pour une approche plus condensée. (Le travail du W3C concernant "l'efficacité des échanges XML" du Groupe de travail EXI pourrait contribuer à répondre à ce besoin mais dès lors qu'il touche aux questions fondamentales de conception XML, il est inéluctablement sujet à controverse.)

Des contraintes de syntaxe ?

Il n'en reste pas moins que certains auteurs travaillant sans éditeurs compatibles XML considèrent les règles détaillées spécifiques de la syntaxe XML comme une véritable contrainte. Pourquoi toutes les valeurs d'attributs doivent-elles figurer entre guillemets ? Ne serait-il pas possible d'omettre ces guillemets quand la valeur contient uniquement des caractères alphanumériques ? Pourquoi tous les éléments doivent-ils être systématiquement encadrés ? À l'évidence, il est assez facile de voir que quand un nouvel élément de liste commence, le précédent doit être terminé. De même pour les paragraphes.

À cet égard, l'expérience du XHTML et du HTML 4 (et 5) est tout à fait à propos. Le XHTML est plus facile à analyser et à traiter que le HTML non-XML, c'est-à-dire qu'il peut être lu par un plus grand nombre d'outils. Or

cette simplicité supplémentaire n'offre que peu d'intérêt aux fabricants de navigateurs qui se sont déjà engagés à prendre en charge le HTML existant. En outre, pour d'obscures raisons, quelques navigateurs existants ne parviennent pas toujours, voire pas du tout, à manipuler utilement du XHTML.

Les auteurs qui se soucient le plus de pouvoir utiliser un large éventail d'outils n'écrivent généralement pas leurs documents en HTML directement (mais passent plus volontiers par des vocabulaires XML tels que DocBook, TEI, etc.), alors que ceux qui rédigent en HTML ne connaissent pas souvent les avantages que confère le contenu réutilisable à dessein. Si bien que le XHTML n'a pas suscité l'intérêt escompté et que l'on envisage sérieusement de reprendre le travail sur des versions non-XML du HTML, dont HTML 5, dans le but de mettre à disposition des vocabulaires XML importants tels que MathML dans des formats non-XML.

Tous ces facteurs ont motivé le souhait de certains de voir apparaître la spécification XML 2.0.

Cela dit, quand l'utilisation des spécifications se généralise, il peut être difficile de les modifier, en particulier si elles sont intégrées dans des couches relativement basses de l'infrastructure. Il serait, aujourd'hui difficile de modifier le XML pour les mêmes raisons qui empêchent le changement de nos jeux de caractères fondamentaux. Toute approche XML 2.0 s'accompagnerait inévitablement de défis d'interopérabilité avec XML 1.0, d'où la difficulté pour le XML 2.0 de remporter l'adhésion de la communauté des utilisateurs. (L'adoption lente du XML 1.1, qui n'apporte que très peu de modifications à XML 1.0, est un exemple qui donne à réfléchir : s'il est vrai que la grande majorité des documents XML

1.1 peut être parfaitement gérée par les analyseurs syntaxiques XML 1.0 et que les modifications d'analyseur requises pour supporter XML 1.1 sont très modestes, cette spécification est actuellement très peu adoptée. Aussi le XML 2.0 devra-t-il être très intelligent pour conserver l'interopérabilité avec les outils XML 1.0 et offrir suffisamment de nouvelles fonctionnalités pour valoir la peine d'une montée de version. La plupart des partisans de l'idée du XML 2.0 sont ceux qui portent un intérêt à un domaine d'application plutôt restreint où les modifications de règles particulières du XML pourraient avoir des avantages visibles et importants. Si le XML est exclusivement utilisé pour la communication inter-programmes, alors les fonctions XML qui sous-tendent la création des documents en langage naturel semblent manifestement inutiles. À l'inverse, si le XML sert uniquement à intégrer du MathML dans des pages HTML (non XHTML), les règles de séparation du XML seront de fait vaines et pesantes.

Un des aspects les plus utiles du XML est qu'il peut être exploité pour toute forme d'informations, précisément parce qu'il sert à la fois le propos de la communication entre les programmes et des documents lisibles par l'homme. Toutefois, cette généralité à un prix : dans un domaine d'application particulier, le XML peut être moins pratique qu'un format de données conçu et optimisé aux seules fins de ce domaine. Dans la mesure où les données et les programmeurs se cantonnent à des domaines spécialisés, les formats de données spécialisés sont susceptibles de l'emporter. En revanche, si les flux d'informations transcendent les frontières pour être réutilisés dans des contextes très diversifiés, les avantages que recèle un format générique tel que le XML dépasseront largement ceux plus traditionnels des formats spécialisés. Le temps nous dira

quelles forces l'emporteront dans la technologie de l'information en général mais le W3C repose sur le principe de l'augmentation du flux d'informations et la possibilité de réutiliser et de publier ces données sur une large échelle. De par sa nature institutionnelle, le W3C ne peut pas se prononcer à la légère sur la question du XML 2.0. Nous devons prendre une décision qui serve au mieux les intérêts de la communauté Web dans son ensemble et comme cette communauté est très étendue et a des objectifs souvent antagoniques, le choix est cornélien.



Photo: Liam Quinn

■ **Michael Sperberg-McQueen**
co-éditeur de la spécification
XML 1.0

L'information du DÉCIDEUR



Choisir, déployer,
exploiter
les logiciels en
entreprise

www.solutions-logiciels.com

INTERVIEW



“ Il est dangereux d'optimiser XML car cela se fait en général au détriment de l'interopérabilité ”

Philippe Le Hégaré, responsable des activités liées à l'architecture Web du W3C et co-éditeur des spécifications DOM 2 et DOM 3, répond à nos questions

Programmez : depuis quelques années XML 2.0

fait parler de lui. Où en est-on aujourd'hui ?

Philippe Le Hégaré : Les discussions continuent, sans qu'il y ait de conclusion. Chaque développeur a tendance à suggérer ses propositions d'améliorations pour XML, qui sont en général applicables dans un domaine très particulier. XML a achevé un haut niveau général d'interopérabilité pour l'échange de données dans un grand ensemble de domaines et il sera difficile de le détrôner de ce rôle, y compris avec un XML 2.0.

P ! : XML est partout, sous une forme ou sous une autre. Il demeure verbeux. Quelles sont les bonnes pratiques que le développeur peut mettre en oeuvre pour optimiser, faciliter sa manipulation ?

PLH : Il est dangereux d'optimiser XML car cela se fait en général au détriment de l'interopérabilité. Certaines communautés ont tout de même réussi à introduire un nouveau langage avec succès mais, encore une fois, en réduisant le domaine d'application. Si la verbosité d'XML pose des problèmes du point de vue de la bande passante, alors dans certains cas, une représentation plus compacte de l'information sera utile. Le W3C travaille actuellement sur une représentation alternative d'XML : "Efficient XML Interchange" (EXI). EXI, un nouveau format binaire, permettra de représenter des données XML de manière plus compacte. Son utilisation ne sera néanmoins pas recommandée pour un usage général, mais pour des environnements contrôlés, où chaque acteur est garanti de supporter EXI.

P ! : Sécurité et qualité du code XML, deux questions essentielles pas toujours abordées par les éditeurs. Quelles sont les préconisations du W3C sur ces deux points et quelles sont les faiblesses actuelles ?

PLH : XML est un langage déclaratif qui ne contient pas de code exécutable. Transgresser cette approche a le potentiel d'ouvrir la porte à pas mal de problèmes de sécurité. Au niveau de la sécurité de données transmises à l'aide d'XML, W3C travaille depuis long-

temps dans ce domaine et a produit deux recommandations "XML Security" et "XML Encryption". Ces deux recommandations sont à la base de la sécurité dans les Services Web. La plus grosse faiblesse d'XML est son API: le modèle objet de document (DOM). Les navigateurs Web donnent un accès sans limite au document XML à travers le DOM, dont tirent avantage de nombreuses applications sur le Web. Ceci peut néanmoins exposer des données sensibles si le document XML provient de plusieurs sources. Plusieurs idées circulent pour contrôler l'accès et certaines sont discutées dans le groupe de travail du W3C "Rich Web Applications".

P ! : De quelle manière peut-on s'assurer d'une bonne implémentation ?

PLH : Les parseurs XML, qui n'exposent pas de problème de sécurité en général, sont plutôt jugés sur leurs performances et leur disponibilité. En général, la conformité et l'interopérabilité des parseurs XML sont correctes. La simplicité de la spécification y contribue. La création de parseurs XML est relativement simple et c'est pour cela qu'un grand nombre d'entre eux ont été développés. Ceux qui respectent les standards ont l'avantage de donner des messages d'erreurs très clairs. Il est donc préférable d'utiliser les parseurs XML existants plutôt que d'en écrire de nouveaux. Les parseurs XML inclus dans les langages de programmation et plates-formes de développement courants (ie, Java SDK ou .Net) sont généralement de bonne qualité. Le choix du parseur dépend souvent des fonctionnalités fournies et des besoins. L'interface joue un rôle important pour le choix du parseur : DOM, ou SAX, ou XOM, ou des interfaces plus spécialisées. La performance et la taille de l'exécutable seront toujours des facteurs importants, et dans certaines conditions, le support de jeux de caractères non Unicode peut être requis, comme par exemple ISO 8859-1 (ISO Latin 1). Par exemple, "XML Security" et "XML Encryption" ont été déployés il y a suffisamment longtemps pour avoir de bonnes implémentations dans les plates-formes de développement existantes.

XSLT sans douleur

Les transformations XSLT sont sans pareil pour le traitement de documents XML: transformation, analyse, publication, tout ou presque est possible à qui maîtrise ce langage.

Les principes de base du langage sont cependant trop souvent méconnus des débutants, la tendance étant d'abuser des instructions procédurales de XSLT: *xsl:if*, *xsl:choose*, *xsl:for-each*. On crée ainsi des transformations peu flexibles et peu efficaces, qui n'utilisent pas le modèle XPath/XSLT à bon escient. Cet article propose une manière d'aborder XSLT basée sur une bonne connaissance de XPath, combinée avec un jeu d'instructions XSLT volontairement réduit. Ce "chemin d'apprentissage" de XSLT a fait ses preuves lors de nos sessions d'enseignement, en permettant aux programmeurs de faire les bons choix lors de la création de leurs transformations. Une connaissance de base de XSLT sera utile pour comprendre nos exemples, et nous ne donnerons pas de détails quant à la syntaxe des instructions XPath et XSLT utilisées. Il est recommandé d'utiliser les références citées en fin d'article pour compléter ces informations.

XPath

XPath est trop souvent ignoré par les débutants en XSLT, car quasiment invisible dans des expressions simples, comme celle ci-dessous:

```
<xsl:template match="section"/>
```

Cette instruction XSLT déclare une template, un ensemble d'instructions (vide dans cet exemple) qui sera appliqué lorsqu'on rencontre un certain pattern dans le document d'entrée. On peut penser que *section* est simplement un nom d'élément; ce n'est pas faux, mais il s'agit en fait d'une expression XPath, mieux visible dans ce second exemple :

```
<xsl:template match="section[@id]"/>
```

Cette seconde template ne s'appliquera qu'aux éléments *section* qui possèdent un attribut *id*. L'important est de noter que l'attribut *match* est une expression XPath. En pratiquant XSLT, on constate rapidement que ces expressions sont responsables pour une grande part de la puissance des transformations XSLT, et que la maîtrise de XPath nous aidera énormément dans la conception de transformations concises et efficaces. Dans ce qui

suit, nous passons en revue les concepts principaux de XPath, en indiquant leur importance dans notre processus d'apprentissage.

L'arborescence XPath

La structure "textuelle" et les balises du document XML sont ignorées par le processeur XPath, qui considère seulement l'arbre résultant de l'analyse (parsing) du document XML. Les deux éléments `<para>` ci-dessous, par exemple, sont absolument identiques du point de vue du processeur :

```
<para/>
<para></para/>
```

L'arbre XPath contient sept types de noeuds différents: quatre sont utilisés fréquemment (racine, élément, attribut, texte) et les trois autres (commentaires, instructions de traitement et espaces de noms) nettement plus rarement.

Rappelons que le noeud racine n'existe pas vraiment dans le document XML d'entrée, ce noeud est rajouté artificiellement dans l'arborescence XPath comme conteneur pour le document. L'expression XPath `/html` sélectionne donc bien l'élément racine *html* du document d'entrée, situé en-dessous du noeud racine artificiel dans l'arbre XPath.

La connaissance de ces types de noeuds aidera à l'élaboration d'expressions XPath précises. Les notations utilisées pour sélectionner les noeuds de ces différents types sont parfois peu évidentes. L'expression `//text()`, par exemple n'est pas un appel de fonction mais bien la sélection de tous les noeuds de type texte du document.

Chemins de sélection

Les chemins XPath sont en apparence très semblables à ceux que l'on utilise dans les systèmes de fichiers, comme `/section/para`. La différence essentielle est qu'un tel chemin pointe dans un système de fichiers sur un seul objet, alors que dans XPath cette expression sélectionne tous les éléments dont le nom est "para", qui sont enfants d'un élément "section" situé à la racine du document. Le résultat est donc un ensemble de noeuds, qui seront des éléments dans notre exemple, mais peu-

vent selon l'expression utilisée être de n'importe quel type parmi les sept types de noeuds mentionnés ci-dessus.

Les chemins XPath sont donc plus proches conceptuellement des requêtes SQL que de chemins sur un disque: on peut les considérer comme des filtres qui, appliqués sur l'arborescence du document XML, laissent passer certains noeuds et pas d'autres.

Les chemins absolus commencent par un slash, comme `/section/para`, et s'appliquent donc à partir du noeud racine. Les chemins relatifs comme `section/para`, sans le slash initial, s'appliquent à partir du *noeud contextuel*, celui que le processeur XSLT est en train de traiter au moment de l'évaluation de l'expression XPath.

Le double slash permet de faire la différence entre une relation parent-enfant, comme dans le chemin `section/para`, ou parent-descendant, comme dans le chemin `section//para` qui utilise ce double slash. Le premier exemple sélectionne les éléments *para* qui sont juste en-dessous d'un élément *section*, alors que le second sélectionne les éléments *para* qui ont un élément *section* comme ancêtre, quel que soit le nombre de niveaux hiérarchiques qui les séparent. Ces principes de construction de chemins XPath sont très importants pour la bonne construction et compréhension de transformations XSLT.

Prédicats

Les prédicats XPath sont similaires à la clause *where* du langage SQL: ils permettent des sélections précises, en ignorant les noeuds d'un chemin qui ne remplissent pas les conditions définies par le prédicat.

L'exemple ci-dessous sélectionne tous les éléments *div* possédant un attribut *style* qui vaut *abc*, et qui sont situés en-dessous d'un élément *section* qui contient au moins un élément enfant *para*.

```
section[para]//div[@style='abc']
```

La concision d'une telle expression, par rapport à sa description en langue française, démontre combien la syntaxe XPath est expressive. Les prédicats sont utilisés très fréquemment dans les transformations XSLT. Ils

permettent souvent d'éviter les constructions procédurales comme *xsl:if*, par l'utilisation de templates XSLT plus "ciblées". Il est donc important de connaître leur principe, ainsi que l'étendue des fonctions XPath que les prédicats peuvent utiliser.

Axes

En XPath, les axes permettent de naviguer dans l'arborescence du document XML, dans toutes les directions: parents/enfants (directs ou à plusieurs niveaux), noeuds frères précédents ou suivants, et noeuds précédents ou suivants sans tenir compte de la hiérarchie. L'exemple ci-dessous sélectionne les éléments `table` qui ont le même parent et suivent un élément `para` ancêtre du noeud courant.

```
ancestor::para/following-sibling::table
```

L'utilisation des axes XPath n'est pas très fréquente, mais il est important de connaître les types d'axes à disposition pour savoir les utiliser à bon escient.

Tout ceci n'est pas très simple...

La promesse d'un XSLT *sans douleur* semble un peu lointaine au vu de l'étendue des connaissances XPath mentionnées comme nécessaires. Mais le plus dur est fait, XPath étant effectivement ce qui fait la puissance des transformations, plus que les instructions XSLT qui sont comme on le verra relativement simples. Bien comprendre le rôle de XPath dans XSLT, et le principe de sélection des noeuds par le processeur XPath, est capital pour utiliser XSLT de manière efficace. Il est donc conseillé, à partir des références figurant en fin d'article, d'étudier les éléments XPath mentionnés ci-dessus, avant même d'aborder l'utilisation de XSLT, ou en tout cas tout au début de l'apprentissage.

XSLT

L'exemple ci-dessous démontre une fois de plus l'importance de XPath dans les transformations XSLT. Les valeurs des attributs `match`, `class` et `select` de cette transformation sont toutes des expressions XPath.

```
<xsl:template
  match="section[para[normalize-space(.)]]">
  <div class="{@style}">
    <h1><xsl:value-of select="title"/></h1>
    <xsl:apply-templates select="para[@ok='true']"/>
  </div>
</xsl:template>
```

En supprimant ces expressions, on fait apparaître les instructions XSLT et les éléments HTML littéraux de notre transformation :

```
<xsl:template
  match="">
  <div class="">
    <h1><xsl:value-of select=""/></h1>
    <xsl:apply-templates select=""/>
  </div>
</xsl:template>
```

Cette nouvelle transformation n'est bien entendu pas utilisable telle quelle, mais sa simplicité montre bien qu'une grande partie de la complexité et de la puissance des transformations XSLT provient de XPath.

Nous présentons dans ce qui suit, les éléments XSLT indispensables à l'apprentissage.

Jeu d'instructions réduit

Notre premier conseil pour XSLT est de se limiter à un jeu d'instructions réduit, pour éviter de tomber dans le piège des constructions procédurales. On se limitera dans un premier temps à ce jeu d'instructions : *xsl:stylesheet*, *xsl:template*, *xsl:apply-templates*, *xsl:value-of* (y compris la variante utilisée pour les attributs, comme dans `<p class="{@id}">` ou les accolades sont équivalentes à une instruction *value-of*). Ces quelques instructions permettent de réaliser des transformations relativement complexes, pour autant que l'on profite au maximum de la puissance de XPath dans leurs attributs *match* et *select*. On sera assez rapidement amené à utiliser également les instructions *xsl:variable*, *xsl:param*, et plus tard *xsl:call-template*, les modes d'exécution de *xsl:template*, et *xsl:key* pour la création de tableaux associatifs. Ces instructions influencent cependant peu la structure générale de la transformation, qui est déterminée principalement par les quatre instructions de notre jeu réduit. Les instructions *xsl:include* et *xsl:import* permettent de mieux modulariser et réutiliser nos templates XSLT, mais, comme les instructions de la liste ci-dessus, influencent peu la structure générale des transformations. Avec XPath, les quatre instructions de base constituent les pièces maîtresses de nos transformations et méritent donc d'être étudiées en détail.

xsl:if, xsl:choose, xsl:for-each

Attention, danger! Ces instructions "procédurales" sont parfois nécessaires, mais nette-

ment moins souvent que la plupart des débutants en XSLT ne le pensent.

L'exemple typique est celui d'une template basée sur *xsl:choose*, qui sera avantageusement remplacée par plusieurs templates avec des expressions de sélection XPath plus spécifiques. Avec *xsl:choose*, notre template (exemple ci-dessous) ressemble un peu à un fragment de code Java ou C :

```
<xsl:template match="something">
  <xsl:choose>
    <xsl:when test="@id">
      ...cas 1
    </xsl:when>
    <xsl:otherwise>
      ...cas 2
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

Alors qu'il suffit de faire de chaque *xsl:when* une template séparée pour obtenir un code plus lisible, plus facile à maintenir et que le processeur exécute en général bien plus efficacement :

```
<xsl:template match="something[@id]">
  ...cas 1
</xsl:template>

<xsl:template match="something">
  ...cas 2
</xsl:template>
```

De la même manière, l'instruction *xsl:if* peut souvent être évitée en utilisant des expressions XPath plus précises dans les attributs `match` de *xsl:template*, alors que *xsl:for-each* sera avantageusement remplacée par une template XSLT récursive. Ces instructions procédurales ne peuvent pas être évitées totalement, elles ont leur utilité. Il est cependant recommandé de ne les utiliser qu'en dernier ressort.

Scénario de traitement du processeur XSLT

Le principe de traitement de l'arbre d'entrée par le processeur XSLT est simple, mais souvent méconnu : Le processeur commence par visiter le noeud racine du document. Pour chaque noeud qui est visité, le processeur :

1. Définit ce noeud comme "noeud contextuel" pour les expressions XPath
2. Vérifie si une template s'applique à ce noeud
- 3a. Si oui, il exécute la template, le traitement est alors terminé.

3b. Si non, il exécute les règles par défaut. Rien de compliqué dans ce scénario, qui s'exécute tant qu'il reste des noeuds à traiter, ce qui dépend de la manière dont le document est visité par les instructions XSLT et les règles par défaut.

Ce scénario explique pourquoi la transformation ci-dessous s'arrête immédiatement, après avoir visité le noeud racine.

```
<xsl:stylesheet>
<xsl:template match="/" />
</xsl:stylesheet>
```

Le processeur commence la visite par le noeud racine, trouve une template qui s'applique à ce noeud et l'exécute. La template ne fait rien, car son corps est vide, et le traitement se termine selon la règle 3a. Aucun autre noeud n'est visité, car la template exécutée n'utilise pas d'instruction de visite comme `xsl:apply-templates`.

Règles par défaut

Selon la règle 3b ci-dessus, les règles par défaut sont exécutées pour chaque noeud que

le processeur visite et auquel aucune template ne s'applique.

Il est donc nécessaire de connaître ces règles par défaut, pour comprendre par exemple que la transformation ci-dessous visite complètement le document d'entrée, en copiant les noeuds de texte sur sa sortie :

```
<xsl:stylesheet/>
```

Cette transformation minimale est tout à fait valide, et son comportement s'explique en combinant le scénario d'exécution du processeur, ci-dessus, avec les règles par défaut. Ces règles définissent que, si aucune template ne s'applique:

Le noeud racine et les éléments sont visités. Les noeuds de texte sont copiés sur la sortie. Les autres types de noeuds sont ignorés.

La connaissance de ces règles par défaut est essentielle, pour comprendre le comportement de nos transformations et pour éviter d'écrire des templates inutiles, comme dans cet exemple :

```
<xsl:template match="p">
<xsl:apply-templates/>
</xsl:template>
```

Cette template reproduit exactement la règle par défaut qui dit qu'un noeud est visité, ce que fait aussi `xsl:apply-templates`. On peut donc, dans le cas général, la supprimer sans modifier le comportement de la transformation XSLT.

Conclusion

Notre rapide tour d'horizon nous montre qu'une bonne connaissance de XPath est essentielle pour utiliser efficacement XSLT. Les instructions XSLT elles-mêmes posent en général moins de problèmes de compréhension, et l'utilisation de notre jeu d'instructions réduit aide à utiliser XSLT de la manière préconisée par ses concepteurs. Le "chemin d'apprentissage" proposé dans cet article devrait vous aider à aborder XSLT dans le bon sens - bonne chance dans votre exploration !

■ Bertrand Delacrétaiz

Développeur senior au sein du team R&D de Day Software (www.day.com). Membre de la fondation



Apache. Il a contribué aux projets Apache Cocoon, FOP, Solr, et Tika, et travaille actuellement sur le projet Apache Sling, un framework web basé sur le Java Content Repository (JCR).

Mieux que DOM et SAX Hydra SDO XML

Combine la souplesse de traitement apportée par DOM, avec la puissance de gestion des gros fichiers fournie par SAX

- Optimise pour une faible consommation mémoire et des temps de réponse rapides, parse de très gros fichiers qui excèdent les capacités de la technologie DOM
- Conserve en mémoire l'ensemble des informations permettant toutes les opérations et traitements sur le contenu du fichier

Une seule version pour 2 langages de programmation

- Implementation strictement identique en Java et C++



Contactez dès aujourd'hui Emmanuel Blattes, Ingénieur au Service Technique. Il vous aidera à évaluer HydraSDO XML avec vos propres fichiers.

Rogue Wave France
40, rue des Vignobles - 78400 Chatou - Tel +33 (0)1 30 09 78 78

Validation des documents XML

Après avoir vu dans la première partie la validation par DTD, nous examinerons ce mois-ci la validation par schéma.

La validation par un schéma W3C

Un schéma W3C (que l'on nommera par la suite schéma) est une grammaire définie dans un formalisme XML. Sauf pour la gestion des entités, on peut considérer les schémas comme remplaçant les DTD. La version officielle est la 1.0 mais une version 1.1 est en préparation au moment de la rédaction de cet ouvrage. Quelques caractéristiques des schémas :

- gestion des espaces de noms ;
- types de base riches et extensibles ;
- réutilisation par importation et héritage ;
- davantage de souplesse dans les cardinalités ;
- proximité avec les schémas XDR (XML-Data Reduced).

Les différentes formes de type

Dans un schéma, les types sont fondamentaux. Ils ont la caractéristique d'être disponibles à la fois pour les éléments et les attributs. Examinons, tout d'abord, le typage des éléments. Pour cela, passons en revue les contenus possibles pour un élément (que l'on nomme parfois modèle de contenu) :

- Vide (entraîne EMPTY avec une DTD) : pas de contenu.
- Simple (entraîne #PCDATA avec une DTD) : du texte.
- Complexe : un ou plusieurs éléments.
- Mixte : un mélange d'éléments et de texte.

À partir de ces contenus possibles, il ressort dans les schémas 2 types :

- simple : du texte pour attribut ou élément ;
- complexe : avec au moins un attribut ou un élément fils.

Remarque
Le typage complexe va également s'appliquer dans le cas d'un contenu simple mais avec au moins un attribut. Il ne faut donc pas faire la confusion entre contenu simple et type simple.

Figure 3-1
Contenu d'un élément et typage

contenu de l'élément		attribut	types
vide et texte	contenu simple	sans attribut	type simple
		avec attributs	
élément fils ou mixte avec texte	contenu complexe	sans attribut	type complexe
		avec attributs	

La figure 3-1 reprend le parallèle entre contenu d'un élément et typage. On le voit bien, la présence d'attributs déclenche nécessairement le passage en typage complexe. Examinons maintenant le typage des attributs : il s'agit d'un typage simple qui suit donc le même principe que le typage d'un élément ne contenant que du texte et sans attributs.

Pour résumer :

- Pas de type : élément sans contenu et sans attribut.
- Type simple : Attribut ; élément avec du texte seulement.
- Type complexe : tous les autres cas d'usage de l'élément.

Les définitions globales et locales

Dans une DTD toutes les définitions sont globales. Cela signifie qu'il n'est pas possible d'utiliser plusieurs définitions pour un même nom d'élément. L'exemple suivant ne peut pas être géré avec une DTD :

```
<document>
<titre>...</titre>
<auteur>

<titre>...</titre>
</auteur>
</document>
```

On voit bien que la balise titre n'a pas le même sens pour le document ou pour l'auteur. Pour réaliser une DTD, il faudrait décrire toutes les possibilités de contenu pour l'élément titre, avec la conséquence que l'on ne pourrait contrôler l'usage de telle ou telle possibilité (par exemple, le titre de l'auteur employé à la place du titre du document). Dans un schéma, nous avons le choix, soit de créer une définition globale, soit de créer une définition locale (donc contextuelle à un élément parent). Une définition globale concerne également d'autres parties du schéma, comme les types, groupes, attributs...

L'assignation d'un schéma à un document XML

Le document XML est considéré comme une instance du schéma. Ce terme emprunte au monde objet certaines caractéristiques que nous aborderons ultérieurement. Pour que le parseur puisse utiliser un schéma au cours de la validation, on insère dans le document XML des attributs supplémentaires sur l'élément racine. Ces attributs devront appartenir à l'espace de noms <http://www.w3.org/2001/XMLSchema-instance>. Il existe deux formes de liens :

- Le cas ignorant les espaces de noms du document. Dans ce cas, il faut utiliser l'attribut `noNamespaceSchemaLocation` pour désigner la localisation du schéma à employer. Cette localisation peut être relative au document XML ou absolue, contenir un chemin (path) ou une URL.
- Le cas prenant en compte les espaces de noms du document. Dans ce cas, on utilise l'attribut `schemaLocation`, qui a pour valeur une liste de couples (espace de noms et localisation d'un schéma). Nous aborderons cette syntaxe en détail de la gestion des espaces de noms.

Exemple :

```
<?xml version="1.0" encoding="UTF-8"?>
<personnel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
_xsi:noNamespaceSchemaLocation="personal.xsd">
...
</personnel>
```

Dans cet exemple, nous indiquons au parseur qu'il peut charger le schéma `personal.xsd` se trouvant dans le même répertoire que notre document XML.

Les catégories de type simple

Les types applicables aux attributs et au contenu d'éléments simples (sans attribut) sont divisés en 2 catégories :

- Types normalisés : tous les caractères blancs (tabulation, retour chariot...) sont remplacés par un caractère unique.
 - Types compactés : les occurrences de blancs en tête et queue sont supprimées. Les séries contiguës de blancs sont remplacées par un blanc.
- Les types de base sont toujours liés à l'espace de noms des schémas pour éviter toute collision avec votre propre type. Lorsqu'on nomme un type de base, on le préfixe par xs: pour signifier la présence d'un espace de noms (et non pour obliger à utiliser le préfixe xs).

Figure 3-2
Répartition des types de base

Non normalisé et non compacté	Normalisé et non compacté	Normalisé et compacté
xs:string	xs:normalizedString	xs:byte xs:int xs:long xs:float xs:double xs:boolean xs:date xs:time xs:dateTime ...

La figure 3-2 représente quelques types de base que nous allons décrire dans la suite. On retiendra que le type xs:string est le type le plus permissif, dont on comprend aisément qu'il désigne une chaîne de caractères dont on souhaite conserver les blancs.

Voici un exemple d'utilisation de type simple :

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"> <xs:
element name="titre" type="xs:string"/> <xs:element name="numchapitre"
type="xs:int"/>
</xs:schema>
```

Sans trop rentrer dans les détails de la structure d'un schéma, vous observerez que le typage des éléments titre et numchapitre est lié à l'attribut type. Voici deux documents possibles, valides par rapport à ce schéma :

```
<titre>Un titre</titre>
```

et

```
<numchapitre>10</numchapitre>
```

Les types pour les chaînes de caractères

Les types disponibles pour les chaînes de caractères sont les suivants :

- xs:string : le plus simple sans normalisation ni compactage ;
 - xs:normalizedString : la forme normalisée et non compactée ;
 - xs:token : la forme normalisée et compactée ;
 - xs:language : les codes de langue (RFC 1766) ;
- xs:NMTOKEN : pas de blanc (comme 1999-10-10, 1234534 ou employe1023) ;
- xs>Name : forme de NMTOKEN commençant par une lettre ;
 - xs:ID : chaîne unique dans le document ;
 - xs:IDREF : chaîne ayant pour valeur un ID (d'où le mot REF pour référence) ;
 - xs:anyURI : la valeur de l'URI utilisera des caractères ASCII, conversion automatique. Par exemple, <http://www.site.com/mon Rep> devient <http://www.site.com/mon%20Rep>.

Les types pour les dates et heures

Les dates et heures sont un sous-ensemble du standard ISO 8601 et s'appuient sur le calendrier grégorien (occidental). Les fuseaux horaires sont pris en compte par décalage avec le temps universel (TU). Cepen-

dant, les heures d'été et d'hiver ne sont pas gérées, ce dont il faudra tenir compte dans vos applications pour les comparaisons d'heure ou de date. Les types disponibles sont les suivants :

- Types : xs:dateTime, xs:date, xs:time, xs:duration, xs:gYearMonth, xs:gYear, xs:gDay, xs:gMonth ;
- Format lexical complet de xs:dateTime : CCYY-MM-DDThh:mm:ssZff.

Quelques exemples du type xs:dateTime :

- 2006-08-21T12:07:00 : fuseau horaire inconnu ;
- 2006-08-21T12:07:00+02:00 : décalage de 2 heures avec le TU ;
- 2006-08-21T12:07:00Z2 : décalage de 2 heures avec le TU.

Les types numériques

Les différents types numériques disponibles sont les suivants :

- Le plus simple xs:int (32 bits) : entier, également xs:unsignedInt.
- xs:long (64 bits), xs:short (16 bits), xs:byte (8 bits) : entier ; il existe également xs:unsignedLong, xs:unsignedShort, xs:unsignedByte.
- xs:float (32 bits) : flottant simple précision ; xs:double (64 bits) : flottant double précision. Quelques exemples : 3.14E10, 3.14, 4.14E-10, INF, -INF, NaN.
- xs:decimal : nombre sans limitation ; le point est le séparateur décimal. Quelques exemples : 3.14, +3.14, -3.14.
- xs:integer : forme de xs:decimal sans la partie décimale.
- xs:nonPositiveInteger : entier négatif avec le zéro ; xs:negativeInteger : entier négatif sans le zéro.
- xs:boolean : true ou false ou 1 ou 0.

La création de nouveaux types simples

Les types simples peuvent être étendus pour créer de nouveaux types. Il existe trois possibilités de création :

- La restriction : comme son nom l'indique, on crée un sous-type en limitant certaines caractéristiques (par exemple, la longueur d'une chaîne). Les limitations sont définies avec des facettes.
- L'union : l'union sert à autoriser plusieurs types, ce qui permet d'offrir des alternatives, comme un attribut ayant pour valeur soit un entier soit une constante.
- La liste : il s'agit d'un ensemble de valeurs de même type séparées par un blanc. Il est également possible d'appliquer une restriction sur une liste, notamment pour en borner la taille.

Une première forme de création : la dérivation par restriction de type simple

Une restriction fait intervenir des balises qui vont réduire l'espace des valeurs possibles pour un type de base. Ces balises ont toutes un attribut value pour définir l'ampleur de la limitation.

La facette whiteSpace agit sur les chaînes de caractères. Elle peut prendre les valeurs reserve (blancs conservés), replace (blancs normalisés) ou bien collapse (occurrences de blanc ramenées à 1 blanc &x20).

Exemple :

```
<xs:simpleType name="chaîneCompacte"> <xs:restriction base="xs:
String"> <xs:whiteSpace value="collapse"/> </xs:restriction> </xs:simple
Type>
```

Dans cet exemple, nous avons créé le type chaîneCompacte qui ne conserve pas les occurrences contiguës de blancs.

La facette `pattern` agit également sur les chaînes de caractères en vérifiant que la valeur est compatible avec une expression régulière. Plusieurs facettes peuvent être utilisées et, dans ce cas, il suffira que la valeur concorde avec au moins l'une des expressions régulières définies. Si vous n'êtes pas familier avec les expressions régulières, voici un rapide résumé.

Les expressions régulières (concept lié au *pattern matching*) représentent un langage pour décrire des mots (motifs) d'un point de vue syntaxique. Elles sont présentes dans la plupart des langages de programmation (Java, Perl...).

Voici une manière simple de les employer pour reconnaître un motif toujours de même forme :

```
<xs:simpleType nom="monEntier">
  <xs:restriction base="xs:int"> <xs:pattern value="10"/> <xs:pattern
  value="30"/>
</xs:restriction> </xs:simpleType>
```

Dans cet exemple, nous autorisons pour le type `monEntier`, ou bien l'entier 10 ou bien l'entier 30. Cet exemple se rapproche d'une énumération. Les caractères {}, *, + et ? vont servir à exprimer des quantités de caractères :

- {m,n} signifie de m à n ; o * signifie 0 ou plus ; o + signifie au moins 1 ;
- ? est l'option (0 ou 1). Exemple :

```
<xs:simpleType> <xs:restriction base="xs:int"> <xs:pattern value="0{1,
2}10?" /> </xs:restriction> </xs:simpleType>
```

Dans cet exemple, nous autorisons les entiers 01, 001, 010 et 0010. Le langage des expressions régulières comprend également des classes de caractères :

- \s : blanc ;
- \S : tout sauf un blanc ;
- \d : un chiffre ;
- \D : tout sauf un chiffre ;
- \w : caractère alphanumérique plus "-" ;
- \W : tout sauf un caractère alphanumérique plus "-". Exemple :

```
<xs:pattern value="\w\d+\w"/>
```

Cet exemple limite une valeur à un nombre alphanumérique de taille quelconque entouré de 2 caractères mot (A123b, r1t...). Il est également possible de spécifier des plages de caractères avec les caractères crochets. Exemples :

[0-9] : un chiffre [a-Z] : une lettre minuscule [a-zA-Z] : une lettre minuscule ou majuscule [^0] : tout sauf 0 [0-9E] : un chiffre ou E

Exemple :

```
<xs:pattern value="[0-9]+[a-z]"/>
```

Ce dernier pattern comprend au moins un chiffre suivi d'un caractère alphabétique en minuscule (1a, 123b, 99z...). Lorsqu'on souhaite reconnaître une valeur dans une liste de valeurs possibles on dispose, outre le pattern, de l'énumération (mot-clé `enumeration`). Exemple :

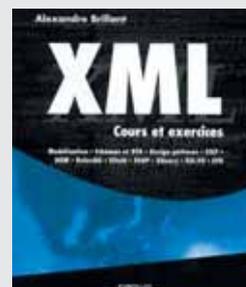
```
<xs:simpleType name="couleurType">
  <xs:restriction base="xs:string"> <xs:enumeration value="bleu"/>
  <xs:enumeration value="blanc"/> <xs:enumeration value="rouge"/>
</xs:restriction> </xs:simpleType>
```

Dans cet exemple, nous autorisons pour le type `couleurType` les valeurs bleu, blanc, rouge.

Pour les chaînes et nombres, nous disposons des facettes suivantes :

- Avec les chaînes :
 - `xs:length` : longueur en nombre de caractères ;
 - `xs:maxLength` : longueur maximale ;
 - `xs:minLength` : longueur minimale.
- Pour les nombres et les dates :
 - `xs:maxExclusive` / `xs:maxInclusive` : borne supérieure ;
 - `xs:minExclusive` / `xs:minInclusive` : borne inférieure.
- Pour les nombres seulement :
 - `xs:totalDigits` : nombre de chiffres d'un entier ;
 - `xs:fractionDigits` : nombre de chiffres après la virgule.

Autre forme de création : la construction de liste. L'autre forme de dérivation d'un type simple est la liste de valeurs. Les valeurs d'une liste sont séparées par un blanc et il est impossible d'utiliser un autre séparateur. Dans une troisième partie, nous terminerons la validation par schéma.



Extrait de l'ouvrage :
XML Cours et exercices
Editions Eyrolles
Alexandre Brillant
29,90 €
284 pages
www.editions-eyrolles.com
Site de l'auteur :
www.abrillant.com

L'information permanente

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

www.programmez.com

La sérialisation XML

1^{re} partie

La sérialisation est un processus qui convertit un objet en un flux qui peut être transporté ou sauvé facilement.

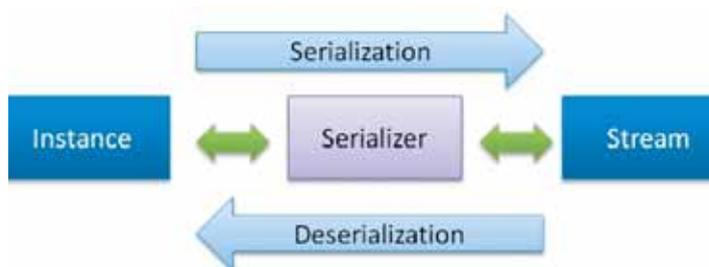
Par exemple, il est possible de sérialiser un objet et de le transporter via Internet entre un client et un serveur à l'aide du protocole HTTP. À l'autre extrémité, le processus inverse va se produire pour reconstruire l'objet à partir du flux : la désérialisation. Le mécanisme qui va retenir notre attention, est celui de la sérialisation XML, à savoir la sauvegarde de l'état d'un objet O à un certain temps T dans le format XML. Les applications sont nombreuses, tout programme devant stocker son état étant un client potentiel pour utiliser ce mécanisme. D'autres technologies comme le remoting, les Web Services ou WCF permettent de sérialiser automatiquement des objets vers du XML. Nous ne verrons pas ces mécanismes puisqu'ils dépassent le cadre de cet article.

Explication du mécanisme de la sérialisation

En .NET, il existe plusieurs types de sérialisations différents :

- La sérialisation binaire
- la sérialisation SOAP (XML)
- la sérialisation XML
- la sérialisation JSON

Toutes ont un point commun, à savoir, enregistrer l'état d'un objet de telle façon que nous puissions le recréer ultérieurement (par le méca-



nisme inverse, à savoir la désérialisation). Chaque type de sérialisation a ses spécificités propres. Il est évident que la grosse différence entre une sérialisation XML et binaire est la sortie, qui produit une fois du texte formaté et donc lisible par un humain, contrairement au binaire, compréhensible seulement par une machine. Comme nous nous intéressons dans cet article à la sérialisation XML, nous mettrons de côté ce qui touche à la sérialisation binaire et JSON. Examinons maintenant les spécialités des deux types de sérialisation utilisant XML

• **Sérialisation XML** Sa principale caractéristique est qu'elle ne convertit pas de méthodes, d'indexeurs, de champs privés ou de propriétés en lecture seule (sauf les collections en lecture seule). Ce type de sérialisation est également connu sous le terme de sérialisation en surface. De plus, la classe doit être publique et contenir un constructeur sans paramètre. Si tel n'est pas le cas, une exception sera levée lors de la (dé)sérialisation.

• **Sérialisation SOAP** L'objet qui va être sérialisé doit absolument être marqué avec l'attribut *Serializable*. Cette fois-ci, l'ensemble des propriétés et champs de l'objet seront sérialisés.

Sauvegarde d'objet sur le disque

Pour montrer le fonctionnement de la sérialisation XML, nous allons construire un petit exemple et le soumettre aux deux types de *sérializer*. Nous pourrions ainsi comparer les sorties obtenues. Implémentons un objet tout simple qui possède deux attributs (un public et un privé) ainsi que deux propriétés publiques. Voici à quoi il va ressembler :

```

[Serializable] // Requis par la sérialisation SOAP
public class Person
{
    private int _age = 0;
    public char sex = 'U'; // Unknow

    // Constructeur vide, requis par la sérialisation XML
    public Person() { }

    public Person(int age, char sex)
    {
        this._age = age;
        this.sex = sex;
    }

    /// Les propriétés publiques DOIVENT être en lecture et
    /// écriture. Si tel n'est pas le cas, une exception sera
    /// levée lors de la sérialisation XML !
    public string Name { get; set; }

    public Color PreferredColor { get; set; }
}
  
```

Commençons par la sérialisation XML

Nous utilisons maintenant l'objet *Person* (nous l'initialisons) et nous le sauvons sur le disque grâce à la création de l'objet *XmlSerializer* du namespace *System.Xml.Serialization*. Les méthodes principales de cet objet sont les méthodes *Serialize* et *Deserialize* qui permettent respectivement d'écrire ou de lire un objet.

```

var person = new Person(24, 'M')
    { Name = "Bidou", PreferredColor = Color.Blue };

using (var xWriter = XmlWriter.Create(path, new XmlWriterSettings()
    { Encoding = Encoding.UTF8 }))
{
    var xSer = new XmlSerializer(typeof(Person));
    xSer.Serialize(xWriter, person);
}
  
```

Comme nous l'avons fait remarquer quelques lignes plus haut, seules

les propriétés et champs **publics** vont être sauvés. Jetons un œil sur le résultat obtenu.

```
<?xml version="1.0" encoding="utf-8" ?>
<Person xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <sex>77</sex>
  <Name>Bidou</Name>
  <PreferredColor />
</Person>
```

Le résultat est bien conforme à notre attente : La propriété publique Name a été sauvée et a pris la valeur 'Bidou' tout comme le champ public qui a pris la valeur du caractère ASCII 'M', à savoir 77. Le champ privé `_age` a été omis et la propriété publique `PreferredColor` est devenue un élément vide car cette structure ne possède aucun champ ou propriété public (cette valeur est donc "perdue").

Il est à noter que les types complexes sont également sauvés par ce processus (nous ne sommes donc pas limités à des variables de type primitif). Autrement dit, si notre objet Person exposait une propriété publique de type Address, la sérialisation aurait également fonctionné et nous aurions eu un résultat de ce type

Il est maintenant fort simple de restaurer notre objet préalablement sauvé. On suit le même cheminement que pour la sérialisation, à la différence que l'on appelle la méthode `Deserialize` et que l'on crée un *reader* au lieu d'un *writer* pour le flux.

```
Person person = null;
using (var xReader =
  XmlReader.Create(path, new XmlReaderSettings()))
{
  var xSer = new XmlSerializer(typeof(Person));
  person = xSer.Deserialize(xReader) as Person;
}
```

Si on inspecte les valeurs de la variable `person`, on constate qu'elle contient bien les valeurs attendues, à savoir 'Bidou' et 'M'. Le champ privé vaut 0, `PreferredColor` est noir ; leurs valeurs par défaut.

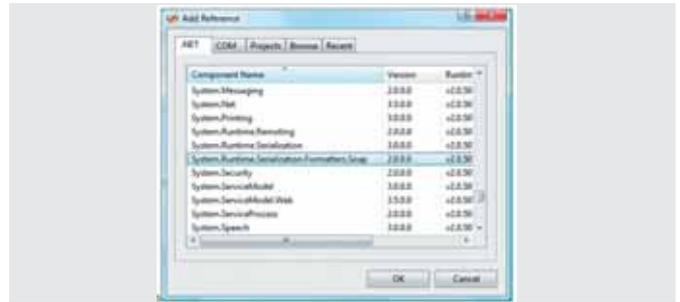
Le code ci-dessus a été allégé au maximum, le programmeur devra naturellement prendre soin de gérer les diverses exceptions pouvant survenir lors de la manipulation...

Inspectons maintenant la sérialisation avec SOAP.

En ajoutant l'assembly `System.Runtime.Serialization.Formatters.Soap` en référence au projet, on aura accès aux classes nécessaires pour ce type de sérialisation. Nous utilisons donc le même objet que précédemment et changeons uniquement de sérialiser. Nous obtenons le code suivant pour la sérialisation

```
var person = new Person(24, 'M')
  { Name = "Bidou", PreferredColor = Color.Blue };

using (var fWriter = new FileStream(path, FileMode.Create))
{
  var SOAPSer = new SoapFormatter();
  SOAPSer.Serialize(fWriter, person);
}
```



Allons voir si le résultat est à la hauteur de nos attentes :

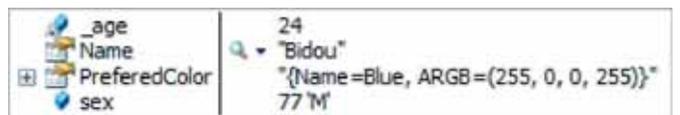
```
<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" ...>
  <SOAP-ENV:Body>
    <ns1:Person id="ref-1"
      xmlns:ns1="http://schemas.microsoft.com/...">
      <_age>24</_age>
      <_x003C_sex_x003E_k__BackingField:M</_x003C_sex_x003E_k__BackingField>
      <_x003C_Name_x003E_k__BackingField id="ref-4">Bidou</_x003C_Name_x003E_k__BackingField>
      <_x003C_PreferredColor_x003E_k__BackingField>
        <name xsi:nil="true" />
        <value>0</value>
        <knownColor>37</knownColor>
        <state>1</state>
      </_x003C_PreferredColor_x003E_k__BackingField>
    </ns1:Person>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Premier constat, le sérialiser SOAP est bien plus verbeux que le traditionnel sérialiser XML. Ensuite, on remarque immédiatement que les types précédemment "oubliés" ont cette fois été inclus (`_age = 24` et `PreferredColor` est devenu un élément possédant lui-même 4 sous-éléments décrivant la couleur, ici du bleu). Si on lit à présent les valeurs du fichier XML préalablement créé avec le sérialiser SOAP, on obtient le code suivant

```
Person person = null;
using (var fReader = new FileStream(path, FileMode.Open))
{
  var SOAPSer = new SoapFormatter();
  person = SOAPSer.Deserialize(fReader) as Person;
}
```

Comme on pouvait s'y attendre, notre objet contient cette fois les mêmes valeurs qu'auparavant, y compris le champ privé et la `Color`.

Le code ci-dessus a été allégé au maximum ; le programmeur devra gérer les diverses exceptions pouvant survenir.



Si nous récapitulons les 3 types de sérialiser que nous avons évoqués, nous obtenons le tableau suivant

	XML	SOAP	Binaire
Human readable	Oui	Oui	Non
Communication inter-plate-forme	Oui	Oui	Non
Sérialisation de types non standard	Non	Oui	Oui
Sérialisation des éléments privés	Non	Oui	Oui
Utilisation faite dans le framework .NET		Web Services	Remoting

On préférera donc la sérialisation XML si nous n'avons pas besoin de sauver tous les éléments composant un état et si les performances sont nécessaires, SOAP pour sa compatibilité multi-plate-forme, sa complétude (induit une lenteur) et sa facilité de passer à travers les firewalls et la sérialisation binaire pour ses bonnes performances.

■ Nicolas Biedermann - MVP C#

www.jaimefouetter.com

www.jaimefouetter.fr



Évitez toute confusion,
déposez votre nom de domaine
en .com ET en .fr !



2€ HT/an

soit 2,39 € TTC/an
au lieu de 12 € HT/an, soit 14,35 € TTC/an

Du 1^{er} juin
au 31 juillet 2008

Code promo : .FR-AZE

**LE PACK "WEB NOM+.FR" :
VOTRE NOM DE DOMAINE .FR,
DE NOMBREUX SERVICES,
FACILEMENT, RAPIDEMENT
ET À 2 € HT/AN SEULEMENT !**

- Votre nom de domaine en .fr
- Une redirection web transparente
- Un compte de messagerie
- Un antivirus et antispam
- Au choix : un blog, un album photo ou un outil de gestion de contenu
- Un outil de création de site en ligne (2 pages)
- Votre boutique en ligne "e-commerce Free"
- Un hébergement de 1 Go
- 25 € offerts pour promouvoir votre site sur GOOGLE



Pour plus de renseignements : 0892 55 66 77 (0.34 €/mn) ou www.amen.fr
NOMS DE DOMAINE - EMAIL - HÉBERGEMENT - CRÉATION DE SITE - E-COMMERCE - RÉFÉRENCIEMENT

Microsoft Expression 2 : une évolution

Un an après la sortie de la version 1.0 et des nombreuses versions " preview " et " bêta ", Microsoft vient de lancer Expression 2 en version anglaise. La date de lancement officielle de la version française a été annoncée pour le 6 Juin. Cette gamme est composée d'Expression Web 2, Blend 2, Design 2, Media 2 et d'un nouveau venu dans la suite : Encoder 2. Vous pouvez d'ores et déjà retrouver les pages web respectives de chacun de ces produits sur le site officiel. Il est aussi intéressant de préciser que le tout est disponible dans un package nommé " Expression Studio 2 ". Celui-ci est téléchargeable à l'adresse suivante : <http://www.microsoft.com/expression/>.

Expression web 2.0 : miser sur PHP

Un nouveau Framework

Expression Web supporte à présent le Framework .Net 3.5 à l'instar de la gamme Visual Studio 2008. Expression Web vous propose donc via la ToolBox la possibilité d'inclure les nouveaux contrôles d'ASP.NET 3.5 tels que le contrôle ListView qui permet de faire des listes HTML sémantique à base de et de , le DataPager qui lui permet de paginer n'importe quelle source de données, et de nombreux autres contrôles (fig. 1). On notera également que cette nouvelle mouture propose un support plus abouti d'ASP.NET Ajax, avec notamment le support de l'IntelliSense JavaScript.

Les standards

Expression Web devient de plus en plus soucieux au niveau des normes W3C (XHTML, CSS...). La prévisualisation des pages en CSS a elle aussi été grandement améliorée, avec

notamment la propriété " overflow ", permettant un rendu beaucoup plus fidèle dans le designer que celui de la version 1 d'Expression Web. A cela, vient s'ajouter l'aide à la saisie des langages tels que XHTML et CSS via les différents schémas. On notera la présence d'un schéma spécifique pour le navigateur IE7.

Intégration de PHP

Enfin ! Expression Web 2 supporte le langage PHP. Expression Web n'a donc presque plus rien à envier à son concurrent Adobe Dreamweaver. Ainsi, Expression Web propose l'IntelliSense dans les fichiers PHP. Il est possible de prévisualiser des sites Web en PHP à l'aide du serveur de développement d'Expression Web 2 et ce, directement via la fenêtre " Design " ou dans le navigateur (fig. 2). Pour cela, il faut installer le client PHP sur sa machine et localiser le php.cgi.exe avec Expression Web 2, si cela n'a pas déjà été fait automatiquement. Plus besoin donc de passer par des logiciels tiers comme EasyPHP pour simplement prévisualiser un site Web en PHP. Néanmoins, tout n'est pas rose, et certaines choses ne sont

pas supportées comme la prévisualisation des variables locales ou des classes. Ce sera donc un réel plus pour les Designers et Intégrateurs, quant aux développeurs PHP, c'est encore une autre histoire...

Les technologies annexes

Tout comme Expression Blend 2, Expression Web 2 supporte désormais Silverlight 1.0, la technologie de Microsoft venue concurrencer Flash dans le domaine des RIA. On notera une nouvelle fonctionnalité très intéressante pour les intégrateurs qui est " l'import de fichier .PSD " vers Expression Web 2. Cela évite, par exemple, de devoir passer d'un logiciel à l'autre pour s'occuper de la découpe d'un template XHTML / CSS et de garder une qualité optimale sur le rendu des images. Expression Web 2 propose de convertir le fichier en .JPEG, .GIF ou encore .PNG, et cela en ayant un aperçu du rendu. Il est possible de sélectionner différentes parties de votre création étant donné que Expression Web prend en compte les différents calques du fichier .PSD. Par contre, la fonctionnalité aura parfois un peu de mal avec

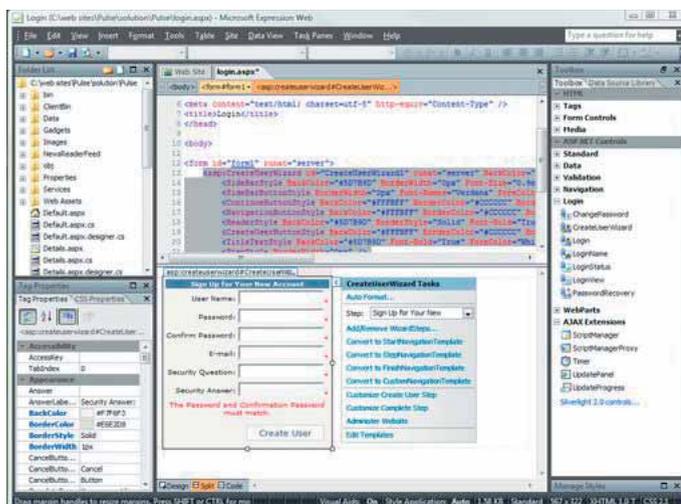


Fig.1

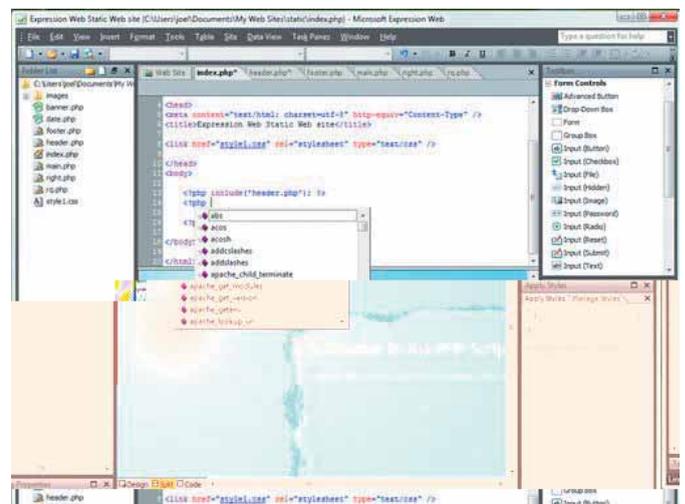
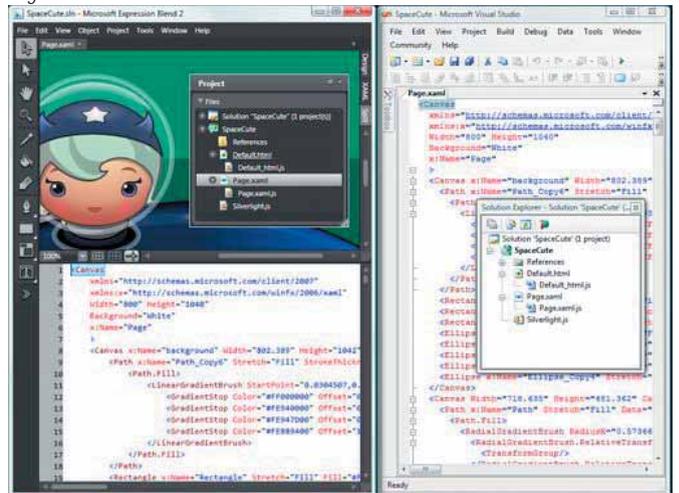


Fig.2

Fig.3



Fig.4



certaines possibilités de Photoshop, comme par exemple la possibilité de baisser l'opacité du fond d'un objet. Cela a un impact seulement si l'on doit sélectionner les calques.

Expression Blend 2

Comme dit plus haut, en plus de supporter Windows Presentation Foundation depuis le début, Expression Blend 2 sait à présent manipuler Silverlight 1.0 (fig. 3) et prochainement Silverlight 2.0 avec la future version 2.5 d'Expression Blend d'ores et déjà disponible en version Preview. Avec la version 2 du logiciel, vient s'ajouter une nouvelle série de "Samples". Nous retrouvons bien entendu les contrôles WPF qu'il est possible de créer et éditer directement depuis la fenêtre "Design" comme par exemple des boutons, des menus, des boutons radios, des checkboxes. Ou en allant encore plus loin, nous avons à notre disposition des treeviews, des sliders, des onglets, ... Pour ce qui est du support de la 3D, il aura pour sa part été revu et donc amélioré.

Prise en charge d'autres langages

A noter qu'il est possible désormais de coder certains langages directement depuis le logiciel. Par exemple, en ouvrant un fichier JS depuis Expression Blend 2, il s'ouvre directement dans celui-ci et non plus dans une fenêtre Bloc-Note sur le côté comme auparavant avec Expression Blend 1. Chose intéressante car elle évite d'avoir une paire de fenêtres encombrante dans la barre des tâches et permet de profiter par la même occasion de la colorisation du script pour une meilleure visibilité du code ! A noter que cette fonctionnalité était déjà présente dans la version Preview de la version 1 mais avait été supprimée lors de la RTM.

Une interface améliorée

Au niveau de l'interface, celle-ci s'adapte aux fonctionnalités suivant le produit, et il est possible de splitter la vue Design/Code comme sur Expression Web 2 (fig. 4). Le dialogue entre les différents logiciels de la gamme Expression est lui aussi amélioré et il est, par exemple, tout à fait possible d'importer des éléments d'Expression Design 2 et aussi bien d'exporter certaines parties vers ce dernier.

Expression Design 2

Lui non plus n'a pas été oublié au niveau des nouveautés. Et on peut déjà prendre en compte le fait que désormais, Expression Design peut importer mais aussi exporter les projets au format Photoshop (.PSD) et .PDF ! Petit bémol à cette fonctionnalité, l'entièreté des calques du fichier .PSD ne sera plus disponible et aura donc fusionné en un seul. Un peu de la même façon qu'en important un fichier .PNG du logiciel Fireworks d'Adobe vers Photoshop. Les calques ne sont plus lisibles entre les 2 logiciels.

Live Effects

A noter l'apparition des Live Effects (fig. 5). Ce sont des effets que l'on applique sur des éléments vectoriels ou non. Comme par exemple le Flou Gaussien, Bevel, Emboss, ... enfin une bonne chose ! Ces effets ne sont pas destructifs envers l'élément et sont entièrement éditables. Il est ainsi tout à fait possible de revenir par la suite sur les effets pour les modifier ou les supprimer, chose que Photoshop ne permet pas avec certains de ses effets.

Interactivité entre logiciels

Au même titre qu'Expression Blend 2, le dialogue a été amélioré entre les deux logiciels. Il

est désormais possible d'importer des parties d'un logiciel à l'autre. Par exemple, en sélectionnant un objet ou un groupe d'objets dans Expression Design 2, en le copiant (CTRL + C) et en le collant (CTRL + V) dans Expression Blend 2, Design le convertira automatiquement en XAML (fig. 6).

Expression Media 2

Anciennement iView Media Pro racheté en 2006 par Microsoft, Expression Media est en quelque sorte l'équivalent de Bridge dans la gamme de produit Adobe (et uniquement outil Expression fonctionnant sous MacOS X). A savoir un logiciel pour gérer, organiser et retrouver vos médias facilement. Dans la version 2 du produit de Microsoft, il y a quand même quelques fonctionnalités qui peuvent être retirées du lot et qui sont plutôt intéressantes dans l'ensemble.

Au menu des nouveautés

On notera, par exemple, la gestion du double moniteur notamment pour l'affichage de la table lumineuse, la possibilité de générer des catalogues partagés, de nouveaux formats de documents pris en charge comme par exemple ceux de la gamme Office 2007, la possibilité de lire les couvertures d'albums des MP3 depuis leurs tags ID3, et enfin la possibilité de sauvegarder un Slide Show en film, ...

Filtre pour l'import

Un nouveau filtre automatique permet à présent de mettre de côté les fichiers corrompus qui pourraient gêner lors d'une opération d'import. L'opération terminée, il est possible de signer chaque fichier automatiquement. Il est aussi possible de générer automatiquement une galerie en Silverlight pour les photos. Expression

Media 2 inclut aussi une fonctionnalité de Virtual Earth permettant de pointer l'emplacement de vos photographies sur une carte du monde si votre photographie contient des informations GPS. Si elle n'en avait pas, il suffit simplement de drag'n droper votre image sur la carte, Expression Media 2 s'occupe alors d'ajouter les informations nécessaires à la photographie et notamment un pointeur sur la carte du monde.

Expression Encoder 2

Le petit nouveau, est comme son nom l'indique un logiciel d'encodage. Auparavant, il n'était pas packagé avec Expression Studio, à présent c'est le cas. Il permet, entre autres, et sans surprise d'encoder vos vidéos pour être diffusées par Silverlight ou par d'autres plateformes (Streaming Vidéo, Zune, ...).

Interaction à la volée

Il est par exemple possible d'ajouter un flux en direct ou encore de mixer entre différents flux, et cela, à la volée, autrement dit, lors du streaming de la vidéo. Pareillement, pour ajouter des animations avec des éléments en vectoriels, gérer les sous-titres ou le chapitrage de la vidéo ou encore la possibilité de générer un player en Silverlight avec tous les contrôles standard. Une gestion des thèmes. n dialogue est possible entre Expression Encoder 2 et Expression Blend 2 si vous souhaitez changer des éléments en XAML ou tout simplement pour changer l'interface en général.

En sauvegardant vos changements, Expression Encoder 2 s'occupera de recharger l'ancien template par le nouveau que vous venez de créer ou modifier à l'aide d'Expression Blend 2.

Migrer d'Adobe à Microsoft

Microsoft étend l'éligibilité des mises à jour vers Expression Web et Expression Media à partir de logiciels proposés par d'Office – Works et d'éditeurs tiers comme Adobe. Tout possesseur d'une des licences éligibles a droit au prix de la mise à jour des produits. Plus de renseignements sur microsoft.com/France/expression/maj.

Période de grâce

Tout achat d'Expression Studio ou d'un produit de la gamme Expression jusqu'au 30 Juin 2008, donne droit à l'obtention gratuite de la Version 2 du même produit acheté. Les détails sur les conditions de la période de grâce seront publiés sur le site Expression. L'enregistrement du produit et une preuve d'achat seront demandés. Plus de renseignements sur microsoft.com/France/expression/maj.

Tarification et disponibilité

La suite complète à partir de la mi-mai au prix public indicatif de vente de 699 \$ (prix de vente sur le marché français non disponible). Par ailleurs, tous les outils à l'exception d'Expression Design 2 sont disponibles à l'achat séparément : Expression Web 2 et Expression Blend 2 sont commercialisés respectivement aux prix publics indicatifs de vente de 299 \$ (prix de vente sur le marché français non communiqué) et 499 \$ (prix de vente sur le marché français non communiqué), et Expression Media 2 et Expression Encoder 2 sont commercialisés au prix public indicatif de vente de 199 \$ (prix de vente sur le marché français non communiqué).

Enfin, Microsoft lancera également l'abonnement Microsoft Expression Professional en juin 2008. Cet abonnement porte sur la suite complète et plusieurs autres programmes* Microsoft, tels que Visual Studio, Windows Vista et Microsoft Office, Visio pour aider les utilisateurs à démarrer au prix exceptionnel de 999 \$ (prix de vente sur le marché français non communiqué). Vous trouverez des informations complémentaires sur cet abonnement à l'adresse <http://www.microsoft.com/expression/subscription>.

Quelques précisions en plus

Il est intéressant de noter qu'avec la sortie d'Expression 2, Microsoft met à votre disposition des forums d'aide sur sa gamme : <http://forums.expression.microsoft.com/>.

Désormais pour les utilisateurs de Mac, Microsoft met à disposition un abonnement permettant d'accéder à la gamme d'Expression Studio 2, ainsi qu'une série de logiciels. Au menu : Expression Studio 2, Visual Studio

2008 Standard, Windows Vista Business, Windows XP SP2, Microsoft Office Standard, Visio Professional, Virtualized Server Environments, Virtual PC et bien sur l'outil qui permettra de faire tourner le tout sans nécessiter Boot-Camp, Parallels Desktop.

Axel Peter

Intégrateur .NET & Designer - Wygwam

<http://www.wygwam.com>

<http://blogs.codes-sources.com/chronos>

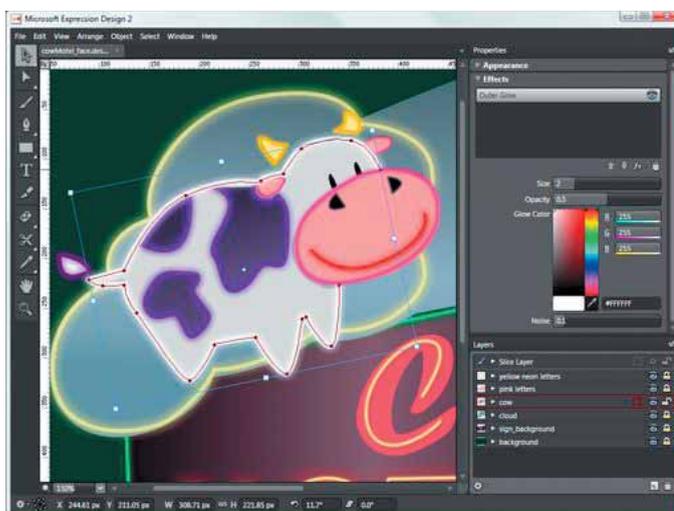
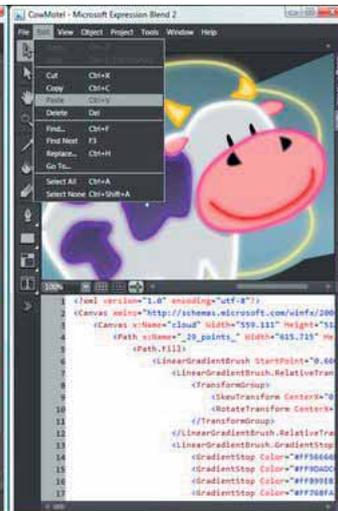


Fig.5



Fig.6



CodeFluent est un produit de génie logiciel qui permet
d'industrialiser la fabrication d'applications professionnelles manipulant des données sur la plate-forme .NET
en automatisant la création des composants à partir d'une modélisation de votre métier.



CODEFLUENT PROCURE LES BÉNÉFICES SUIVANTS

UN GAIN TRÈS SIGNIFICATIF DE PRODUCTIVITÉ

par l'automatisation des tâches répétitives et la mise en œuvre du modèle métier selon une architecture orientée services.

UNE MAINTENABILITÉ ET UNE QUALITÉ ACCRUES

grâce à l'approche intrinsèque de génération qui évite les erreurs et permet d'effectuer des mises à jour sur toutes les couches à l'aide d'une modification centralisée.

LA LIMITATION DU RISQUE

par la structuration du travail des développeurs autour d'une modélisation objet évolutive qui garantit une mise en œuvre selon les meilleures pratiques d'implémentation SOA sous plate-forme Microsoft.

L'ÉVOLUTIVITÉ DE L'APPLICATION

car la prise en charge de nouvelles technologies et de nouvelles versions de plate-forme est assurée par la mise à disposition régulière de nouveaux producteurs (Windows Presentation Foundation, Office Business Applications, mobilité).

LES OFFRES CODEFLUENT

GRATUIT

499€ HT

2490€ HT

VERSION EXPRESS
génération limitée à 10 entités

VERSION PROFESSIONNELLE
génération limitée à 20 entités

VERSION ENTREPRISE
génération illimitée

Pour commander votre licence, envoyez un email à sales@softfluent.com

Echangez vos données grâce à JSON

Le format d'échange de données le plus communément utilisé dans les applications utilisant AJAX est XML. Normal, me direz vous, de par la définition première d'AJAX (Asynchronous JavaScript and XML). Mais vous en conviendrez, sa mise en place n'est pas des plus aisée.

Pourquoi ne pas se passer du fichier XML, en utilisant un autre format d'échange de données plus approprié à JavaScript ? C'est maintenant possible grâce à JSON (JavaScript Object Notation).

Vous avez dit JSON ?

De nombreuses définitions de JSON sont disponibles sur divers sites internet, mais la plupart font référence aux définitions du site officiel (www.json.org). Après lecture des informations contenue sur le site, je vous avoue que l'on n'est pas du tout convaincu et leur compréhension demande souvent une relecture. Afin de simplifier tout ceci, je vous propose d'illustrer ces concepts théoriques à travers un exemple concret qui énoncera les concepts de JSON et vous permettra de comprendre sa mise en œuvre.

Comprendre la structure de JSON

L'application que nous allons développer est un cas d'école consistant à lister les contacts présents dans un carnet d'adresses. Il convient tout d'abord de réaliser notre fichier JSON. (carnet.json) et d'y insérer les lignes suivantes :

```
{
  "contacts":[
    {"prenom":"Aurelien","telephone":"03.00.00.00.00"},
    {"prenom":"Delphine","telephone":"03.00.10.00.00"},
    {"prenom":"Alexis","telephone":"08.00.00.41.00"},
    {"prenom":"Nicole","telephone":"04.00.20.00.00"},
  ]
}
```

Quelques explications s'imposent.

Nous créons tout d'abord ce que l'on appelle dans le jargon de JSON un objet. Un objet commence par une accolade à gauche ({) et se termine par une accolade à droite (}). Selon JSON, un objet porte un nom et fait référence à une valeur, l'ensemble formant un couple nom/valeur. Noté " nom " : " valeur " .

- Une valeur peut être :
- Un objet,
- Un tableau,
- Une chaîne de caractères,
- Un nombre,
- Les valeurs true, false ou null.

Dans notre cas, l'objet principal créé a pour nom " contacts " et a pour type de valeur un tableau délimité par les caractères [et] .

Ce tableau contenant lui-même un ensemble de valeurs séparées par une virgule. Si vous avez bien suivi, ces valeurs sont elles-mêmes des objets étant donné que ce sont des couples nom/valeur commençant par une accolade gauche et se terminant par une accolade droite. Vous savez tout ou presque sur la structure d'un fichier JSON. On se rend compte à quel point cela est très facile à réaliser, une fois les concepts de base assimilés. Voyons maintenant comment utiliser ce fichier à l'aide d'AJAX.

La page HTML

Créons une page index.html contenant le code suivant :

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

<title>Mon carnet d'adresses</title>
</head>

<script src="javascript.js"></script>

<body onLoad="listerContacts()">

<div id="zone_de_resultat">

</div>

</body>
</html>
```

Cette page fera appel à un fichier javascript.js contenant la fonction listerContacts() qui sera appelée dès l'affichage de la page. A noter également la présence d'une zone de résultat qui accueillera la liste des contacts présents dans le carnet d'adresses.

Le fichier JAVASCRIPT

```
Fonction listerContacts()

function listerContacts()
{
  //Création de l'objet XMLHttpRequest
  var xhr = null;

  if(window.XMLHttpRequest)
  {
    xhr = new XMLHttpRequest();
  }
}
```

```

else
{
    if(window.ActiveXObject)
    {
        try
        {
            xhr = new ActiveXObject("Msxml2.XMLHTTP");
        }
        catch (e)
        {
            xhr = new ActiveXObject("Microsoft.XMLHTTP");
        }
    }
    else
    {
        alert("Votre navigateur ne supporte pas les objets XMLHttpRequest...");
        xhr = false;
    }
}

//Chargement et traitement du fichier JSON
xhr.open("GET", "carnet.json", true);
xhr.onreadystatechange=function(){ traitementJSON(xhr);
xhr.send(null);
}
}

```

Je ne reviens pas sur les fonctions AJAX que vous devez certainement connaître. Je porte juste votre attention sur le fait que le chargement du fichier JSON s'effectue de la même façon qu'un fichier XML classique.

Traitement du fichier JSON

```

function traitementJSON(xhr)
{
    if (xhr.readyState == 4)
    {
        if (xhr.status != 404)
        {
            var donnees = eval("(" + xhr.responseText + ")");

            //Parcours et mise en forme de l'objet contact
            var codeHtml = "<table width='274' border='1'>"
            codeHtml += "<tr>";
            codeHtml += "<th width='134' scope='col'>Prenom</th>"
            codeHtml += "<th width='124' scope='col'>Telephone</th>"
            codeHtml += "</tr>";

            for (i=0 ; i<donnees.contacts.length ; i++)
            {
                codeHtml += "<tr>";
                codeHtml += "<td>" + donnees.contacts[i].prenom + "</td>";

```

```

                codeHtml += "<td>" + donnees.contacts[i].telephone + "</td>";
                codeHtml += "</tr>";
            }
            codeHtml += "</table>";

            //Affichage du resultat
            document.getElementById("zone_de_resultat").innerHTML = codeHtml;
        }
    }
}
}
}

```

La méthode `eval()` invoque le compilateur Javascript qui va se charger de créer une structure de type objet. A partir de cet instant, il nous est facile de manipuler l'objet créé pour en extraire les informations souhaitées. Pour récupérer la valeur " prénom " du premier contact, il suffit de l'extraire de cette façon :

`Donnees.contacts[0].prenom`

Ce petit morceau de code placé dans une boucle accompagnée d'une mise en forme html, permet d'obtenir le résultat illustré par la figure n°1 lors du chargement de page.

Prenom	Telephone
Aurelien	03 00 00 00 00
Delphine	03 00 10 00 00
Alexis	08 00 00 41 00
Nicole	04 00 20 00 00

JSON exclusivité JavaScript ?

Et bien non ! JSON peut être utilisé dans de nombreux langages : C, C++, ActionScript, Java, Python, Lisp, PHP et bien d'autres ! PHP, dans sa version 5.2 inclut les fonctions `json_encode()` et `json_decode()` permettant de manipuler les fichiers JSON.

Conclusion

Comme vous avez certainement pu le constater tout au long de cet article, la mise en place de JSON est on ne peut plus simple. Tout comme son homologue XML, il est au format texte, offre une structure et peut être porté sur de nombreux langages. Le format de JSON étant plus léger que XML, cela permet de faire un gain de ressource lors de son traitement. Son intégration avec JavaScript fait de JSON le format d'échange privilégié lors de l'utilisation de la technologie AJAX. XML, quant à lui, offre une présentation des données beaucoup plus agréable et contrairement à JSON, il convient mieux pour un traitement de données plus importantes. Malgré cela, JSON mérite à être connu et utilisé. XML ou JSON ? A vous de vous forger votre propre opinion et de faire votre choix !



■ Aurélien Vannieuwenhuyze

Chef de projet - Association Outcomerea (www.outcomerea.org) Promotion et développement de la recherche et de l'enseignement en Réanimation
 Blog : aurelienv.no-ip.org

La prochaine génération de navigateurs au banc d'essai

Avec l'arrivée annoncée de nouveaux standards web (Javascript 2, HTML5), il est plus que jamais temps de s'intéresser à la prochaine génération de navigateurs. Supporteront-ils toutes les fonctionnalités promises par le W3C ? Qu'apporteront-ils de plus aux utilisateurs finaux ? Faciliteront-ils enfin la vie du développeur web ?

Les forces en présence

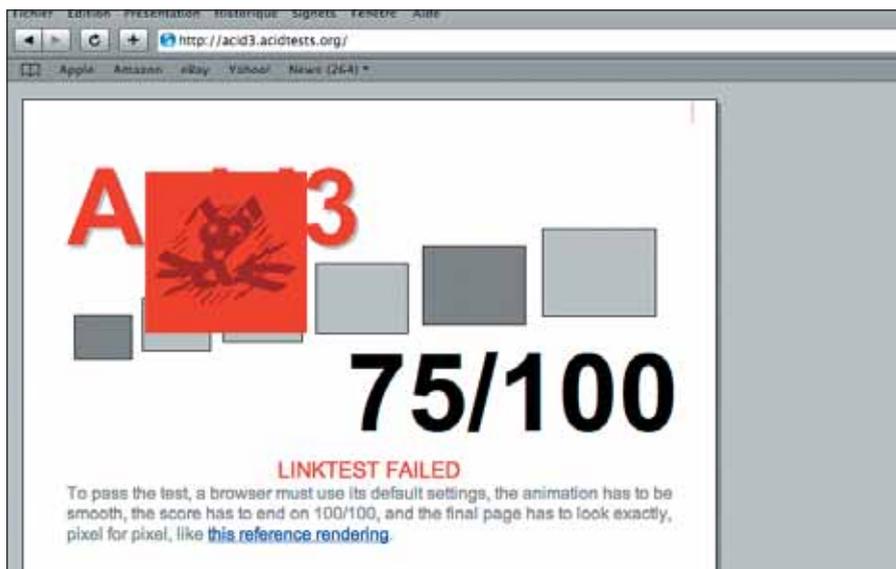
Internet Explorer 8 bêta

Internet Explorer, malgré toutes les critiques qui ont pu lui être adressées reste le navigateur leader sur le marché, et de très loin. Pour l'utilisateur lambda, Internet Explorer 7 a certes comblé le retard accumulé par la version précédente, mais elle n'a guère arrangé le sort des développeurs web. En effet, Internet Explorer 6 n'a pas disparu aussi vite qu'on l'espérait, et les incompatibilités qui lui sont si particulières restent plus que jamais d'actualité. D'autre part, si Internet Explorer 7 a considérablement rapproché sa gestion des styles des standards, son implémentation de Javascript restait encore largement perfectible.

Internet Explorer 8 faisait donc face à un double problème : ne pas casser le mode de rendu d'IE7 par rapport aux sites existants, tout en se rapprochant des standards. Pour satisfaire ces aspirations contradictoires, les concepteurs d'IE8 ont utilisé une méthode peu subtile mais efficace : le nouveau navigateur embarque trois moteurs : le premier en mode "standards", le second en "compatibilité IE7" et le dernier pour maintenir la compatibilité avec les versions précédentes. L'effort de Microsoft pour faciliter la compatibilité entre navigateurs est donc bien réel... nous en verrons plus loin les résultats. Soulignons enfin qu'IE8 fait plus que suivre les autres navigateurs puisque quelques innovations nous sont proposées. Nous verrons par exemple ce qu'il en est des "webslices" qui pourraient devenir un complément tout à fait pertinent de la syndication RSS.

Firefox 3 bêta 5 (maintenant en RC)

Le projet "Gran Paradiso" touche au but puisque la première release officielle de Firefox 3 devrait nous arriver cet été. Navigateur très prisé des geeks, et en particulier des développeurs Ajax, du fait de sa fameuse



extension "Firebug", Firefox a su dépasser ce public pour conquérir une part de marché d'un peu plus de 20%, notamment grâce à ses régulières innovations. Or, cette fois-ci, au delà des nouvelles fonctionnalités, c'est surtout une moindre consommation de mémoire et plus de stabilité que les utilisateurs attendaient de cette nouvelle mouture. Il faut dire que nous sommes à une époque où les navigateurs sont confrontés à des contraintes considérables par rapport au passé : les utilisateurs ouvrent de multiples onglets, restent connectés de nombreuses heures, vont sur des sites utilisant intensivement Javascript... bref, la gestion de la mémoire et des erreurs devient véritablement critiques. Comme nous le verrons, Firefox 3 améliore ses performances et continue à innover... mais cela sera-t-il suffisant à l'heure où la guerre des navigateurs fait de nouveau rage ?

Safari 3.1 bêta

Apple a étonné tout son monde en sortant l'année dernière une version Windows de son navigateur-phare et enfonce le clou avec cette

version 3.1. La société de Steve Jobs avait déjà infligé une très grosse claque à l'univers du mobile avec sa version iPhone du navigateur qui laisse encore aujourd'hui loin derrière tous les concurrents. Cela est bien entendu révélateur du dynamisme actuel de Safari mais aussi et surtout de celui de Webkit, le moteur Open-source sur lequel Safari repose. Bien plus que Firefox, Webkit (utilisé par Trolltech, Nokia, Android, etc. ndr) a su attirer les concepteurs de mobiles voulant développer un navigateur sur une base open source, ce qui en fait aujourd'hui un projet très suivi. Safari 3.1 s'est rapidement fait une renommée dans le petit monde du web avec son support de fonctionnalités HTML5 d'avenir (comme la base SQLite embarquée) ou son succès sur les tests ACID3. Des atouts qui ne seront certainement pas de trop pour exister un peu plus entre Internet Explorer et Firefox.

Opera 9.5 bêta

Si sa part de marché sur PC est symbolique, c'est ailleurs qu'Opera marque des points : sur mobile et PDA avec Opera Mobile et Opera



Mini ou encore sur les consoles Nintendo DS et Wii. Opera possède pourtant bien des atouts qui auraient pu lui donner une place plus importante dans le monde des navigateurs... s'il avait disposé de la puissance marketing de ses concurrents.

Opéra est un produit audacieux et original : il est par exemple le seul à combiner navigateur web et client de messagerie comme au bon vieux temps de Netscape... mais il y ajoute également un client Bittorrent de manière native, ce qui en fait un outil polyvalent à l'usage des mordus d'Internet.

Le projet Kestrel, nom de code d'Opera 9.5, sortira de sa période bêta durant l'été avec une vague d'innovations prometteuses, en attendant l'aboutissement du projet Peregrine, alias Opera 10.

Le benchmark " end-user "

Ergonomie

Il est bien entendu difficile d'évaluer de manière réellement objective l'ergonomie des navigateurs en compétition tant les habitudes des utilisateurs comptent dans la productivité et le confort d'utilisation. Néanmoins, les attentes ergonomiques de base vis-à-vis des " butineurs " sont assez simples : parvenir rapidement aux sites favoris, chercher facilement de nouvelles ressources, savoir aisément où l'on est et où l'on peut aller. Les 4 navigateurs passés au banc d'essai ne s'y sont pas trompés et l'on retrouve chez eux quelques constantes : interfaces dépouillées, moteur de recherche intégré et accès en un clic aux flux RSS par exemple.

Opéra est depuis quelque temps déjà le plus audacieux en la matière : les " mouse ges-

tures ", comprenez la possibilité de naviguer simplement à partir de mouvements de souris peuvent se révéler terriblement addictifs pour les utilisateurs qui sauront les maîtriser. Plus simple, mais tout aussi efficace, le système " Speed Dial ", vous donne accès à une page contenant des liens vers vos sites préférés dès l'ouverture d'un nouvel onglet vide. On pourra au passage apprécier la visualisation d'une miniature du contenu d'un onglet lors de son survol... très pratique pour ceux qui comme moi ne savent pas naviguer avec moins de 10 onglets ouverts à la fois ! Enfin, la recherche " full text " au sein de l'historique du navigateur est une belle idée parfaitement exécutée. Seul bémol pour Opéra : l'affichage étrange de certains sites optimisés pour les principaux navigateurs du marché.

Firefox 3 progresse indéniablement par rapport à son prédécesseur, en particulier grâce à deux nouveautés très séduisantes. La première est " le bookmark en un clic ", grâce à une petite étoile intuitive placée dans la barre d'adresse, qui offre un gain de temps considérable par rapport à l'habituel accès par le menu " marque-page ". La seconde est le système de suggestion automatique des adresses, très agréable : l'essayer, c'est l'adopter.

L'interface d'Internet Explorer 8, dans la droite lignée de la version 7, est épurée, bien pensée et met en évidence l'essentiel. Au niveau de la barre d'adresse par exemple, le nom de domaine sur lequel vous vous trouvez est désormais écrit d'une couleur plus sombre que le reste de l'url afin de vous retrouver plus facilement lors de vos longs périples de " surf " intensif. Mais si du point de vue ergonomique le bilan est plutôt positif pour IE8, on peut se

demander quel impact aura la possibilité de changer de mode de rendu (" ie7 " ou standard ") sur des utilisateurs lambda : cela ne risque-t-il pas de semer la confusion chez eux ? Enfin, Safari 3.1 est une relative déception sur le plan ergonomique, ce qui est plutôt inhabituel chez Apple. En effet, si le navigateur ne comporte pas de véritable défaut, il n'évolue pas non plus franchement... contrairement à ses concurrents ! Résultat : son interface paraît terne et peu innovante en regard des autres futurs butineurs du marché.

Performances

Avec l'augmentation des débits, les pages envoyées par nos sites préférés se sont considérablement enrichies... au point que la vitesse de rendu devient souvent plus problématique que les temps de téléchargement des pages. De même, l'avènement d'Ajax et le recours grandissant à Javascript sur des opérations de plus en plus complexes, donnent aux utilisateurs des exigences supplémentaires à la prochaine génération de navigateurs. Les différentes " écuries " le savent bien et clament chacune de leur côté les meilleures performances avec des tests plus ou moins valables. Mais avant d'aller plus loin, il me faut vous adresser un avertissement par rapport à tous ces benchmarks qui comportent, tous sans exception, des failles vicieuses.

Premier problème, celui de la représentativité : les tests sont souvent exécutés sur des batteries d'opération Javascript intensives sans grand rapport avec les exigences des sites courants. Second problème, celui de la mesure : l'événement " onLoad " qui sert de base à la plupart des tests de vitesse de chargement des pages n'est pas déclenché au même moment chez tous les navigateurs. De plus, l'objet Javascript Date semble aussi avoir un comportement inconsistant entre les navigateurs. Ces deux éléments font que Safari " explose " souvent la concurrence sur nombre de benchmarks populaires alors que sa vitesse d'affichage ne semble pas à ce point supérieure. Enfin, après moult tests et malgré les doutes qui entourent la mesure de ses performances, Safari n'a semble-t-il pas usurpé son titre de navigateur plus rapide... Suivent ensuite Opéra 9.5 et Firefox 3 dans des temps assez comparables (Opera chargeant un peu plus vite les pages, Firefox exécutant mieux Javascript), tandis qu'Internet Explorer 8 récolte les moins bons résultats, ce qui n'est guère

Développement Web

étonnant, tant le prochain navigateur de Microsoft est loin de sa sortie officielle comparative à ses concurrents.

Les fonctionnalités innovantes

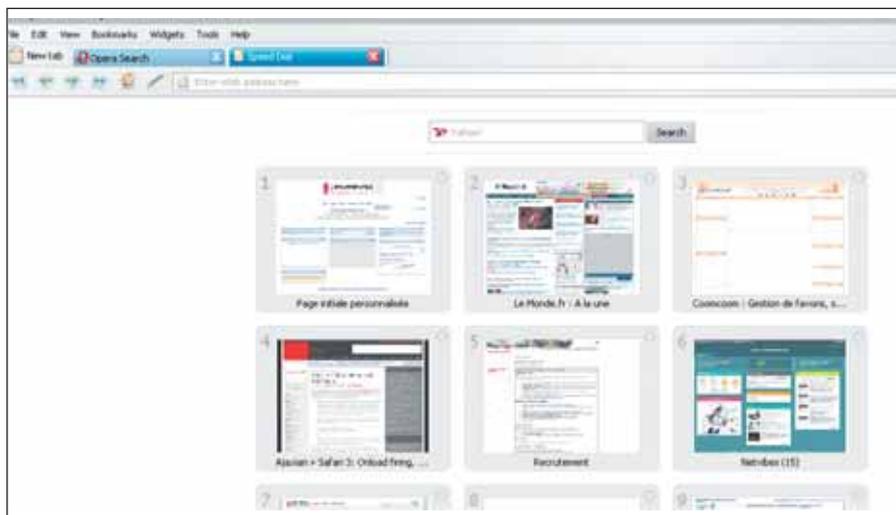
Opera se fait remarquer dans cette nouvelle mouture en tirant parti de son implantation sur mobile pour proposer le service " Opera Link " : il s'agit de synchroniser toutes propriétés de ses différentes instances d'Opera, que ce soit depuis un PC, un PDA ou un SmartPhone.

Si les releases de Safari et de Firefox sont de qualité, force est de constater qu'elles n'apportent pas véritablement de nouvelles fonctionnalités. En effet, de ce point de vue, les meilleures surprises de cette nouvelle génération de navigateur viennent probablement d'Internet Explorer 8. Première fonctionnalité, les webslices sont une nouvelle forme de flux RSS : ils vous permettent de vous abonner à un morceau de page afin de suivre ses mises à jour. La seconde nouveauté, nommée " Activities " est tout aussi excitante : elle offre de nouveaux services contextuels directement au sein du navigateur. On peut par exemple imaginer de rechercher dans wikipedia un mot trouvé dans une page d'un simple clic droit ou encore de trouver sur Amazon la référence d'un livre cité par votre blog préféré.

Fluidité et stabilité

Internet Explorer 8 reprend le concept de " crash recovery " introduit par Firefox afin de rendre moins pénible pour l'utilisateur les crashes de son navigateur et de la perte de l'ensemble des onglets ouverts par ce dernier. Internet Explorer 8 se base de plus sur l'architecture " Loosely Coupled Internet Explorer " afin d'isoler chaque onglet dans un processus particulier pour éviter que tout le navigateur ne plante ou ne soit bloqué par un onglet défectueux, un mal bien connu des utilisateurs de Firefox 2. Evidemment, version bêta oblige, le navigateur Microsoft connaît encore quelques soucis, mais sa stabilité est globalement bonne par rapport à la maturité du produit.

S'il était un point où Firefox 3 était attendu, c'était bien sur la stabilité. Les développeurs de la communauté Mozilla ont travaillé dur pour revoir en profondeur la gestion des ressources mémoire de la machine et leurs efforts n'ont pas été vains. Les utilisateurs peuvent en effet être rassurés : la frustrante sensation d'inertie que donnait souvent Firefox 2 s'est envolée. De manière générale, tout est plus fluide qu'avant, en particulier lors d'une utilisation



prolongée, tandis que les crashes purs et simples, s'ils n'ont pas disparus, sont nettement moins nombreux qu'ils ne l'étaient jusque là. De manière assez inattendue, Opera et Internet Explorer se distinguent par leurs empreintes mémoire assez réduites en utilisation... une caractéristique que l'on espère conserver lors de leur passage en version finale ! Safari, enfin, corrige dans cette version les principaux problèmes de stabilité rencontrés sur plate-forme Windows tout en restant remarquablement efficace sous MacOS X.

Conclusion " end-user "

Si Opera 9.5 et Firefox 3 semblent être à ce jour les navigateurs les plus efficaces pour l'utilisateur lambda, c'est peut-être d'abord parce qu'ils sont les plus matures des navigateurs à venir. Safari 3.1, en particulier sur Windows qui a servi de système étalon pour ce benchmark, est encore très récent, tandis qu'Internet Explorer 8 en est véritablement à ses balbutiements. Quoi qu'il en soit, chacun de ces navigateurs présente un véritable intérêt pour les utilisateurs à l'avenir : Safari pour ses performances, Firefox pour son ergono-

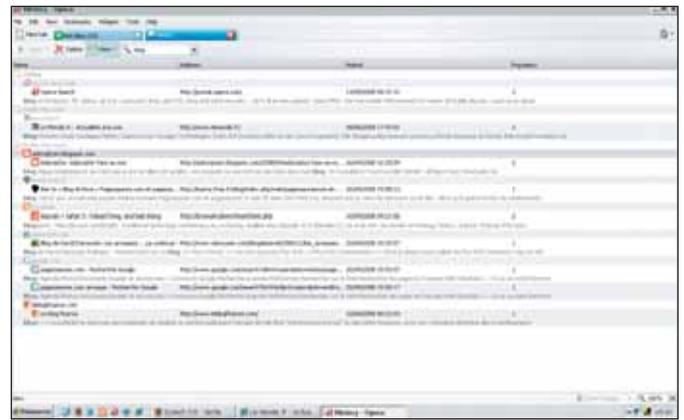
mie, Opera pour ses idées nouvelles et Internet Explorer pour ses innovations conceptuelles et techniques.

Etat des lieux à l'attention du développeur web

Si l'utilisateur final ne s'intéresse qu'aux aspects les plus opérationnels du navigateur, le développeur est confronté à des aspects plus techniques : respect des standards W3C afin d'éviter les problèmes de compatibilité, implémentation des spécifications de la prochaine norme du principal langage du web, HTML5, et bien entendu disponibilité d'outils de développement pour la mise en oeuvre de DHTML avancé.

Respect des standards

Le test ACID 2, qui date d'avril 2005, a pour objectif de contrôler le respect d'un certain nombre de spécifications HTML et CSS 2 par les navigateurs. Ce test, bien qu'il ne soit pas exhaustif par rapport aux spécifications concernées, donne une bonne idée du niveau d'implémentation des normes définies par le W3C. Depuis mars 2008 et la sortie de la bêta



d'Internet Explorer 8, on peut quasiment considérer ce test comme réussi par l'ensemble des navigateurs dont il est ici question, même si un point de détail fait encore débat au sujet d'Internet Explorer. Tous les regards se tournent désormais vers la troisième génération de tests, ACID3. Ce dernier va beaucoup plus loin que son aîné puisqu'en plus de CSS 2 et HTML, sont testés des aspects de :

- DOM, le modèle objet des documents HTML,
- CSS 3, qui bien qu'encore en travaux au W3C pour une bonne partie, complète CSS1 et 2
- EcmaScript 3, qui sert notamment de base à Javascript et à JScript
- SVG, une norme de dessin vectoriel basée sur XML
- SMIL, une norme destinée aux animations multimédia basée elle aussi sur XML
- HTTP et URI

Les versions téléchargées au début du mois d'avril de chacune des versions Windows de nos navigateurs donnent les résultats suivants :

- Safari : 74/100
- Firefox : 71/100
- Opera : 61/100
- Internet Explorer : 19/100

Ces scores sont toutefois à modérer puisque certaines nightly builds de Webkit et d'Opera atteignent déjà les 100/100... difficile de dire néanmoins quand ces modifications pourront être intégrées aux versions officielles, mais une chose est sûre : les progrès sont là.

Implémentation HTML5

HTML5 s'apprête à faire faire au web un bond technologique majeur : la spécification actuelle contient en effet tous les ingrédients pour enrichir nos pages de manière décisive. Imaginez un peu notre bon vieux web doté de possibilités de dessin dynamique 2D voire 3D avec la balise canvas, de contenus multimédia natifs avec des balises audio et video, d'API

pour faire fonctionner les applications web hors ligne... et bien HTML5, c'est tout cela et plus encore !

On comprend dès lors pourquoi ces implémentations sont capitales pour les développeurs que nous sommes. Pour le moment, les navigateurs se lancent dans cette spécification en ordre dispersé. Tour d'horizon :

- **Canvas** (dessin 2D) : la balise est désormais supportée par Firefox, Safari et Opera
- **Multimedia** : Safari supporte les balises audio et video (version MacOS uniquement), Opera uniquement la balise Audio
- **API Cross document messaging** (dialogue entre plusieurs pages) : l'API est implémentée dans Firefox, Opera, Internet Explorer et Safari.
- **API drag'n drop** : présente partiellement dans Safari et Internet Explorer
- **API offline** (détection du passage en online/offline) : implémentée dans Firefox 3 et IE8
- **Stockage offline** : dans Firefox et Internet Explorer sous forme de " super-cookies ", dans Safari sous forme d'une base SQLite embarquée dans le navigateur
- **Nouvelles balises de formulaires (Web form 2.0)** : implémentées dans Opera

Bien entendu, il reste encore beaucoup de travail par rapport à l'ensemble des spécifications d'HTML5. Néanmoins, le fait que l'ensemble des éditeurs de navigateurs prennent déjà des initiatives sur des aspects variés de la norme est de bon augure pour la suite.

Outils de développement

Jusqu'à présent le couteau suisse du développeur web s'appelait *Firebug*, une extension devenue incontournable du fait de sa puissance : débogueur Javascript, moniteur de requête AJAX, explorateur DOM, inspecteur CSS, moniteur réseau... bref, rien ne manque pour régler les problèmes observables sur Firefox. Et c'est bien là que le bât blesse puisque la ques-



tion des bugs spécifiques aux autres navigateurs reste entière. Si Firebug Lite, une librairie Javascript, offrait jusque là une solution de dépannage intéressante, c'est avec le plus grand des plaisirs que tous les développeurs Ajax accueilleront le nouvel outil de débogage intégré à Internet Explorer 8. Les ingénieurs de Microsoft ont en effet l'excellente idée de cloner Firebug, si bien que le débogage du Javascript nous devient désormais accessible sur toutes les versions d'Internet Explorer, grâce à la faculté d'IE8 de basculer sur les moteurs de rendu des navigateurs précédents. Côté Safari, Drosera, le debugger Javascript de Webkit, devient maintenant disponible pour les versions Windows du navigateur. Opera faisait jusqu'à récemment figure de parent pauvre du développement Javascript, mais l'arrivée début mai de " DragonFly " a changé la donne : le débogage des interfaces web est désormais un standard de fait des navigateurs du marché.

And the winner is...

Si Firefox 3 reste le meilleur ami du développeur encore pour quelque temps, les progrès accomplis par IE8, Opera et Safari devraient nous permettre d'être plus efficaces et plus ambitieux pour l'avenir. Et au regard des premiers pas d'HTML5 dans ces navigateurs, il nous est plus que jamais permis de rêver...

■ **Jean-Baptiste Boisseau**
Eutech SSII

Introduction à la programmation Qt

Qt est une librairie de programmation graphique écrite en C++. Totalement Open Source elle est disponible sur les principales plates-formes du marché, de l'AIX au téléphone portable en passant par l'ordinateur de bureau et le PDA !

Cette bibliothèque est écrite avec une préoccupation majeure de portabilité. Qt est, en effet, multi-plate-forme par simple recompilation. Ceci fait gagner un temps précieux lors du développement d'applications multi-plates-formes. Google ne s'y est pas trompé lorsque le choix de Qt fut fait pour le développement de Google Earth. Dans cette série de 3 articles, nous allons développer une application complète avec Qt 4 en C++. Si le lecteur n'est pas familier de ce langage ce n'est pas réhébitoraire. En effet, il est possible de développer avec Qt dans quasiment n'importe quel langage. Trolltech propose Jambi, la version Java de son framework, la communauté quant à elle propose des binding pour les principaux langages de scripts tels que Perl (Perl Qt 4), Python (PyQt4) ou encore Ruby. Les concepts sont les mêmes pour tous ces langages et les 3 articles de cette série sont très facilement transposables.

Au commencement...

... Était le cahier des charges.

Dans un premier temps, donc, parlons de l'application que nous allons développer. Nommée QuickDesktopNotes, celle-ci permet une prise de notes rapide sur le bureau via un icône dans le "system tray". Le but est d'avoir une application à la fois rapide et peu intrusive en terme d'occupation du bureau. Ajoutons au cahier des charges la nécessité de gérer les notes multiples, la sauvegarde de ces notes sous forme d'un flux RSS (pour l'interopérabilité) ainsi qu'une fenêtre de configuration pour gérer l'application.

En résumés nous implémenterons 2 fenêtres communicantes, un " tray icon " et un parser RSS. Le tout en moins de 500 lignes de code.

QuickDesktopNotes va nous permettre de nous familiariser avec les principaux concepts de programmation Qt (héritage, signaux/slots, etc.), les outils de développement (designer, assistant, qmake, etc.) et la programmation d'IHM (Interface Hommes Machines).

Quelques petites remarques avant d'entrer dans le vif du sujet :

- QuickDesktopNotes a été développé en partie sous Linux, en partie sous Windows XP.
- En de (très) nombreuses occasions il sera fait référence à la (volumineuse) documentation de Qt. Celle-ci est livrée avec Qt et est accessible via le logiciel "assistant", qui permet une navigation simple et rapide dans l'API, les exemples et autres tutoriels disponibles.
- Pour la gestion de projet ainsi que pour la compilation de l'application, un shell est nécessaire, assurez-vous donc de posséder cet outil avant de commencer.

PNoteEditor : la fenêtre d'édition des notes.

Dans un premier temps, nous nous occuperons de la création de la fenêtre d'édition des notes. Ceci se fera selon le processus classique : nous allons dessiner la fenêtre dans le designer, puis hériter de la classe générée afin d'implémenter les fonctionnalités de notre fenêtre. En effet, nous ne voulons pas perdre de temps à réinventer la roue, nous

souhaitons par contre profiter de toutes les fonctionnalités d'une fenêtre d'édition classique (copier/coller, drag'n'drop, etc.). Commencez par créer un répertoire QuickDesktopNotes. Puis lancez le designer, vous devriez vous trouver en face de la fenêtre suivante : (Fig.1)

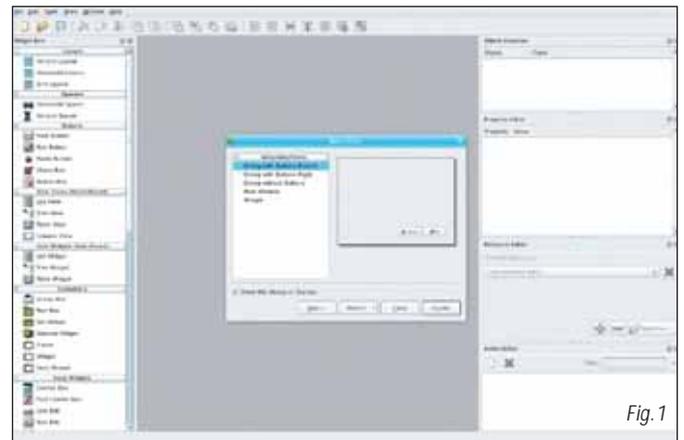


Fig.1



Fig.2

Choisissez de créer un nouveau widget en sélectionnant le template " widget ". Designer crée alors un formulaire vide de type QWidget. Modifiez sa propriété objectName pour la positionner à " PNoteEditor ". (Fig.2)

Ajoutez un QLineEdit au formulaire vide. Ceci se fait en glissant l'élément désiré depuis la " widget box " vers le formulaire. Modifiez ensuite la propriété " objectName " pour affecter le nom " title " à la place de " lineEdit ". Cet élément servira, comme son nom le laisse supposer, à saisir le titre d'une note. Cela sera utile pour identifier une note précise par la suite. (Fig.3)

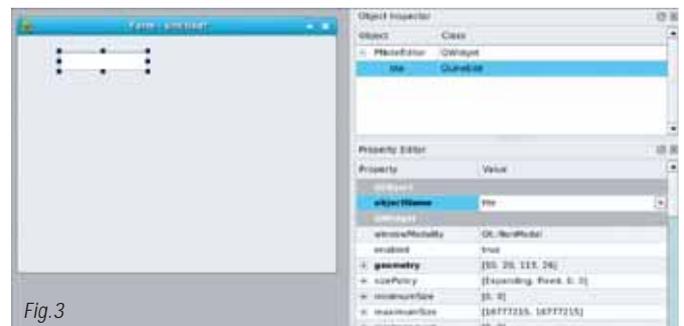


Fig.3

Il faut ensuite ajouter une zone de saisie de texte. Cette fonctionnalité est prise en charge par le widget QTextEdit, que l'on ajoutera de la même façon que précédemment. Nommez ce dernier " editor ". (Fig.4)

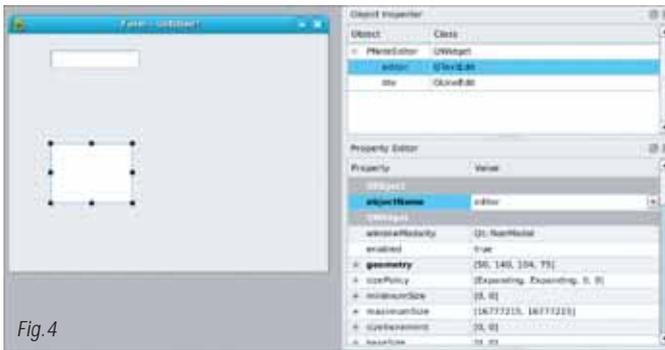


Fig.4

Une fois ces 2 éléments mis en place il reste à s'assurer d'une disposition dynamique des 2 éléments graphiques à l'intérieur de notre formulaire. Le mécanisme proposé par Qt pour gérer les dispositions est nommé QLayout. Il se décline en différentes versions comme les dispositions " box " ou " grid " (QBoxLayout ou QGridLayout). Le principe général des " layout " est basé sur la taille optimale d'un widget. En effet, chaque widget retourne une taille qu'il considère comme optimale pour lui-même (détenue par la propriété sizeHint). Les objets layout se servent de cette taille pour disposer les éléments graphiques sur le formulaire. Dans le cas de notre fenêtre, nous choisirons une disposition sur grille. Pour cela, sélectionnez le fond du widget (ni le QLineEdit ni le QTextEdit), puis cliquez sur l'icône " Layout in a Grid " dans la barre d'outils. Cela va disposer automatiquement les 2 éléments de façon optimale. (Fig.5)

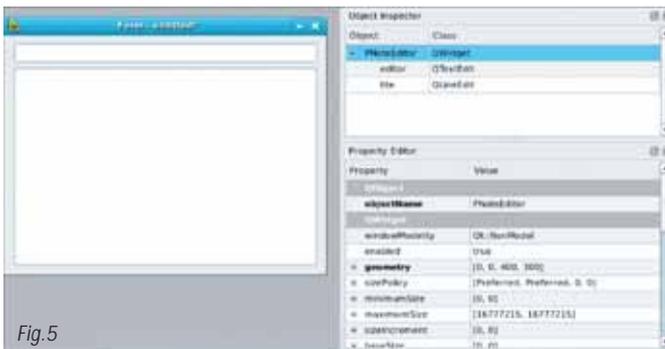


Fig.5

Il reste désormais à sauvegarder le formulaire dans le répertoire Quick-DesktopNotes sous le nom *pnoteeditor.ui*. Les fichiers .ui sont automatiquement compilés par l'uic (User Interface Compiler) en un fichier d'en-tête (.h) préfixé par ui_ (et déclarant l'espace de nom Ui::<object-Name>, ici Ui::PNoteEditor), ce fichier contient toutes les définitions de code en ligne. Nous ne serons pas contraints d'appeler uic nous-mêmes car cela sera effectué automatiquement à la compilation. En l'état notre fenêtre est utilisable, il suffirait de s'en servir dans un programme avec un QWidget comme base, d'instancier la classe Ui::PNoteEditor généré par uic, puis d'appeler la méthode setupUi() sur le QWidget précédemment instancié. Le code ressemblerait à cela :

```
QWidget *widget = new QWidget;
Ui::PNoteEditor ui;
ui.setupUi(widget);
```

Cependant, bien que la fenêtre s'affiche et que les fonctionnalités d'édition de QTextEdit et QLineEdit soient bien présentes, il manque certains mécanismes que nous désirons avoir à disposition.

Premièrement, il serait très utile que les 2 éléments de saisie signalent les modifications qui sont apportées à leur contenu. Il est aussi souhaitable de fournir des accesseurs qui permettront de retirer ou de positionner les valeurs des 2 zones de saisies. En effet, cette fenêtre ne sera pas utilisée comme plusieurs éléments graphiques séparés mais comme un widget unique.

Nous allons donc mettre en place une solution plus évoluée que la simple instanciation d'un formulaire : nous allons créer une classe PNoteEditor qui va hériter de Ui::PNoteEditor.

Afin de fournir aussi les possibilités des widgets Qt, nous allons également hériter (publiquement) de QWidget. L'héritage de Ui::PNoteEditor sera lui privé, car nous ne désirons pas que la classe soit accédée autrement que par les accesseurs que nous allons proposer.

PNoteEditor : le code derrière la fenêtre.

Voyons dans un premier temps le fichier d'en-tête contenant la définition de la classe :

```
#ifndef PNOTEEDITOR_H
#define PNOTEEDITOR_H

#include <QWidget>
#include "ui_pnoteeditor.h"

class PNoteEditor : public QWidget, private Ui::PNoteEditor {
    Q_OBJECT
public:
    PNoteEditor( QWidget *parent=0);
    QString noteText();
    QString noteTitle();

signals:
    void noteTitleChanged(const QString&);
    void noteTextChanged(const QString&);

public slots:
    void setNoteText(const QString&);
    void setNoteTitle(const QString&);

private slots:
    void on_title_textChanged();
    void on_editor_textChanged();
};

#endif
```

Dans ce fichier nous commençons par nous prémunir contre l'inclusion multiple aux lignes 1 et 2, puis nous incluons la définition de la classe QWidget (ce qui est obligatoire car nous en héritons) ainsi que l'en-tête de Ui::PNoteEditor stockée dans le fichier ui_pnoteeditor.h. Notez au passage que le fichier ui_pnoteeditor.h n'existe pas encore, nous verrons plus loin comment faire pour qu'il soit généré automatiquement. Vient ensuite la déclaration de la classe PNoteEditor (l. 7) :

```
class PNoteEditor : public QWidget, private Ui::PNoteEditor {
    ...
};
```

PNoteEditor hérite simultanément de QWidget de façon publique (c'est-

à-dire que l'API de QWidget sera directement accessible depuis PNoteEditor ou d'une de ses instances) et de Ui::PNoteEditor de façon privée cette fois. Nous avons, ici, recours à l'héritage privé car la classe générée à partir de pnoteeditor.ui déclare toutes ses méthodes et attributs publics. Or ce n'est pas un comportement souhaitable et ce mécanisme n'est disponible que dans le but de simplifier la création de l'interface en permettant l'héritage de l'intégralité des variables, pointeurs et autres attributs de la classe Ui:: de base.

La première chose à faire lors de la création d'un objet Qt est l'insertion de la macro Q_OBJECT. Celle-ci permet l'intégration par le compilateur de méta-objet " moc " (pour Meta Object Compiler) de tout le code nécessaire aux objets Qt (tels que le mécanisme des signaux/slots, le système de propriétés et plus généralement tous les mécanismes d'inspection).

Ensuite, nous définissons les méthodes publiques, qui sont au nombre de 2 plus le constructeur. Celui-ci peut prendre un argument éventuel : son widget parent. Cela permet d'utiliser l'objet PNoteEditor dans un autre programme Qt. Les 2 méthodes sont noteTitle() qui retourne le contenu du QLineEdit (title) sous forme de QString, et noteText() qui retourne aussi une QString représentant le corps de la note renvoyé depuis le QTextEdit (editor).

Les méthodes permettant de positionner le contenu de ces 2 éléments sont déclarées comme des slots publics : setNoteText(const QString &) et setNoteTitle(const QString&) qui permettent de positionner respectivement le corps et le titre de la note.

Le mécanisme signal/slot de Qt est à la fois pratique et ingénieux, il permet à un objet Qt de sélectionner les événements importants, de les digérer et d'émettre un signal spécifiquement adapté au contexte. Prenons un exemple : l'événement transmis aux objets Qt lors de l'appui d'une touche de clavier (QKeyEvent) n'est pas toujours intéressant. Dans le cas d'un widget tel que QLineEdit, en revanche, cela peut être très important si la touche sur laquelle l'utilisateur vient d'appuyer modifie le texte contenu dans la zone de saisie. Le widget QLineEdit filtre donc les événements qui lui sont rapportés et quand il détecte l'appui sur une touche ayant un impact sur le texte de la zone de saisie, il émet le signal textChanged() qui informe les objets qui y sont connectés que le texte a changé dans le QLineEdit. Un signal ne fait rien en lui-même, il doit être connecté à un slot qui s'occupe de le gérer. Un slot est une fonction C++ classique et de ce fait peut être virtuel, public, privé, surchargé, etc. La seule différence qui existe entre un slot et une fonction C++ traditionnelle est qu'un slot peut être connecté à un signal. Dans ce cas de figure, le slot est appelé automatiquement à chaque fois que le signal est émis.

Un signal peut être connecté à un ou plusieurs slots, plusieurs signaux peuvent être connectés au même slot, enfin, un signal peut être connecté à un autre signal.

Pour connecter un signal à un slot il suffit de faire appel à la fonction statique QObject::connect (le prefix QObject n'est pas nécessaire si la macro Q_OBJECT a été placée dans la classe).

Exemple :

```
connect(widget, SIGNAL( unSignal() ), this, SLOT( unSlot() ) );
```

Pour qu'un signal soit correctement connecté à un slot, il faut que les arguments soient compatibles. Par exemple, le signal textChanged(const QString &) ne peut pas être correctement connecté au slot setPa-

ge(int) alors qu'il se connecte parfaitement au slot setNoteText(const QString &). Une exception à cette règle existe : la connexion à un slot avec moins de paramètres que le signal est possible mais pas encouragée. PNoteEditor déclare lui même 2 signaux : noteTitleChanged(const QString&) et noteTextChanged(const QString&). Ceux-ci sont émis à chaque fois que le texte de leur élément de saisie respectif change. L'émission des signaux à proprement parler est prise en charge par les 2 slots privés : on_title_textChanged() et on_editor_textChanged(). Ces derniers sont un peu spéciaux. En effet, nous pouvons constater que le nom de ces 2 méthodes respecte le même format on_<nom d'un élément graphique>_<quelque chose>(<paramètre(s)>). Ceci est une fonctionnalité particulièrement pratique introduite dans Qt 4 : la connexion par nom de slot !

Il n'est pas obligatoire, en effet, de connecter explicitement slots et signaux. Le moc ajoute, à la compilation, le code nécessaire à la connexion par nom de slots (voir QObject::connectSlotsByName dans l'assistant pour plus d'informations à ce propos).

Le mécanisme signal/slot est proposé par le système de méta-objets de Qt. Celui-ci permet d'écrire des classes communicantes sans qu'aucune de celles-ci ne sache quoi que ce soit sur les autres. Ce système propose aussi un mécanisme d'inspection que nous mettrons en pratique dans la suite de ce dossier.

Voyons maintenant le code de la classe (stocké dans le fichier source pnoteeditor.cpp). Celui-ci n'est pas bien complexe. Tout d'abord il faut inclure le fichier d'en-tête comme dans n'importe quel programme C++.

```
#include "pnoteeditor.h"
```

Il est ensuite nécessaire de mettre en place le constructeur de la classe PNoteEditor :

```
PNoteEditor::PNoteEditor( QWidget *parent ) : QWidget(parent){
    setupUi(this);
}
```

Ce dernier ne fait rien d'autre que d'appeler le constructeur de la classe parent (QWidget) et de mettre en place l'interface graphique que nous avons dessinée dans le designer (setupUi() provient de Ui::PNoteEditor). A la suite de cela viennent les 2 slots privés. Ceux-ci sont appelés à chaque fois que le texte de l'éditeur (le QTextEdit editor) ou de la barre de titre (le QLineEdit title) change. Nous en sommes avertis car chacun de ces widgets émet le signal textChanged(), et grâce à la connexion par nom de slots, on_editor_textChanged() et on_title_textChanged() sont appelés à chaque émission du signal par leur widget respectif.

```
void PNoteEditor::on_editor_textChanged(){
    emit( noteTextChanged( editor->toPlainText() ) );
}
void PNoteEditor::on_title_textChanged(){
    emit( noteTitleChanged( title->text() ) );
}
```

On remarquera l'utilisation du mot clé emit() pour l'émission d'un signal. C'est là tout le code que nous devons écrire pour profiter de ce puissant mécanisme. Le code des accesseurs est lui aussi très simple dans la mesure où il suffit de retourner ou de positionner le texte de la barre de

titre et de l'éditeur. Le widget QLineEdit possède une méthode de lecture de son contenu : text() et QTextEdit quant à lui implémente la méthode toPlainText(). Ces 2 méthodes retournent des QString, il suffit donc de retourner directement le résultat de leur appel respectif :

```
QString PNoteEditor::noteTitle(){
    return title->text();
}
QString PNoteEditor::noteText(){
    return editor->toPlainText();
}
```

Pour finir, il reste à écrire le code des 2 slots setNoteText(const QString&) et setNoteTitle(const QString&) :

```
void PNoteEditor::setNoteText(const QString& str){
    editor->setText(str);
}
void PNoteEditor::setNoteTitle(const QString& str){
    title->setText(str);
}
```

Là encore, la simplicité du code laisse rêveur !

Afin de tester ce code nous allons écrire un dernier fichier source : main.cpp. Celui-ci ne contient que le code de la fonction principale main() :

```
#include <QtGui>
#include "pnoteeditor.h"
int main(int argc, char *argv[]){
    QApplication app(argc, argv);
    PNoteEditor window;
    window.show();
    return app.exec();
}
```

Cette fonction crée une nouvelle application Qt puis instancie et affiche PNoteEditor. Le recours à QApplication nous permet, par exemple, d'ajouter le support des options de base des applications Qt comme "-style" ou "-title". Le code de cette première fenêtre étant désormais terminé il reste à le compiler. En avant !

Compilation et exécution d'un programme Qt :

Dans cette dernière partie, nous allons avoir besoin d'un terminal car les outils de compilation sont en lignes de commande.

La première chose à faire est de se rendre dans le répertoire QuickDesktopNotes dans lequel sont stockés tous les fichiers sources du programme :

```
cd <chemin d'accès>/QuickDesktopNotes/
```

Une fois dans ce répertoire l'affichage de son contenu devrait donner cela :

```
arno@Shai_Hulud:~/progz/Qt4/QuickDesktopNotes$ ls
pnoteeditor.h main.cpp pnoteeditor.cpp pnoteeditor.ui
```

Les utilisateurs des systèmes d'exploitation Windows doivent utiliser la

commande " dir " au lieu de " ls ". Afin de créer les fichiers de compilation pour le projet, nous allons utiliser l'outil qmake fourni avec toutes les distributions de Qt :

```
qmake -project
```

Ceci crée le fichier (texte brut) QuickDesktopNotes.pro, je vous conseille de vous familiariser avec sa syntaxe car nous allons très vite le modifier à la main ! Une fois le fichier projet (multi-plate-forme) généré, il faut créer les directives de compilation propres à la plate-forme hôte en appelant qmake sur le fichier projet :

```
qmake QuickDesktopNotes.pro
```

Enfin, suivant votre environnement de compilation il suffit d'invoquer les commandes de compilation, dont le plus classique :

```
make
```

Une fois la compilation terminée, il ne reste plus qu'à lancer le programme et à admirer le résultat !

Dans le prochain article nous mettrons en place le " system tray icon " et nous utiliserons PNoteEditor comme éditeur de note.

D'ici là bon développement.

Pré-requis :

Pour toutes les plates-formes, la bibliothèque Qt-4.3.x (à l'écriture de ces lignes la version courante est la 4.3.4) est un pré-requis impératif à cet article.

Linux :

Quasiment toutes les distributions Linux proposent un paquetage Qt 4 installable via l'outil de gestion de paquetage de la distribution.

Mac OS X :

Trolltech fournit un paquetage (.dmg) précompilé dans la section " download " de son site. Il est fortement conseillé d'installer aussi le paquetage contenant les bibliothèques de debug (QDebug).

Windows :

Comme pour Mac OS X un paquetage (.exe) est proposé par la société Trolltech. Pour cette plate-forme en revanche le paquetage est proposé sous 2 formes : incluant le compilateur MinGW ou non. Si vous n'avez pas d'environnement de développement pré-installé, nous conseillons fortement cette version basée sur GCC. Cela garantit un passage d'une plate-forme à l'autre moins douloureux...

Tous ces paquetages plus les sources sont disponibles ici : <http://trolltech.com/downloads/>

■ Arnaud Dupuis

Consultant Uperto/ Groupe Devoteam

Comment manipuler vos règles métier avec la DLR ?

Quand les sociétés de services développaient, il y a plus de 10 ans déjà, des Ateliers de Génie Logiciel, leur ambition était déjà de permettre aux équipes fonctionnelles d'implémenter elles-mêmes les règles métier.

Se passer de ces satanés développeurs ! Une véritable quête du Graal. Mais en réalité, que ce soit via l'utilisation des diagrammes ou encore en utilisant des moteurs de règles, on aboutissait souvent à un outil bien trop lourd et complexe à utiliser, même pour les développeurs ! Attendre que tout le monde sache parfaitement coder en Java ou C# est utopique, alors qu'il existe des langages bien plus accessibles : les langages de script.

- Une application va mettre à la disposition du trader des données (variables) à manipuler.
- Le trader écrit une règle en Python qui manipule ces données et produit un résultat.
- Quand elle en aura besoin, l'application exécutera ce script et récupérera le résultat.

Exemple n°1 : Une règle "standard"

Commençons par utiliser un objet pour gérer les données destinées à être manipulées par la règle métier.

```
class Equipment {
    int Output;
    int InputA;
    int InputB;
}
```

Dans ce cas, la règle écrite en Python sera :

```
" MonEquipment.Output = MonEquipment.InputA + MonEquipment
.InputB "
```

L'exécution de la règle consiste donc à instancier l'objet, initialiser les données d'entrées, exécuter à proprement dit la règle externe puis à exploiter le résultat de la règle. Voici le code C# équivalent :

```
// Instanciation de l'objet
Equipment myEquipment = new Equipment();
// Initialisation des données d'entrées
myEquipment.InputA = ...
myEquipment.InputB = ...
// Définition de la règle
String myRule = "MyEquipment.Output = MyEquipment.InputA + My
Equipment.InputB";
// Exécution de la règle
var myContext = new Dictionary<string, int>();
myContext.Add("MyEquipment", myEquipment);
PythonEngine pythonEngine = PythonEngine.CurrentEngine;
ScriptModule pythonModule = pythonEngine.CreateModule("myModule",
myContext,
ModuleOptions.None);
ICompiledCode myCompiledCode = pythonEngine.CompileCode(myRule,
pythonModule);
myCompiledCode.Execute(pythonModule);
```

```
// Exploitation du résultat
myEquipment = myContext["MyEquipment"] ;
... = myEquipment.Output ;
```

Les points importants

Dans cet exemple, l'insertion d'une règle métier a nécessité d'adresser différents points :

- La saisie de la règle (codée en dur pour l'exemple)
- L'alimentation de la DLR avec l'objet à manipuler
- L'exécution de la règle (pré-compilation, puis exécution)
- L'exploitation du résultat

En réalité, ces points sont systématiquement traités dans une application industrielle. Dans l'exemple précédent, l'écriture de la règle ne poserait aucune difficulté particulière à un développeur car il a une très bonne connaissance de la programmation objet et peut manipuler les objets dans les règles comme bon lui semble. Mais c'est beaucoup moins vrai avec une personne "non développeur". La syntaxe "MyEquipment.Something" n'est pas forcément limpide pour le commun des mortels.

Simplifions la syntaxe des règles métiers

Lorsque les règles doivent être écrites par une personne "non développeur", typiquement un expert fonctionnel, il devient alors nécessaire de porter une attention toute particulière à l'accessibilité. Habituellement, c'est le développeur ou l'architecte qui intègre les contraintes techniques et impose à l'utilisateur final le fonctionnement. Efforçons nous, pour une fois, de simplifier la vie de l'utilisateur quitte à faire un peu plus de travail d'un point de vue technique. Dans notre cas, le plus simple serait de faire disparaître notre objet "MyEquipment" : **Output = InputA + InputB**. Alors, au lieu de directement présenter l'objet à moteur de script, nous allons décomposer les données une par une :

Exemple n°2 : une écriture de règle simplifiée

```
...
// Définition de la règle
String myRule = "Output = InputA + InputB";
// Exécution de la règle
var myContext = new Dictionary<string, int>();
myContext.Add("InputA", MyEquipment.InputA);
myContext.Add("InputB", MyEquipment.InputB);
myContext.Add("Output", MyEquipment.Output);
...
// Exploitation du résultat
myEquipment.Sortie = myContext["Sortie"] ;
... = myEquipment.Sortie ;
```

Bilan

Nous avons ainsi fait un pas vers la simplification pour l'utilisateur, mais en contre partie nous avons un peu rigidifié l'application : les membres

DEVELOPPEZ VOTRE SAVOIR-FAIRE

ABONNEMENT
.NET



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

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

PROGRAMMEZ

- Abonnement 1 an au magazine : 45 €
(au lieu de 65,45 € tarif au numéro) Tarif France métropolitaine
- 1 an au magazine + 5 numéros HS .NET : 63 €
Tarif France métropolitaine
- Abonnement Intégral : 1 an au magazine + archives
sur Internet et PDF : 57 € Tarif France métropolitaine
- Abonnement PDF / 1 an : 30 € - Tarif unique
Inscription et paiement exclusivement en ligne
www.programmez.com
- Abonnement Etudiant : 1 an au magazine : 39 €
(au lieu de 65,45 € tarif au numéro) Offre France métropolitaine

PROGRAMMEZ HORS SERIE .NET Seulement

- Abonnement 1 an aux Hors Série : 5 numéros : 20 €
(au lieu de 25 € tarif au numéro) Tarif France métropolitaine



Abonnez-vous :

Les Hors-Série .Net
5 numéros par an : 20 €*

soit 1 Numéro
GRATUIT

*Tarifs France métropolitaine

Abonnez-vous :

Le magazine
mensuel
11 numéros par an : 45 €*
soit 3 Numéros
GRATUITS

*Tarifs France métropolitaine

+ Abonnement INTÉGRAL

NOUVEAU

ACCÈS ILLIMITÉ aux ARCHIVES du MAGAZINE pour 1€ par mois ! Prix de lancement

Cette option est réservée aux abonnés pour 1 an au magazine, quel que soit le type d'abonnement (Éco, Numérique, Etudiant). Le prix de leur abonnement normal est majoré de 12 € (prix de lancement, identique

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

OUI, je m'abonne

Vous pouvez aussi vous abonner en ligne et trouver tous les tarifs www.programmez.com

PROGRAMMEZ

- Abonnement 1 an au magazine : 45 € (au lieu de 65,45 € tarif au numéro) Tarif France métropolitaine
- 1 an au magazine + 5 numéros HS .NET : 63 € Seulement (au lieu de 90,45 € tarif au numéro) Tarif France métropolitaine
- Abonnement Intégral : 1 an au magazine + archives sur Internet et PDF : 57 € Tarif France métropolitaine
- Abonnement Etudiant : 1 an au magazine : 39 € (au lieu de 65,45 € tarif au numéro) Offre France métropolitaine

PROGRAMMEZ HORS SERIE .NET NOUVELLE OFFRE

- Abonnement 1 an aux Hors Série : 5 numéros : 20 € (au lieu de 25 € tarif au numéro) Tarif France métropolitaine

M. Mme Mlle Entreprise : Fonction :

Nom : Prénom :

Adresse :

Code postal : Ville :

Tél : E-mail :

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

A remplir et retourner sous enveloppe affranchie à :
Programmez ! - Service Abonnements - 22 rue René Boulanger - 75472 Paris Cedex 10.
abonnements.programmez@groupe-gli.com

Programmez!
LE MAGAZINE DU DEVELOPPEMENT

Offre limitée,
valable jusqu'au
30 juin 2008

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.

PROG 109

d'entrée et de sortie sont écrits en dur dans le code. Plusieurs solutions sont possibles dont l'introspection de la classe manipulée. Nous ne nous intéresserons pas à cette solution mais allons plutôt élargir la réflexion à la définition des objets manipulés.

Rendre le code plus générique

Dans certaines applications, typiquement lorsque l'enjeu principal consiste à faire paramétrer des règles de calcul sur de nombreuses entrées / sorties, il devient utile de définir une classe générique. Plutôt que d'avoir :

```
class Equipment1 {
    int Output;
    int InputA;
    int InputB;
}

class Equipment2 {
    int Output1;
    double Output2;
    int InputA;
    int InputB;
    double InputC;
}
```

On préférera avoir :

```
class GenericEquipment {
    Dictionary<string, object> inputs;
    Dictionary<string, object> outputs;
}

// Instanciation et initialisation des données d'entrées
myEquipment1 = new GenericEquipment();
myEquipment1.outputs.Add("Output");
myEquipment1.inputs.Add("InputA");
myEquipment1.inputs.Add("InputB");

// Instanciation et initialisation des données d'entrées
myEquipment2 = new GenericEquipment();
myEquipment2.outputs.Add("Output1");
myEquipment2.inputs.Add("Output2");
myEquipment2.inputs.Add("InputA");
myEquipment2.inputs.Add("InputB");
myEquipment2.inputs.Add("InputC");
```

Cette méthode offre de nombreux avantages :

- Une définition simple des entrées et sorties
 - basée sur un type générique " object ", compatible avec les opérateurs IronPython (par exemple : " + " sur int, string, double, etc.)
 - pouvant être automatisée à partir de fichier de configuration définissant la structure des objets eux-mêmes.
- Une alimentation souple du contexte de la DLR en entrée et en sortie (par rapport à l'exemple 2)
- Un transport natif de ces objets sur le réseau, de façon simple et efficace, directement via .Net Remoting ou WCF (la définition étant connue et stable)

Exemple n°3 : Core Domain composé d'objets issus d'une classe générique

```
// Instanciation et initialisation des données d'entrées
GenericEquipment myEquipment = new GenericEquipment();
myEquipment.outputs.Add("Output");
myEquipment.inputs.Add("InputA", ...);
myEquipment.inputs.Add("InputB", ...);

// Définition de la règle
String myRule = "Output = InputA + InputB";

// Exécution de la règle via la DLR
var myContext = myEquipment.inputs.Union(myEquipment.outputs).ToDictionary(
    s => s.Key, s => s.Value);

PythonEngine pythonEngine = PythonEngine.CurrentEngine;
ScriptModule pythonModule = pythonEngine.CreateModule("myModule",
    myContext,
    ModuleOptions.None);
ICompiledCode myCompiledCode = pythonEngine.CompileCode(myRule,
    pythonModule);
myCompiledCode.Execute(pythonModule);

// Exploitation du résultat
foreach (var element in myContext)
{
    if (myEquipment.outputs.ContainsKey(element.Key))
        myEquipment.outputs[element.Key] = element.Value;
}
... = myEquipment.Sortie ;
```

Le référencement de plusieurs objets dans une même règle

Le revers de la médaille de cette dernière implémentation, bien que très pratique pour l'écriture des règles, est qu'elle ne permet pas de référencer des membres d'autres objets. Notre expérience projet montre que les besoins utilisateurs peuvent parfois nécessiter le référencement d'objets multiples au sein d'une même règle et donc d'introduire une écriture plus " objet " des règles. Par exemple : **Output = InputA + InputB + MyReference.InputA + MyReference.InputB**. On retrouve alors les défauts d'écriture auxquels nous avons réussi à échapper. Partiellement et localement tout du moins : sur une partie des opérandes de la règle (pas tous les opérandes), sur une règle (pas toutes les règles). Le contexte DLR doit donc s'enrichir de nouveaux objets : ceux référencés par la règle. La problématique est que la définition au sens objet de " MyReference " n'existe plus, celle-ci ayant été remplacée par une instance de la classe générique `GenericEquipment`. La solution technique proposée est donc d'injecter dynamiquement les types nécessaires à la bonne exécution de la règle. Ainsi, nous conservons le principe des classes génériques et profitons des avantages intrinsèques liés à la définition des objets manipulés, tout en offrant le référencement d'autres objets au sein des règles.

Exemple 4 : une règle référençant plusieurs objets

```
// Instanciation et initialisation des données d'entrées
myEquipment1 = new GenericEquipment();
```

```

myEquipment1.outputs.Add("Output");
myEquipment1.inputs.Add("InputA");
myEquipment1.inputs.Add("InputB");

// Instanciation et initialisation des données d'entrées
myEquipment2 = new GenericEquipment();
myEquipment2.outputs.Add("Output1");
myEquipment2.outputs.Add("Output2");
myEquipment2.inputs.Add("InputA");
myEquipment2.inputs.Add("InputB");
myEquipment2.inputs.Add("InputC");

// Définition de la règle
String myRule = "Output = InputA+InputB + MyReference.InputA + My
Reference.InputB";

// Exécution de la règle via la DLR
var myContext = myEquipment.inputs.Union(myEquipment.outputs).To
Dictionary(
    s => s.Key, s => s.Value);

Type myType = createTypeFromEquipement(myEquipment2);
object myReference = createInstanceOfType (myType);
setOutputsOfInstance (myReference, myType);

myContext.Add("MyReference", myReference);

PythonEngine pythonEngine = PythonEngine.CurrentEngine;
ScriptModule pythonModule = pythonEngine.CreateModule("myModule",
myContext,
    ModuleOptions.None);
ICompiledCode myCompiledCode = pythonEngine.CompileCode(myRule,
pythonModule);
myCompiledCode.Execute(pythonModule);

// Exploitation du résultat
foreach (var element in myContext)
{
    if (myEquipment.outputs.ContainsKey(element.Key))
        myEquipment.outputs[element.Key] = element.Value;
}
... = myEquipment.Sortie;

private Type createTypeFromEquipement (GenericEquipment generic
Equipment)
{
    AssemblyName aName = new AssemblyName("DynamicTypes
Container");
    AssemblyBuilder ab = AppDomain.CurrentDomain.DefineDynamic
Assembly(
        aName, AssemblyBuilderAccess.RunAndSave);

    // For a single-module assembly, the module name is usually
    // the assembly name plus an extension.
    ModuleBuilder mb = ab.DefineDynamicModule(aName.Name, aName.

```

```

Name + ".dll");

    TypeBuilder tb = mb.DefineType(typeName, TypeAttributes.Public);

    // Create one field per output
    foreach (var field in genericEquipment.outputs)
    {
        FieldBuilder fb = tb.DefineField(field.Key, typeof(object), FieldAttributes.
Public);
    }

    // Finish the type.
    Type t = tb.CreateType();
    ab.Save(aName.Name + ".dll");

    return t;
}

private object createInstanceOfType(Type t)
{
    object myObject = Activator.CreateInstance(t);
    return myObject;
}

private object setOutputsOfInstance(object reference, Type referenceType,
GenericEquipment genericEquipment)
{
    foreach (var field in genericEquipment.outputs)
    {
        FieldInfo fi = referenceType.GetField(fieldName);
        fi.SetValue(reference, ...);
    }
}

```

Bilan

3 nouvelles méthodes pour la création dynamique du type, largement inspirées de la documentation MSDN de TypeBuilder et FieldBuilder. Des spécificités sont ajoutées par la gestion des membres à partir de la définition de l'objet générique (ajout et affectation de valeur).

Conclusion

Les 4 exemples démontrent que la gestion de règles applicatives et modifiables une fois l'application compilée, nécessite des points particuliers d'attention notamment sur :

- La syntaxe de la règle
- L'alimentation de la DLR avec le ou les objets à manipuler
- L'exécution de la règle
- L'exploitation du résultat

N'oubliez jamais que quelle que soit la complexité des mécanismes techniques que vous mettez en place, la manipulation de ces règles métier par les utilisateurs finaux doit être la plus simple possible.

■ Olivier Dessyn - Architecte Senior - OCTO Technology

■ Eric Groise
 Responsable CCI .NET
 OCTO Technology



Tutoriel sur
www.programmez.com/tutoriel.php

Optimisez votre code Python avec Pyrex et Psyco

Python est un langage de script de qualité, performant dans l'univers des langages de script, mais incapable de rivaliser avec du code natif. Toutefois, avec Pyrex ou Psyco, votre code Python peut devenir un foudre de guerre.

Quand Guido van Rossum a créé Python, il a pensé performance. Il n'a pas tellement pensé performance à l'exécution, même si Python est brillant parmi les langages de script. Il a essentiellement pensé performance à l'écriture. Tous les pythoneurs le savent bien, griffonner un malin petit script en Python, pour se fabriquer un petit utilitaire d'administration ou essayer un algorithme, consomme au minimum une ressource précieuse entre toutes pour le développeur: le temps. Ecrire en Python est tellement rapide et agréable que souvent on se laisse aller pour produire un code qui grossit, grossit et devient alors une véritable application, avec naturellement quelques goulets d'étranglement. Qu'il est dommage que les scripts Python ne puissent être compilés en code natif. Et bien si! c'est possible, avec Pyrex, sorte de langage hybride entre C et Python, et avec Psyco, sorte de compilateur JIT pour Python. Avec cet article nous allons découvrir comment utiliser et exploiter pertinemment ces outils.

1 Les outils nécessaires

Vous devez tout d'abord avoir, qui l'eût cru, Python installé sur votre machine. Au-delà de la boutade, cela est nécessaire car Pyrex et Psyco sont eux-mêmes des modules d'extensions à Python. Un compilateur C est nécessaire à l'installation de Pyrex et Psyco, ainsi que pour l'utilisation de Pyrex. Si vous travaillez avec un système Unix ou Linux, tout va bien car le compilateur C de votre système sera utilisé. Si vous travaillez sous Windows c'est un peu plus compliqué. Ainsi le Python 2.5 que j'utilise a été compilé avec Visual Studio 2003 qui, en ce qui me concerne, hante le fond d'un placard poussiéreux depuis longtemps. C'est ce compilateur qui sera recherché lors de l'installation de nos deux lascars, et s'il n'est pas trouvé, il vous sera proposé d'utiliser Mingw32, le compilateur gcc pour Windows, en remplacement. Disposer de ce compilateur sur Windows est de toute façon une bonne idée, donc installez-le depuis <http://www.mingw.org/>, le plus pratique étant de procéder depuis l'utilitaire d'installation en ligne.

2 Installation de Pyrex et Psyco

Une fois Mingw32 installé, vous devez faire pointer votre PATH sur le sous-répertoire bin de son arborescence. Ensuite, le plus simple ici est de demander à Python de l'utiliser comme compilateur C par défaut. Pour cela, on éditera (ou on créera), dans l'arborescence de Python, le fichier Lib\distutils\distutils.cfg et dans ce fichier, on ajoutera la section :

```
[build_ext]
compiler=mingw32
include_dirs=G:\MinGW\lib\gcc\mingw32\3.4.5\include
```

Pyrex et Psyco seront téléchargés respectivement à <http://www.cosc.canterbury.ac.nz/greg.ewing/python/Pyrex/> et <http://psyco.sourceforge.net/>



ge.net/. Ils viennent tous deux sous la forme d'une archive qu'il suffit de décompresser dans un répertoire de travail quelconque. Chaque archive contient un fichier setup.py, c'est à dire un script Python standard pour l'installation. Il suffit donc de donner pour chacun la commande :

```
python setup.py install
```

Pyrex et Psyco seront alors installés sous la forme de modules d'extensions à Python.

3 Un Hello World! en Pyrex

Tradition oblige... mais en l'occurrence c'est un excellent moyen de se familiariser avec Pyrex et ses possibilités. Le but fondamental de Pyrex est de pouvoir écrire des modules d'extensions à Python en C, mais sans devoir, aux dires de son créateur, plonger dans le cambouis de la programmation bas niveau en C. Pour cela, Pyrex est un langage hybride qui emploie une syntaxe à la Python pour la plus grande partie du code, mais qui demande à déclarer les variables sous une forme "à la C". Le code Pyrex est alors compilé par le compilateur Pyrex pour générer un fichier source C qui sera à son tour compilé par gcc ou Mingw32. A la fin de l'opération nous disposons d'un module d'extension à Python, performant car en code natif. Ce caractère hybride de Pyrex permet aussi d'appeler facilement du code d'une librairie C existante. C'est d'abord cette possibilité que nous allons utiliser pour notre Hello World. Nous voulons appeler l'API Windows MessageBox. Nous nous intéressons aujourd'hui seulement au cas de Windows. Le lecteur qui souhaiterait faire un équivalent sous Linux suivra les mêmes principes, sachant que les choses sont plus faciles sous Linux. Voici d'abord le prototype C de l'API Windows que nous désirons appeler :

```
int MessageBox(HWND hWnd, LPCSTR lpText, LPCSTR lpCaption, UINT uType);
```

En C on l'invoquerait comme ceci :

```
MessageBox(NULL, "Programmez! avec Pyrex", "Demo Pyrex", MB_OK);
```

Comment cela va-t-il aller avec Pyrex ? MB_OK est un entier, on pourrait facilement trouver sa valeur dans les en-têtes Windows. Qu'est-ce donc que HWND ? Si l'on explore les en-têtes, on voit que HWND revient à void * donc dans notre cas nous devons simplement passer un pointeur nul. Ceci posé, nous ne devons pas explorer les en-têtes Windows à chaque fois, heureusement. Pyrex le fera pour nous, il suffit de lui

demandeur comme ceci, dans du code pyrex (fichier fred.pyx sur le CD-Rom, ou le site).

```
cdef extern from "windows.h":
    int MB_OK
    extern int __stdcall MessageBox(void *, char *, char *, unsigned int)
```

cdef signale d'aller chercher des types et prototypes C. Nous allons les chercher dans windows.h, bien que MessageBox soit en fait déclarée dans l'en-tête winuser.h. Donner ici winuser.h serait une faute qui aboutirait à des messages d'erreurs effrayants lors de la phase de compilation C, en raison de définitions manquantes. Nous donnons donc windows.h et nous laissons Pyrex se débrouiller. Ensuite nous demandons à Pyrex de se renseigner sur MB_OK et sur l'API MessageBox dont nous avons quand même clarifié le prototype à l'intention de Pyrex. En outre on notera bien la présence, obligatoire pour les API Windows, de __stdcall pour "standard call", et qui définit la convention d'appel des fonctions, à savoir ici l'empilage des paramètres de la droite vers la gauche comme en C et le réajustement de la pile par l'appelé, comme en Pascal. Un appel qui n'a donc de standard que le nom, mais nous sommes sous Windows, ceci explique cela :) Tout est prêt, nous pouvons fièrement écrire, toujours dans le fichier fred.pyx, le code qui invoquera l'API :

```
def helloworld():
    MessageBox(NULL, "Programmez! avec Pyrex", "Demo Pyrex", MB_OK)
```

Nous voyons que nous avons là ni plus ni moins que du code Python classique, avec MessageBox qui est vue comme une fonction intégrée, ce qui est bien pratique. Reste à construire le module. Nous pourrions invoquer manuellement le compilateur Pyrex, puis gcc. Cependant, le plus simple et le plus dans l'esprit de Python, est d'écrire un classique fichier setup.py pour la construction et l'installation du module. Voici ce fichier :

```
from distutils.core import setup
from distutils.extension import Extension
from Pyrex.Distutils import build_ext
setup(
    name = "fred",
    ext_modules=[
        Extension("fred", ["fred.pyx"])
    ],
    cmdclass = {'build_ext': build_ext}
)
```

Nous donnons alors la commande

```
python setup.py build_ext install
```

pour voir notre module construit et installé. On essaiera notre module comme tout autre module Python, comme ceci :

```
python
>>> import fred
>>> fred.helloworld()
```

4 Un exemple plus complexe

Nous souhaitons cette fois invoquer l'API Windows GetVolumeInformation pour obtenir des informations sur le disque dur de notre Win-

dows, à savoir le nom de volume, le numéro de série et le système de fichiers. L'API est prototypée ainsi :

```
BOOL GetVolumeInformation(
    LPCSTR lpRootPathName,
    LPCSTR lpVolumeNameBuffer,
    DWORD nVolumeNameSize,
    LPDWORD lpVolumeSerialNumber,
    LPDWORD lpMaximumComponentLength,
    LPDWORD lpFileSystemFlags,
    LPSTR lpFileSystemNameBuffer,
    DWORD nFileSystemNameSize);
```

Nous nous heurtons à deux difficultés. D'abord comment faire pour passer un tampon à l'API, ensuite comment faire pour lui passer des pointeurs ? Pour passer un tampon, il faut l'allouer. Comment faire en Pyrex ? Quand nous utilisons un module Pyrex, nous le faisons forcément depuis l'interpréteur Python. C'est donc finalement lui qui devra allouer/désallouer les tampons. Nous devons alors, depuis Pyrex, invoquer des fonctions de l'interpréteur Python. Par exemple PyMem_Malloc et PyMem_Free. Cependant, profitant du fait qu'une chaîne Python peut contenir des zéros, il est encore plus simple d'utiliser PyString_FromStringAndSize. En effet celle-ci va créer un objet Python, et la libération de ce dernier sera gérée automatiquement par l'interpréteur. En ce qui concerne l'accès aux fonctionnalités externes, notre script Pyrex devient donc :

```
cdef extern from "windows.h":
    int MB_OK
    extern int __stdcall MessageBox(void *, \
        char *, char *, unsigned int)
    int MAX_PATH
    extern int __stdcall GetVolumeInformation(char *, \
        char *, long, long *, long *, long *, char *, long)

cdef extern from "Python.h":
    object PyString_FromStringAndSize(char *v, Py_ssize_t len)
    object PyString_FromString(char *v)
```

Et voici maintenant la fonction qui va obtenir les informations relatives à notre disque dur puis les afficher :

```
def infos():
    cdef char *root
    cdef long serial
    cdef long max_com_len
    cdef long flags
    cdef unsigned long true_serial

    size = MAX_PATH+1
    root = NULL
    name = PyString_FromStringAndSize(NULL, size)
    filesystem = PyString_FromStringAndSize(NULL, size)
    GetVolumeInformation(root, name, size, &serial, \
        &max_com_len, &flags, filesystem, size)
    print "Infos sur disque racine"
    true_name = PyString_FromString(name)
```

```
print "Nom du volume: %" %true_name
true_serial = serial
print "Numero de serie: %X" %true_serial
true_filesystem = PyString_FromString(filesystem)
print "Systeme de fichier: %" %true_filesystem
```

On remarque que, lors de l'appel à GetVolumeInformation, on passe à cette dernière aussi bien un type C, par exemple serial, qu'un pointeur sur un type C ou un objet Python, par exemple size, ou un objet Python qui est vu comme un pointeur par l'API, par exemple name. Pyrex à l'amabilité de s'occuper de toute la plomberie pour nous. On notera encore l'astuce de l'emploi de PyString_FromString. Normalement celle-ci construit une chaîne Python depuis un pointeur char *. Ici nous lui passons en fait un objet Python, mais là aussi Pyrex se charge de la conversion. Nous employons cette astuce pour obtenir des chaînes humainement lisibles, ce que ne sont pas name et filesystem à l'issue de l'invocation de l'API. En effet, le contenu du tampon est aléatoire et une chaîne Python pouvant contenir des zéros, l'interpréteur afficherait la totalité de ce tampon au lieu des chaînes C à zéro terminal censées être obtenues via l'API et qui seules nous intéressent.

5 Pyrex, Psyco et les performances

Maintenant que nous savons manier Pyrex, et sachant que mettre en œuvre Psyco est une simple formalité voyons ce que ces deux là nous apportent en termes de performances. Commençons par écrire (fichier fredpy.py) une fonction Python qui dévore du temps CPU :

```
def calcul():
    for i in range(100000):
        v = 0
        for j in range(1000):
            v = v + j
```

Dans un fichier nommé fredpsyco.py nous écrivons la même fonction exactement, mais nous la baptisons calcul_lent. Puis nous ajoutons le code qui fera que Psyco compilera la fonction en code natif lorsque le module fredpsyco.py sera chargé :

```
try:
    import psyco
    psyco.bind(calcul_lent)
    calcul = psyco.proxy(calcul_lent)
except ImportError:
    print "Installez psyco d'abord"
```

Enfin nous dotons le fichier fred.pyx de la même fonction, mais bien entendu écrite à la Pyrex :

```
def calcul():
    cdef int i, j, n, m, v
    n = 100000
```

```
m = 1000
for i from 0 <= i < n:
    v = 0
    for j from 0 <= j < m:
        v = v + j
```

Nous lançons ensuite notre programme de test (fichier test.py) qui commence ainsi :

```
import time
import fred
import fredpy
import fredpsyco

print "Temps Python:"
start = time.time()
fredpy.calcul()
temps_py = time.time()-start
print temps_py
# et ainsi de suite ...
```

Voici les résultats obtenus:

```
PythonPyrex Psyco
15.98 s 0.094 s0.25 s
```

Nous voyons qu'avec Pyrex l'accélération est fantastique, ce qui ne surprend pas, puisqu'au final Pyrex produit du code C et donc natif. Également remarquable est le résultat affiché par Psyco. A tel point qu'on en vient à se demander si ça vaut le coup de travailler avec Pyrex ? Certes celui-ci est facile et agréable à utiliser, mais avec Psyco, il n'y a rien à faire. La réponse est pourtant oui. En effet nous avons utilisé Psyco dans le cas où il est le plus efficace: l'optimisation d'une fonction Python totalement inefficace et qui comporte une boucle. En dehors de ce cas particulier Psyco restera intéressant ou le sera moins selon la configuration du code Python. Par ailleurs, Psyco doit être installé sur la machine hôte, tandis qu'un module Pyrex est autonome. Enfin, avec Psyco la possibilité d'invoquer une API du système hôte n'existe pas. Psyco n'est qu'un outil, certes remarquable. De son côté, Pyrex présente l'intérêt de l'autonomie et d'une plus grande souplesse. C'est un vrai langage qui permet non seulement de booster du code Python existant moyennant des modifications mineures mais aussi d'écrire du code Pyrex pur, avec une efficacité très proche de C, et sans les tracasseries inhérentes à un langage de bas niveau. A vous de choisir celui qui vous conviendra le mieux :)

■ Frédéric Mazué
fmazue@programmez.com

+ **Tutoriel sur**
www.programmez.com/tutoriel.php

Boutique Boutique Boutique

Achetez les magazines, les articles en PDF
et abonnez-vous en ligne

www.programmez.com



Constituer et expédier un e-mail depuis Windows Vista

Constituer un e-mail est une tâche aisée grâce aux API Windows Vista. En revanche, l'expédier est moins immédiat. Nous étudions tout cela avec C++.

Les mois précédents nous nous sommes intéressés à ce qui pouvait être les éléments constitutifs d'une application prenant des photos à intervalles réguliers avec une webcam. Il est maintenant temps de voir comment il est possible d'expédier ces photos en fichiers attachés dans des e-mails.

1 Etat des lieux

Ce ne sont pas les possibilités qui manquent pour atteindre notre but d'aujourd'hui, et paradoxalement, c'est un peu le problème. Windows est un système d'exploitation présentant un jeu d'API très riches. Pourtant, c'est seulement quand la suite Office, et donc Outlook Office, est installée que le travail est le plus facile. On utilisera alors les interfaces Automation pour créer et expédier un e-mail depuis Outlook Office. Facile ? Pas tellement finalement. Ça le serait avec un langage à l'aise avec Automation, par exemple Visual Basic, un langage .Net ou encore Python, exemples non limitatifs. Or nous voulons travailler en C++ et on ne peut pas dire que l'horrible interface Automation IDispatch facilite l'écriture d'un code élégant et concis avec ce langage. Regardons maintenant du côté de Windows Mail. Celui-ci remplace les Outlook Express des Windows précédents. Il est donc a priori présent sur la machine, sans que cette présence puisse être toutefois certaine. Si l'on consulte la MSDN à son sujet, on voit que les interfaces COM qui existaient sous Outlook Express sont toujours répertoriées, mais qu'un grand nombre sont déclarées dépréciées, ne sont pas garanties être présentes ou, si elles le sont, peuvent être implémentées partiellement seulement. L'affaire se présente mal. Les interfaces qui permettent de générer un e-mail, ou de créer un répertoire dans Windows mail, ne sont pas dépréciées, sont bel et bien présentes et opérationnelles. C'est le point positif. Par contre dès que l'on regarde des interfaces telles ISMTPTransport pour l'expédition d'un mail, ça ne va plus. Cette interface est dépréciée. En examinant la base de registre de mon Vista, j'ai pu vérifier que cette interface était bien enregistrée, mais qui sait si ce sera le cas après l'installation d'un service pack à venir ? La prudence s'impose. En outre, des interfaces comme ISMTPCallback n'ont pas d'implémentation par défaut. Nous devrions donc en fournir une. Programmer COM côté client, c'est déjà quelque chose, côté serveur (c'est-à-dire en définissant une interface COM) c'en est une autre, et dans ce contexte, la perspective est peu engageante. Pourquoi alors ne pas utiliser une API de plus bas niveau comme la MAPI de Windows ? Cela ne va pas très bien non plus. D'abord la MSDN nous informe que la MAPI ne sera fonctionnelle que si Windows Mail (alias Outlook Express) est présent sur le système. Nous ne sommes donc toujours pas affranchis de ce dernier. Mais surtout, un message envoyé via la MAPI peut provoquer l'apparition d'une boîte de dialogue informant que "quelqu'un essaie d'envoyer un courrier depuis votre compte". On peut même imaginer qu'un anti-virus bloque carrément l'e-mail. Même sans cela, nous voulons que notre application

soit autonome, et la perspective de cette boîte de dialogue est inacceptable. Comment faire alors ? En adoptant un moyen terme et en procédant en deux étapes. Dans la première, on génère un mail avec les API COM de haut niveau de Windows Mails. Cela nous évitera la fastidieuse tâche de devoir nous occuper de l'encodage, des parts relatives aux fichiers attachés, ou autres. Nous avons en outre par ce moyen la possibilité de placer les mails dans un répertoire donné de Windows Mails, ou même, pourquoi pas, dans la boîte d'envoi pour une expédition ultérieure. Un fois le mail constitué, nous l'expédierons simplement à travers un socket avec le bon vieux protocole SMTP. Nous plaçant ainsi en-dessous des MAPI et autres, nous nous affranchissons des complications. Plus de boîtes de dialogue intempestives, et, en toute objectivité, le code est infiniment plus simple en procédant ainsi :

2 Les outils

Tout le code d'exemple, que vous trouverez sur le CD-Rom accompagnant le magazine ou sur notre site, a été écrit avec Visual Studio 2005 et essayé avec Visual Studio 2008. Sa compilation avec un autre compilateur ne posera pas de problème. Dans tous les cas, la Windows SDK Vista, c'est-à-dire la dernière SDK en date doit être utilisée. Par contre, on se méfiera comme du feu des exemples de code pas toujours de bonne qualité venant avec cette SDK. Un bon nombre d'exemples ne fonctionnent carrément pas, en admettant qu'ils compilent. On retombe sur le problème d'une documentation de mauvaise qualité que nous avons signalé dans notre article "Maîtriser la nouvelle API Windows Contacts de Vista" de Programmez 106, article qui était déjà relatif à Windows Mail. La programmation COM, abondamment traitée récemment, notamment dans Programmez! 106 est assez rébarbative au niveau de l'acquisition des interfaces, de leur libération et de la gestion des erreurs. C'est pourquoi, comme précédemment, nous fournissons quelques classes utilitaires pour profiter de l'idiome "acquisition de ressources par initialisation" de C++. Ainsi nous pouvons écrire un code beaucoup plus concis, car d'une logique allégée. En ce qui concerne les sockets, cet article ne veut pas traiter de ce sujet. Nous fournissons donc quelques classes représentant une encapsulation légère des sockets Windows, le mot "légère" n'étant pas du tout péjoratif, bien au contraire. Ces classes ont été prises sur le site <http://www.adp-gmbh.ch/win/misc/sockets.html>, votre serveur y ayant seulement apporté de petites modifications sur lesquelles nous reviendrons plus loin.

3 Compiler des fichiers IDL et utiliser les en-têtes en résultant.

Conformément aux indications de la documentation MSDN, nos exemples incluent les en-têtes *imnexport.h*, *mimeole.h*, *msoeapi.h*. Cependant, si l'on examine le répertoire include de la SDK, on constate

que ces fichiers n'existent tout simplement pas. En revanche, on y trouve des fichiers de mêmes noms, mais suffixés `.idl`. De quoi s'agit-il ? Ces fichiers `.idl` contiennent les définitions des interfaces COM de Windows Mail, exprimées dans le langage IDL, pour Interface Definition Language. Il s'agit de fichiers textes contenant moult informations sur lesquelles il n'est pas nécessaire de s'attarder aujourd'hui. Ce que nous devons savoir est que ces fichiers `.idl` doivent être compilés par un compilateur approprié. A l'issue de cette compilation, de nombreux fichiers seront produits, parmi lesquels seuls les en-têtes (fichiers `.h`) nous intéressent. L'opération de compilation des `.idl` demande d'être minutieuse. D'abord, nous devons savoir que nous devons en fait compiler quatre fichiers, car l'un des en-têtes mentionnés plus haut en inclut un autre. Voici donc les quatre fichiers à compiler :

```
imnexport.idl
mimeole.idl
msoeapi.idl
imnact.idl
```

Ensuite nous ne devons pas utiliser n'importe quel compilateur IDL. L'exécutable de celui-ci s'appelle `midl.exe`. Dans le Visual Studio 2005 professionnel, utilisé pour cet article, il s'en trouve un. Il s'en trouve également un dans toutes les SDK Windows. C'est toujours le compilateur IDL de la SDK qui contient les fichiers `.idl` qui doit être utilisé, faute de quoi, on court le risque d'être ensevelis sous une flopée de messages d'erreurs abscons, à faire dresser les cheveux sur la tête. Ainsi, bien que j'ai travaillé avec Visual Studio 2005, c'est bien le MIDL de la SDK Vista que j'ai utilisé. Voici à titre d'exemple une commande complète pour compiler un fichier. La commande est à saisir sur une seule ligne :

```
midl /app_config /I "C:\Program Files\Microsoft SDKs\Windows\v6.0\Include" /I "C:\Program Files\Microsoft SDKs\Windows\v6.0\VC\INCLUDE" msoeapi.idl
```

Les compilations effectuées, il est possible d'inclure les en-têtes dans notre code, ce qui provoquera ici aussi de redoutables messages d'erreurs. Le remède consiste à écrire dans le code :

```
#define INITGUID
```

à l'endroit psychologique. Nous renvoyons le lecteur à nos exemples sur le CD-Rom. Nous ne nous attardons pas ici sur ce mécanisme. Le lec-

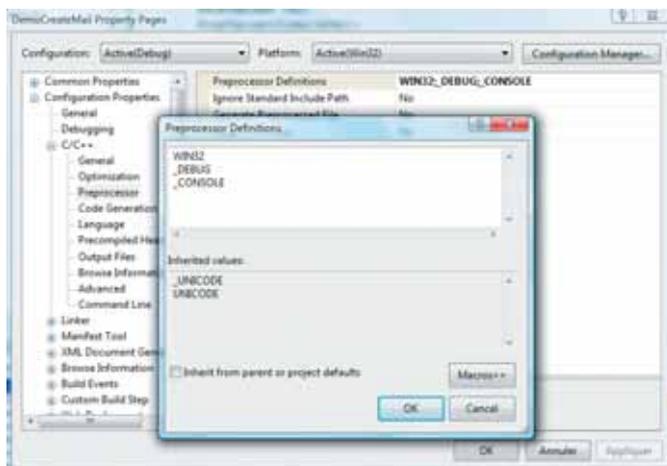


Fig. 1 : La macro `UNICODE` ne doit pas être définie, pour une bonne compilation de nos exemples.

teur intéressé pourra étudier l'en-tête Windows `guiddef.h`. Une dernière recommandation avant d'entrer enfin dans le vif du sujet. Windows Mail propose (parfois) deux jeux d'interfaces COM. Par exemple `IMimeMessage` et `IMimeMessageW`. La deuxième permet de travailler en Unicode. Afin d'alléger notre code, nous avons évité les interfaces Unicode. Cependant Visual Studio définit par défaut la macro `UNICODE`, et le compilateur cherchera les interfaces correspondantes. Nous devons donc procéder à un petit réglage qui consiste à décocher la case "Inherits from parent or project defaults" dans les définitions à l'intention du préprocesseur, comme illustré figure 1.

4 Windows Mail, Espace de noms et répertoires

A l'affichage arborescent des dossiers dans l'interface de Windows Mail correspond une structure arborescente de répertoires sur disque. Cette structure est dénommée "Espace de noms de stockage". Sur le disque les répertoires se situent à :

```
C:\Users\ votre nom utilisateur\AppData\Local\Microsoft\Windows Mail\Local Folders
```

Pour accéder à ces répertoires dans l'Explorateur Windows, vous devrez d'abord demander l'affichage des dossiers cachés dans les options des dossiers. Pour gérer le tout, deux interfaces COM sont proposées: `IStoreNamespace` et `IStoreFolder`. La première permet de parcourir l'arborescence, d'accéder aux propriétés des répertoires, de créer, déplacer, effacer un répertoire, etc. La seconde permet de travailler avec le contenu d'un répertoire. Notamment en parcourant les messages qui y sont stockés. La manipulation des répertoires est assez triviale et ne mérite pas qu'on s'y attarde longuement. Voici un exemple qui ouvre l'espace de nom, affiche les propriétés du répertoire racine et crée un répertoire du nom de "Mailing Programmez!" :

```
#include <iostream>

#define INITGUID
#include <COMUtils.h>
#include <msoeapi.h>

void DisplayFolder(FOLDERPROPS& p)
{
    std::cout << "Folder: " << p.szName << std::endl;
    std::cout << "ID: " << p.dwFolderId << std::endl;
    std::cout << "Nb de sous repertoire: " << p.cSubFolders << std::endl;
    std::cout << "Type: " << p.sfType << std::endl;
    std::cout << "Messages: " << p.cMessage << std::endl;
    std::cout << "Non lus: " << p.cUnread << std::endl;
}

int main(int argc, char* argv[])
{
    try
    {
        COMInitializer ci;
        HRESULT hr;
        SmartCOMPtr<IStoreNamespace> sn;
```

```

hr = ::CoCreateInstance(CLSID_StoreNamespace,
    NULL,
    CLSCTX_INPROC_SERVER,
    IID_IStoreNamespace,
    (void **)&sn);
COMHelper::TestOk(hr);

hr = sn->Initialize(NULL, 0);
COMHelper::TestOk(hr);

FOLDERPROPS folder_props = {0};
folder_props.cbSize = sizeof(FOLDERPROPS);
HENUMSTORE enum_store;

hr = sn->GetFirstSubFolder(
    FOLDERID_ROOT, &folder_props, &enum_store);
COMHelper::TestOk(hr);
DisplayFolder(folder_props);

hr = sn->GetSubFolderClose(enum_store);
COMHelper::TestOk(hr);

STOREFOLDERID mp_id;
hr = sn->CreateFolder(
    FOLDERID_ROOT, "Mailing Programmez!", 0, &mp_id);
COMHelper::TestOk(hr);
std::cout
    << "repertoire \"Mailing Programmez! cree\""
    << std::endl;
}
catch(COMException ce)
{
    std::cout << ce.raison() << std::endl;
    return EXIT_FAILURE;
}
return EXIT_SUCCESS;
}
    
```

Il est bon d'être averti du fait que si le répertoire existe déjà lors de la tentative de création, celle-ci a lieu quand même, avec un nom de répertoire suffixé. Il est également bon de savoir que Windows Mail ne sait pas placer un répertoire dans la corbeille si un répertoire de même nom y existe déjà. Veillez à vider la corbeille si vous avez des difficultés à effacer un répertoire créé, et si tout va mal, agissez directement sur l'arborescence disque. Il est également à remarquer le côté peu pratique de l'API qui renvoie un identificateur lorsqu'un répertoire est créé. Un répertoire ne peut pas être ouvert à partir de son nom, mais seulement à partir de cet identificateur qui doit donc être conservé quelque part pour une réouverture ultérieure. Il est vrai qu'il peut difficilement en être autrement car les noms de répertoire sont volatils puisque l'utilisateur lambda a la possibilité de les modifier depuis l'interface utilisateur. Tout ceci étant valable seulement dans le meilleur des cas, car un répertoire détruit puis recréé ne se verra pas affecté le même id.

5 Générer un e-mail avec fichier attaché

Grâce à l'interface *IMimeMessage*, ce travail est aisé. L'interface s'occupe de gérer les différentes parts du mail: en-têtes, corps,

fichiers attachés et s'occupe de l'encodage. Vous trouverez le code complet sur le Cd Rom accompagnant notre magazine ou sur le site.

La partie difficile de ce code concerne la création du corps (body) du message, c'est-à-dire le texte proprement dit du message. Nous devons utiliser la méthode *SetTextBody* de l'interface *IMimeMessage*. Or cette méthode demande à lire les données constituant le texte depuis un flux Shell (interface *IStream*). Notre exemple écrit les données dans un fichier temporaire puis instancie une *IStream* à partir de ce fichier au moyen de la fonction *SHCreateStreamOnFile*. Si on trouve le procédé du fichier temporaire inacceptable, il est possible de créer directement le flux en mémoire à partir de la fonction *CreateStreamOnHGlobal*. On retrouve une démarche similaire plus loin, après l'ajout au message du fichier attaché. Nous sauvegardons (méthode *Save*) le message entier dans un fichier temporaire, lui aussi chapeauté par une interface *IStream*. Pour la démonstration, le procédé présente l'avantage d'offrir la possibilité de contrôler le travail en ouvrant ce fichier temporaire par exemple dans le Wordpad de Windows. (Fig.2) Si l'on veut se passer du fichier temporaire, on créera un flux en mémoire, à nouveau avec *CreateStreamOnHGlobal*. Dans tous les cas, le travail avec une interface *IStream* est incontournable car la méthode *CommitStream* du répertoire du Windows Mail dans lequel le mail est finalement généré, et par laquelle le mail est effectivement écrit dans ce répertoire, demande elle aussi à travailler avec un *IStream*. Quand tout est terminé, le message peut être visualisé dans Windows Mail, comme tout message propre sur lui :) (Fig.3).

6 Expédier le mail.

Nous abordons notre dernier exemple, *DemoSendMailSMTP.cpp*. Cet exemple commence par générer un message comme ci-dessus, mais sans les en-têtes, car ceux-ci seront définis lors de l'expédition via le protocole SMTP. Disons quelques mots de ce protocole qui est la simplicité même. Ce protocole est défini dans le document RFC2821 qu'il est possible de consulter à <http://www.ietf.org/rfc/rfc2821.txt>. En résumé, lorsqu'un socket client se connecte à un serveur SMTP, en principe sur le port 25, le serveur émet une ligne de texte pour s'identifier. Le client émet ensuite une ligne commençant par HELO ou EHLO lui aussi pour s'identifier. A la suite de quoi le serveur répond, et ainsi de suite. Chaque fois que le serveur répond, son message commence par un nombre qui est un code de succès ou d'erreur. La spécification de l'expéditeur du mail et son destinataire se font de manière similaire à l'identification décrite ci-dessus. L'envoi du mail est un peu différent. On commence par envoyer la commande DATA. Le serveur répond alors qu'il est prêt à recevoir les données. A ce moment il est possible d'envoyer des lignes (qui deviendront des en-têtes) pour définir le sujet du mail, les destinataires d'éventuelles copies, etc. Lorsque l'envoi des données est terminé, on le signale par une ligne contenant seulement un point. Il est possible d'expérimenter avec un simple client telnet. Voici à titre d'illustration le dialogue entre votre serviteur et un serveur smtp, pour lui envoyer un message.

```

telnet smtp.wanadoo.fr 25
220 mwinf2028.orange.fr ESMTP ABO *****
ehlo fred
250-mwinf2028.orange.fr
250-PIPELINING
250-SIZE 10485760
250-AUTH PLAIN LOGIN
    
```

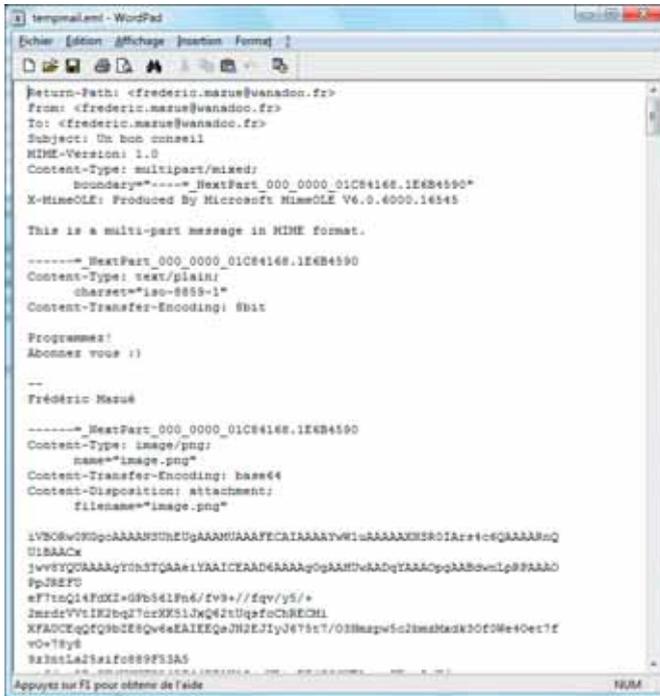


Fig.2 : Notre mail, visualisé dans Wordpad.

```
250-AUTH=PLAIN LOGIN
250 8BITMIME
mail from: <frederic.mazue@wanadoo.fr>
250 Ok
rcpt to: <frederic.mazue@wanadoo.fr>
250 Ok
data
354 End data with <CR><LF>.<CR><LF>
subject: demo telnet smtp
Hello World!
.
250 Ok: queued as D8BC61C00089
quit
221 Bye
```

Deux petites remarques. Toutes les lignes doivent se terminer par un retour chariot ET un saut de ligne. En C++ cela correspond à la chaîne "\r\n". Un simple saut de ligne (\n) ne suffit pas. Ensuite, le document RFC2821 indique que l'identification par HELO reste légale, mais recommande EHLO. Dans ce cas, le serveur répond non par une ligne, mais par un nombre de lignes inconnu à l'avance. Dans une connexion socket, il est possible de savoir s'il existe des données en attente de lecture, c'est-à-dire, dans notre cas, s'il y a des lignes derrière la première ligne émise par le serveur. La classe de socket utilisée pour cette démonstration n'offrait pas de méthode pour le faire, j'en ai ajoutée une :

```
bool Socket::Available()
{
    u_long nb;
    int result;

    result = ioctlsocket(s_, FIONREAD, &nb);
```

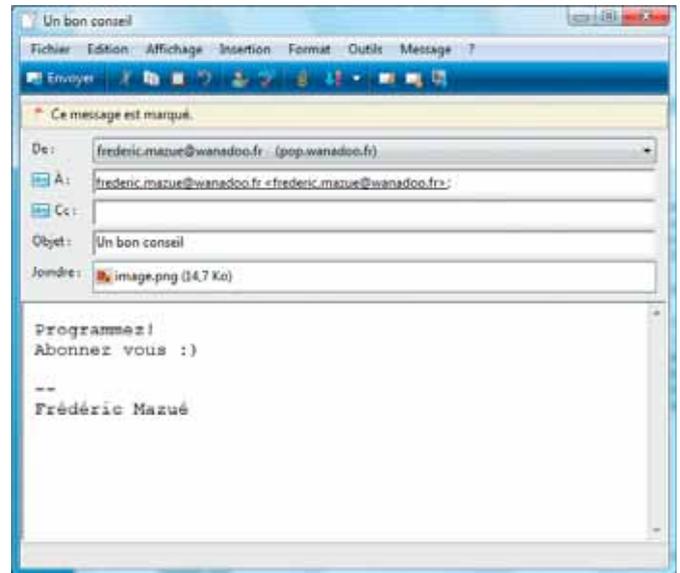


Fig.3 : Notre mail, visualisé dans Windows Mail.

```
if(result != 0)
    throw "Error in Available";
return nb == 0 ? false : true;
}
```

Voici maintenant le code qui expédie un mail, en suivant le protocole décrit plus haut.

```
#include "socket.h"

#include <iostream>
#include <string>
#include <fstream>
#include <sstream>

#include <COMUtil.h>

char* rcpt_to = "rcpt to: <frederic.mazue@wanadoo.fr>\r\n";
char* mail_from = "mail from: <frederic.mazue@wanadoo.fr>\r\n";
const char* smtp_serveur = "smtp.wanadoo.fr";

#pragma comment(lib, "Ws2_32.lib")

void CreateMail();

void CheckSMTP(const std::string& l, int code)
{
    std::stringstream ss(l);
    int returned;
    ss >> returned;
    if(returned != code)
        throw "Erreur dans le protocole SMTP";
}

void SendMessage()
{
```

```

SocketClient s(smtp_serveur, 25);
std::string l = s.ReceiveLine();
CheckSMTP(l, 220);
std::cout << l << std::endl;

//-----
// EHLO
std::cout << "ENVOI ehlo" << std::endl;
s.SendLine("ehlo fred\r");

l = s.ReceiveLine();
std::cout << l;
while (s.Available())
{
    l = s.ReceiveLine();
    std::cout << l;
}

//-----
// MAIL FROM:
std::cout << std::endl <<
    "ENVOI MAIL FROM" << std::endl;
s.SendLine(mail_from);
l = s.ReceiveLine();
CheckSMTP(l, 250);
std::cout << l << std::endl;

//-----
// RCPT TO:
std::cout << "ENVOI TCPT TO" << std::endl;
s.SendLine(rcpt_to);
l = s.ReceiveLine();
CheckSMTP(l, 250);
std::cout << l << std::endl;

//-----
// DATA
std::cout << "ENVOI DATA" << std::endl;
s.SendLine("data\r");
l = s.ReceiveLine();
CheckSMTP(l, 354);
std::cout << l << std::endl;

//-----
// SUBJECT:
// Pas de réponse du serveur ici
std::cout << "ENVOI SUBJECT:" << std::endl;
s.SendLine("subject: Un bon conseil\r");

//-----
// Envoi des données

std::ifstream ifs("G:\\temp\\mail\\tempmail.eml");
if(ifs.good())
{
    std::cout << "envoi donnees proprement dites" << std::endl;

```

```

char c;

while(ifs.get(c))
    s.SendOneChar(&c);
}

//-----
// FIN (.\\r\\n)
std::cout << std::endl << "ENVOI FIN (\\.\\r\\n)" << std::endl;
s.SendLine(".\\r");
l = s.ReceiveLine();
CheckSMTP(l, 250);
std::cout << l << std::endl;

//-----
// QUIT
std::cout << "ENVOI QUIT" << std::endl;
s.SendLine("quit\r");
l = s.ReceiveLine();
CheckSMTP(l, 221);
std::cout << l << std::endl;

std::cout << "Message correctement expedie"
    << std::endl;
}

int main(int argc, char* argv[])
{
    try
    {
        COMInitializer ci;
        CreateMail();
        SendMessage();
    }
    catch(COMException ce)
    {
        std::cout << ce.raison() << std::endl;
        return EXIT_FAILURE;
    }
    catch (const char* s)
    {
        std::cerr << s << std::endl;
    }
    catch (std::string s)
    {
        std::cerr << s << std::endl;
    }
    catch (...)
    {
        std::cerr << "exception innatendue\n" << std::endl;
    }
    return EXIT_SUCCESS;
}

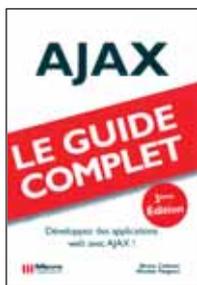
```

Guide du Calcul



- ▶ **Difficulté :** ***
- ▶ **Editeur :** Dunod
- ▶ **Auteur :** collectif
- ▶ **Prix :** 25 €

Vous êtes féru de calculs ? Vous cherchez un logiciel si possible open source, mais vous ne savez lequel choisir ? Il y a l'embaras du choix : XCAS, Scilab, Maxima, etc. Ce n'est pas une simple présentation d'outils mais plutôt un guide. À chaque section (calcul, interface, thèmes...), les auteurs déclinent outil par outil les possibilités, les avantages, les limites des solutions libres. À chaque fois, on trouve des exemples concrets pour chaque outil, des commentaires. Pratique et intéressant à lire.



- ▶ **Difficulté :** **
- ▶ **Editeur :** Micro Application
- ▶ **Auteur :** collectif
- ▶ **Prix :** 15 €

Ajax continue d'être à la mode. Pour preuve, voici un nouveau guide pratique pour le développement sur Ajax. L'ensemble des fondamentaux Ajax sont abordés par les auteurs : votre première page, CSS, Javascript, document HTML, Ajax et les navigateurs, le côté serveur, les aspects asynchrones, l'accès aux données. Mais comme il faut à un moment donné aborder la question des frameworks, l'ouvrage se focalise sur Script.aculo.us, l'une des bibliothèques les plus en vue. On peut regretter l'absence d'un panorama des frameworks et surtout des outils ainsi que le manque d'une section sécurité spécifique. Hormis cela, ce guide complet reste une bonne introduction au sujet.



- ▶ **Difficulté :** **
- ▶ **Editeur :** Eyrolles
- ▶ **Auteur :** Andy Clarke
- ▶ **Prix :** 32 €

Les feuilles de style sont toujours vues comme une corvée, car pas toujours simples à créer, à manipuler. L'auteur se propose ici de vous

Essential Silverlight 2

- ▶ **Difficulté :** ***
- ▶ **Editeur :** O'Reilly
- ▶ **Auteur :** Christian Wenz
- ▶ **Prix :** 28 €

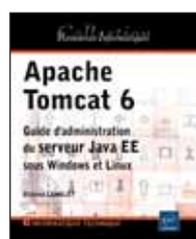


Alors que la v2 de Silverlight, le concurrent de Flex, est en pré-version, on peut d'ores et déjà se familiariser avec la nouvelle plate-forme. Ce n'est pas un livre ordinaire, car cette seconde version évoluant rapidement, vous pouvez mettre à jour l'ouvrage qui se présente sous forme d'un classeur avec des feuilles blanches à imprimer quand le PDF de mise à jour sera disponible ! Une excellente initiative que l'on voudrait bien voir arriver en France ! Sur le fond, l'auteur revient sur les fondamentaux de Silverlight, les outils, le langage XAML, la programmation .net pour Silverlight 2, les API JavaScript, etc. On retrouve la précision O'Reilly avec des conseils, de nombreux codes d'exemples, les pièges à éviter. Un must pour bien débuter. En Anglais.

dépasser en allant au-delà de cette présentation. Superbement illustré, ce livre tente de concilier l'approche designer et développeur web. Le contenu se veut très didactique pour bien comprendre chaque élément, chaque CSS utilisé. À chaque étape, on y explique pour tel choix, le code à mettre en place, etc. Idéal pour le développeur web !

Apache Tomcat 6

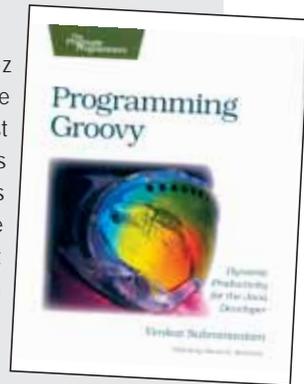
- ▶ **Difficulté :** ***
- ▶ **Editeur :** Eni éditions
- ▶ **Auteur :** Etienne Langlet
- ▶ **Prix :** 27,14 €



On connaît tous Tomcat. Mais sait-on l'administrer sous Windows ? Ce livre ambitionne de vous fournir les fondamentaux pour administrer ce serveur Java EE. On y trouvera donc les parties attendues : installation, configuration, le cluster, installation de MySQL 5, Tomcat et le développement, l'utilisation de LDAP, sans oublier la sécurité SSL, les tests de montée en charge, etc. Vous ne pouvez plus dire que vous ne saviez pas...

Programming Groovy

- ▶ **Difficulté :** ***
- ▶ **Editeur :** O'Reilly
- ▶ **Auteur :** Venkat Subramanian
- ▶ **Prix :** 28 €



Vous ne connaissez pas encore Groovy ? Ce langage dynamique est pourtant l'un des plus puissants et des plus prometteurs du monde Java. On explique tout d'abord ce qu'est un langage dynamique, son utilité, sa structure et la raison d'être de Groovy. Puis on attaque par un classique : l'installation. L'auteur voit Groovy par les yeux d'un développeur Java. Et parle ainsi des fonctions de Java 5 supportées. Puis, on attaque dans le dur : parsing XML, accès aux données, usage des collections et strings, utilisation de GDK, manipulation des Meta Object Protocol, les tests, les DSL et Groovy. En Anglais.

BusinessObject XI : web intelligence

- ▶ **Difficulté :** ***
- ▶ **Editeur :** Eni éditions
- ▶ **Auteur :** Thierry Petibon
- ▶ **Prix :** 27,14 €



Vous verrez comment maîtriser les différentes fonctionnalités de BusinessObjects Web Intelligence (Webi) afin de construire des documents de qualité : création de requêtes, mise en forme de rapports (sections, ruptures, filtres, etc.) analyse des données (exploration ascendante, descendante, etc.), création de graphiques... De nombreux exemples permettent de mieux comprendre chaque situation, chaque fonction. Vous découvrirez aussi des techniques permettant de répondre à des besoins plus complexes d'analyse ou de mise en forme : rapports interactifs avec la notion d'invite, requêtes avancées, utilisation de fichiers de données personnelles, liaison entre sources de données, insertion de calculs dans un document, regroupement de valeurs, etc. Niveau confirmé.

apt-get install obm*



* POUR LES NON "GEEKS" ;-)

OBM est la solution de messagerie, de calendriers partagés et de travail collaboratif de LINAGORA. Cette solution est dorénavant disponible sur la dernière Ubuntu 8.04. Un simple "apt-get" vous permet d'installer OBM depuis les dépôts officiels (universes).

Un grand merci à nos équipes de packageurs notamment Anthony², Medhi, Sylvain, Thomas, Nicolas et Pierre, ainsi qu'à toute la communauté Ubuntu.

Venez contribuer sur
WWW.OBM.ORG

CHERS CLIENTS, VENEZ NOUS RENCONTRER LORS DE NOS "MATINÉES POUR COMPRENDRE ..."

LA PUISSANCE DES OUTILS LIBRES POUR LA GESTION DE VOTRE PARC INFORMATIQUE :

- 12 juin Paris
- 19 juin Toulouse
- 26 juin Lyon

ADMINISTRER, SUPERVISER ET EXPLOITER LES SYSTÈMES LIBRES :

- 10 juillet Paris
- 17 juillet Toulouse
- 24 juillet Lyon

Séminaires gratuits - Plus d'informations sur
www.linagora.com

LINAGORA
GROUPE

Votre potentiel, notre passion.[™]

Microsoft®

Microsoft
Visual Studio

**SATISFAIRE CHACUN, SANS COMPROMIS.
VISUAL STUDIO® 2008.**

Votre défi : concevoir de meilleures applications pour la bureautique et le web dans les meilleurs délais et presque sans effort.

Votre arme : Visual Studio 2008. Pour gagner en temps, en expérience utilisateur et en qualité, tout simplement. Plus d'informations sur www.relevetouslesdefis.com

RELEVEZ TOUS LES DEFIS