

BORLAND :
TURBO le retour
aux sources



Vista

Peut-on
encore
utiliser

Win32?

Téléphonie sur IP

Programmez un Client VoIP
en Open source avec Asterisk ou en Java



Développer avec AJAX

Comparatif : 4 outils au banc d'essai

WEB SERVICES

Créer mes premiers
Web Services.

La jungle
des standards.

.Net

- ✓ Des fonctions
GPS en VB

Java

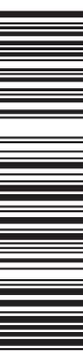
- ✓ Utiliser Groovy
- ✓ Découvrir
Bea Workshop

Linux

- ✓ Développer
avec Anjuta

Méthodologie :
Comprendre CMM-i

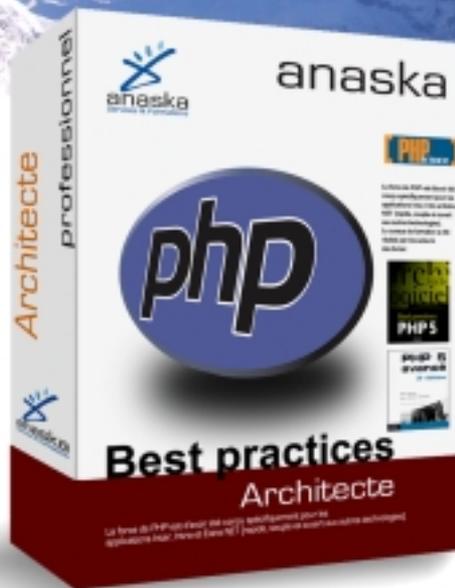
Technique :
Gérer les exceptions



anaska

Formation

photographe : benoit Thinet



PHP Architecte Bonnes pratiques

“ Concevez vos applications de façon optimale ”

AU PROGRAMME :

- * Mise en place d'une architecture scalable et optimale
- * La programmation orientée objet en pratique
- * Les motifs de conception (design pattern)
- * Optimisation de l'environnement de travail
- * Travail en équipe et environnement global
- * Les frameworks

La formation a été
réalisée par l'auteur du
livre

“Best Practices PHP 5”



anaska
Conseil & formation en logiciels libres

Nos parcours de formation PHP

PHP est devenu la plateforme de développement web la plus utilisée par les entreprises. Nos formations vous accompagnent de la prise en main à des niveaux plus experts.

Les parcours PHP

Les références



PHP Maîtrise



PHP Expert



PHP Tuning



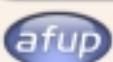
Best practices



PHP5 Avancé



Best practices
PHP5



Anaska adhère à l'Association
Française des Utilisateurs de PHP

Informations : 01 45 28 09 82
<http://www.anaska.com>



Anaska adhère à l'Association
des Sociétés de Services en Logiciels Libres

SOMMAIRE

N°91

> Actus

L'actualité en bref	6
Agenda	6

> Evénements

Borland fait renaître Turbo	10
IBM complète Rational v7 avec Caspian	11

> SGBD

Brèves	12
SGBDO, SGBDRO, SGBDR: l'objet au cœur de la donnée ?	14

> Projets

Le modèle CMM-i	16
Red Hat déploie Linux chez Lapeyre	18

> Gros plan : VoIP, mettez de la voix dans vos applications

Voix sur IP : cap sur la qualité !	21
Un client IAX avec la librairie IAXClient	22
La VoIP avec Java Media Framework	24

> Test

Les outils pour le web 2	28
--------------------------------	----

> Dossier : Windows Vista et .NET 3 [2^e partie]

La compatibilité Win32 en question	33
Le framework ADO.Net Entity	38
Introduction à Windows Presentation Foundation (2e partie)	42

> Dossier : Web Services, comprendre et construire

Web Services : une définition	47
Mes premiers pas en services web	51
Interopérabilité des web services entre .Net et Java	55
Les standards des Web Services	57

> Carrière

L'EPSI : de la passion de l'informatique au diplôme d'ingénieur	58
---	----

> Technique

Un plug-in Eclipse pour Team System	60
La gestion des exceptions	61

> Code

WebLogic Workshop : l'usine à SOA	63
Développement Linux avec Anjuta et les bindings GTKmm et SDLtnet	66
Utilisation d'un GPS en Visual Basic.Net pour Pocket PC	69
Développer un plug-in pour Netbeans 5.0 (2e partie)	72
A la découverte de Groovy	76

> Temps libre

Ludique	80
Livres	82



CD-Rom 91 Programmez !

Microsoft .Net Framework 3.0 RC 32 bits

Package nécessaire pour faire fonctionner sous Windows des applications .NET. Release Candidate de la version 3.0 pour processeurs 32 bits. Aucune limitation.

Microsoft .Net Framework 3.0 RC 64 bits

Package nécessaire pour faire fonctionner sous Windows des applications .NET. Release Candidate de la version 3.0 pour processeurs 64 bits. Aucune limitation.

Asterisk 1.4.0 - bêta 2

Asterisk est un logiciel Open Source qui transforme un PC sous Linux en standard téléphonique IP. Aucune limitation.

Adobe Contribute 4 pour Windows et Mac

Outil de publication et de mise à jour de sites Web d'Adobe
Limitation : 30 jours

Adobe Contribute Publishing Server 1.11 pour Windows et Linux

Contribute Publishing Server (CPS) est une application serveur qui fournit les fonctions d'administration et de suivi des activités de publication au niveau du site web. Limitation : 30 jours

VIGUARD 12

antivirus HIPS, sans mise à jour de signatures
Limitation : 15 jours

Kdevelop 3.4

IDE gérant de nombreux langages comme le C, C++, Java et Ruby

AjaxAnywhere 1.2 RC2

AjaxAnywhere permet d'ajouter un comportement Ajax à des pages réalisés avec des composants JSP ou JSF.

Yale : outils de données data mining

Zk 2.1.2

Framework Ajax sans javascript

SharpDevelop 2 2.1

IDE Open Source pour les projets C#, VB.Net et Boo (executable et sources)

Wamp5 1.6.5

WAMP5 est une plate-forme de développement Web sous Windows, comprenant Apache, PHP et MySQL

ADD-ON PHP4.4.4 pour WAMP5 1.6.5

Cet add-on installe PHP4.4.4 sur votre WAMP5

ADD-ON MySQL administration toolkit pour WAMP5 1.6.5

Cet add-on ajoute MySQL Administrator et MySQL Query Browser dans le menu de gestion MySQL de WAMP5

ADD-ON ACTIVESTATE PERL pour WAMP5 1.6.5

Cet add-on installe ACTIVESTATE PERL sur WAMP5

ADD-ON ZEND OPTIMIZER pour WAMP5 1.6.5

Cet add-on va vous permettre d'accélérer considérablement les performances de votre serveur

ADD-ON WEBALIZER pour WAMP5 1.6.5

Cet add-on installe l'outil de statistiques Webalizer sur WAMP5

ADD-ON XDEBUG pour WAMP5 1.6.5

Cet add-on permet d'ajouter l'extension xdebug ainsi que l'application WincacheGrind à WAMP5. Il vous sera ainsi possible de faire du profiling sur vos scripts afin de trouver leurs failles et de les optimiser

SweetDev RIA

bibliothèque de composants Ajax

Leonardi Free

Leonardi est un framework MDA en Java/XML permettant d'automatiser la production d'interfaces homme-machine adressant des systèmes d'information complexes et évolutifs. Leonardi permet de créer des IHM pour plusieurs cibles d'affichage (AWT, Swing, SWT, HTML, DHTML, SVG) et d'interfaçage avec différents types de sources de données (SGBD, serveurs d'application, LDAP, Corba, XML, CSV...).

EN ROUTE POUR WINDOWS VISTA ? PRENEZ LA BONNE VOIE. DÈS MAINTENANT.

Vous vous préparez pour la sortie de Windows Vista ? Faites-le avec Qt, l'environnement multi-plateforme de développement C++ d'applications riches. Depuis 11 ans, Qt est à la pointe de la technologie sur les plateformes Windows, mais aussi sur Mac®, et Linux®. C'est donc tout naturellement que les applications créées avec Qt sont déjà compatibles avec les releases Vista les plus récentes. Des milliers d'utilisateurs Qt – parmi lesquelles des sociétés comme Google®, Adobe® ou Synopsys® - peuvent ainsi pleinement profiter des avantages qu'offrent les technologies innovantes Windows dans les années à venir.

Et ça, c'est une bonne nouvelle.

Pour en savoir plus, téléchargez dès aujourd'hui votre version gratuite d'évaluation:
www.trolltech.com/vista

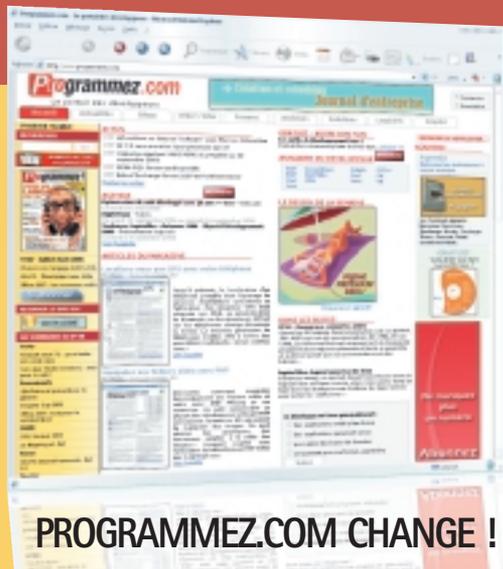
Qt® BY TROLLTECH
CODE LESS. CREATE MORE.



Optimized for

 **Visual Studio.net**

Trolltech, Qt, and the Qt logo are registered trademarks of Trolltech AS in Norway and/or other countries worldwide. All other trademarks are property of their respective owners.



PROGRAMMEZ.COM CHANGE !

Programmez!

LE MAGAZINE DU DÉVELOPPEMENT

Rédaction : redaction@programmez.com

Directeur de la Rédaction : Jean Kaminsky

Rédacteur en Chef : François Tonic.

Ont collaboré : C.P. de Geyer, X. Leclercq, J.M. Maman, F. Mazué, C. Padych, A. Zanchetta, G. Delamarre, P. Manac'h, F. Santin, E. Chaysinh, R. Braun, P. Guedez, F. Le Droff, S. Sibué, C. Hartz, A. Desasteker, E. Margraff, D. Payet, O. Smedile, J. Chable.

Dessins : Michel Piédouze, Jissey.

Maquette : AJE Conseils

Publicité : régie publicitaire : ECC, Stéphane Kaminsky - Pour la publicité uniquement :
Tél. : 01 49 76 06 42 - Fax : 01 49 76 07 35
publicite@programmez.com.

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

Directeur de la publication : Jean-Claude Vaudecrane.
Le numéro comporte un CD Rom.

Abonnement : Programmmez 22, rue René Boulanger, 75472 Paris Cedex 10 - abonnements.programmez@groupe-gli.com - Tél. : 01 55 56 70 55 - Fax : 01 55 56 70 20 - du lundi au jeudi de 9h30 à 12h30 et de 13h30 à 17h00, le vendredi de 9h00 à 12h00 et de 14h00 à 16h30

Tarifs abonnement (magazine seul) : 1 an - 11 numéros - France métropolitaine : 45 € - Etudiant : 39 € - CEE et Suisse : 51,83 € Algérie, Maroc, Tunisie : 55,95 € - Canada : 64,33 € Tom : 79,61 € - Dom : 62,84 € - Autres pays : nous consulter.
PDF : 35 € (Monde Entier) souscription en ligne.

Le serpent se mord-t-il la queue ?



Avec l'emballage des annonces, on assiste à une course aux fonctionnalités de la part des éditeurs. Ils veulent rattraper le concurrent d'en face, étoffer leur offre par rapport à l'open source, montrer aux marchés et aux utilisateurs, qu'ils sont capables d'apporter du sang neuf à leurs outils. Au risque de fournir un environnement de travail ressemblant à un empilement de fonctions, manquant parfois de cohérence. Et rien que pour installer, configurer et lancer l'outil, le développeur doit subir 2 à 3 jours de formation avant de créer son Hello world. On peut alors se demander s'il n'y a pas un décalage temporel entre l'apparition de la nouveauté et son utilisation dans le monde réel.

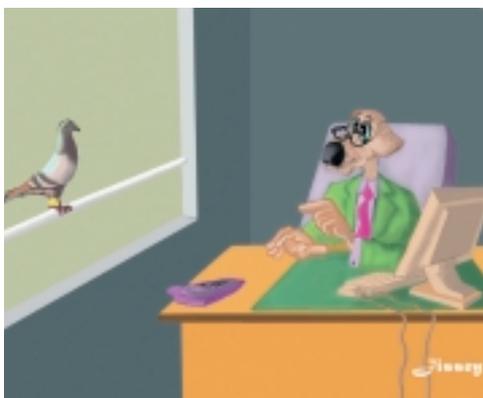
Comme le serpent qui se mord la queue, à force de suivre l'autre, l'éditeur ne perdrait-il pas le contact avec le développeur, tout en mettant en péril, sur le long terme, l'existence même de ses produits ?

Si on remonte à la sortie de Java 5 (Tiger), les éditeurs -qu'ils soient open source ou commerciaux -avaient mis plusieurs mois à intégrer la nouvelle version du langage alors qu'une attente existait auprès du développeur.

Borland, avec la résurgence de la gamme Turbo, proclame son désir de retrouver le développeur et de mieux prendre en compte le retour terrain de leur part, avec des outils plus simples, pas chers. L'éditeur, depuis quelques années, s'était délibérément tourné vers le cycle de vie, au risque de se couper d'une partie des développeurs et l'annonce de la cession de ses IDE avait jeté un certain froid, qu'il faut maintenant faire fondre. L'argument de l'IDE gratuit n'est qu'un argument, car ce n'est pas parce que c'est gratuit que le développeur va se jeter dessus, il faut que l'outil lui apporte quelque chose !

Nous pensons que l'approche modulaire d'un Eclipse représente l'avenir des outils de développement. MySQL rejoint cette approche et envisage plus de modularité, en laissant à l'utilisateur le choix des moteurs, des plug-ins... Il s'agit d'avoir une base technique commune à tous les outils et ensuite, l'utilisateur choisit les plug-ins nécessaires pour pouvoir mettre en oeuvre telle ou telle fonction. La valeur ajoutée n'est plus dans l'IDE en tant que tel mais dans les fonctions. Mais il faut aussi que les plug-ins soient de qualité et qu'ils soient capables d'interopérer ensemble... L'empilement des fonctions aurait-il encore de l'avenir ?

■ François Tonic



MESSAGE

VOUS TOMBEZ BIEN : J'AI, DEPUIS 15 JOURS,
UN MESSAGE À TRANSMETTRE
À MON FOURNISSEUR D'ACCÈS INTERNET

Souriez avec Jissey

www.programmez.com

chaque semaine, un gif animé

PROCHAIN NUMERO - N°92 - décembre 2006 - Parution le 30 novembre 2006

Microsoft **OFFICE**

La nouvelle version de la suite bureautique de Microsoft arrive. Comment personnaliser le Ruban Office, quelles sont les nouveautés de VSTO 3.0, comment profiter des fonctions avancées de Sharepoint Services ?

PHP

Découvrez toute l'actualité du langage et les meilleures techniques pour optimiser vos applications PHP. Choisir les outils PHP.

Mon NetBeans idéal

Comment bien utiliser NetBeans, les meilleurs plug-ins, comment compiler sa propre distribution NetBeans.

SOLUTIONS

IBM dévoile de nouveaux outils

Outre Baltic et Caspian (cf p.11), IBM a procédé à de nombreuses annonces depuis la rentrée. Voici ce qu'il faut retenir :

- Une messagerie nomade : avec l'évolution des solutions Notes, IBM a annoncé une messagerie nomade à partir de **Notes / Domino**. Il est possible avec les dernières évolutions de personnaliser Notes sur un petit matériel USB de type clé ou même iPod et de le connecter à un PC pour créer une réplique de son environnement Notes et se connecter à sa messagerie comme cela !
- Sur le front du **mainframe**, début octobre, le constructeur a annoncé un ambitieux plan de 5 ans pour simplifier et ouvrir cette plate-forme à de nouveaux marchés. Cela passera par une simplification de la configuration, une interface modernisée, un meilleur contrôle des coûts et un environnement de développement entièrement rénové.
- Sur la partie **Workspace**, IBM lance Workspace Web Content Management 6.0. L'outil doit permettre de créer et de gérer du contenu. Pour son déploiement, l'outil s'intègre à **WebSphere Portal**. Cette version intègre Java Content Repository pour le stockage des données. IBM a fourni un effort sur la simplification de la création de contenus.

Enfin, IBM a renforcé son offre **SOA** et notamment sur les processus métier. Pour ce faire, on dispose maintenant de **BPM Methodology** et **BPM Competency Center**. Pour l'administration, on dispose de **Workplace Dashboard Framework** pour la réutilisation de composants. Il propose aussi des outils d'administration.

Forrester, cabinet d'études et conseils, a été, dans une de ses dernières études, sévère sur la modélisation de processus business proposée par IBM. Le cabinet pointe du doigt : une architecture pauvre, un manque de modèles, et l'absence d'un processus spécifique de référentiel et de superposition avec les outils de modélisation d'IBM Rational. IBM a parfois du mal à bien faire interagir ses différentes solutions. Une bonne intégration des solutions SOA avec les outils Rational permettrait de proposer une solution intégrée de bout en bout et de bénéficier d'un unique référentiel à l'ensemble des outils. Même si sur la partie gouvernance / cycle de vie, WebSphere Registry and Repository se base sur Rational BuildForge, ClearQuest et Software Architect.

SÉCURITÉ

Aladdin se met à .Net 2

Aladdin, pour sa famille HASP, prend en charge le framework .Net 2.0. Il devient possible d'utiliser dans des applications .Net les mécanismes de protections de HASP. " La prise en charge de Microsoft .NET Framework version 2.0 par les produits Aladdin est vitale pour le succès continu de nos solutions protégées ", a déclaré Todd Mancini, Directeur de la technologie chez Liska Biometry, important développeur de systèmes d'identifiants biométriques et sécurisés basé à Dover, dans le New Hampshire (Etats-Unis). " L'assistance fiable et permanente de l'équipe Aladdin HASP renforce notre volonté de travailler avec les toutes dernières technologies. La rapidité de la prise en charge de .NET Framework version 2.0 est rassurante. Cette nouvelle est importante pour nous, mais elle l'est encore plus pour nos clients. "

Site : www.aladdin.fr/HASP

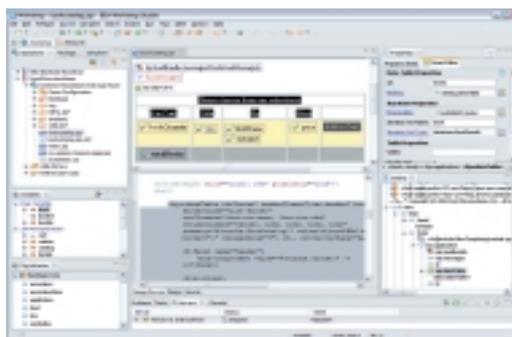
INTÉGRATION

Software AG complète Crossvision

La gamme Crossvision se complète de Legacy Modernization. L'environnement doit aider à migrer, moderniser les applications héritées en les rendant plus modulaires. L'annonce permet à Software AG de compléter son offre SOA avec de l'intégration legacy. L'entreprise pourra, sans trop bouleverser ses applicatifs s'ouvrir au monde web, SOA, intégrer les données à SAP. Site : www.softwareag.com

WEB

OpenAjax se renforce



L'alliance OpenAjax, ayant pour ambition de définir un Ajax standard, commence à réellement prendre forme. Depuis février 2006, le nombre d'éditeurs adhérents a doublé. L'objectif est désormais de sortir une première version d'OpenAjax Hub début 2007. OpenAjax Hub doit servir de pierre fondatrice au travail de l'alliance en définissant un ensemble de fonctions standard de JavaScript, tout en poussant à une forte interopérabilité entre les différentes bibliothèques Ajax. L'interopérabilité est l'un des objectifs majeurs de l'alliance. Reste à savoir si OpenAjax Hub et les futures spécifications pourront fédérer les nombreux éditeurs et projets open source. Site : openajax.org

Agenda

NOVEMBRE

Lyon, le **mardi 7 nov 2006**

Paris, le **mercredi 22 nov 2006**

Mini Mix, après-midi techniques organisées par Microsoft

Le web nouvelle génération, quel impact pour le développeur web?

<http://www.microsoft.com/france/msdn/minimix/default.mspx>

Les **9 et 10 novembre 2006**, Paris 7ème, 84 rue de Grenelle

Forum PHP 2006, 6e édition, organisé par l'association des utilisateurs PHP www.afup.org

Du **07 au 08 novembre 2006**

Paris-Expo, Porte de Versailles

Convention VoIP 2006, le rendez-vous des professionnels de la voix et de la vidéo sur IP www.conventionvoip.com

Du **7 au 9 novembre 2006**

Paris Nord Villepinte,

Cartes 2006, 21e édition du salon des professionnels de la carte à puce et de l'identification <http://www.cartes.com/en/2006/index.htm>

Le **16 novembre 2006**, Paris 9e, 97 rue St Lazare

Pourquoi et comment migrer vers BusinessObjects XI

Séminaire de présentation des nouveautés de la version XI

http://www.synergy.fr/seminaire_BOXI_migration.htm

13 au 17 novembre

Cannes,

Sommet mondial sur les technologies IPv6 (nouvel internet), avec un panel d'intervenants extrêmement qualifiés. <http://www.v6summit.com>

Du **22 au 23 novembre 2006**

CNIT - PARIS la Défense

• **Salon de la Sécurité Informatique 2006**, le rendez-vous incontournable des spécialistes de la sécurité informatique. www.infosecurity.com.fr

• **Storage expo**

<http://www.storage-expo.fr/>

INTERNATIONAL

Du **7 au 10 novembre 2006**

Espagne, Barcelone

Microsoft TechEd

<http://www.mseventseurope.com/TechEd/06/pre/defaultdev.aspx>

Pour le XML choisissez la qualité



Avec la suite XML d'Altova, passez à la vitesse supérieure et économisez la moitié du prix des meilleurs outils en matière de développement XML.

Inclus dans la suite XML 2007 d'Altova:

- Altova XMLSpy, MapForce, et StyleVision éditions Entreprise ou Professionnelle
- Plus Altova SchemaAgent, SemanticWorks, et DiffDog avec l'édition Entreprise
- Recevez également un exemplaire GRATUIT d'Altova DatabaseSpy 2007 pendant une durée limitée*

La suite XML 2007 d'Altova comprend les toutes dernières versions des outils de développement XML les plus évolués du marché, le tout sous une offre attractive. Elle contient Altova XMLSpy, l'environnement de développement XML de référence ; MapForce, le célèbre outil d'intégration de données et d'implémentation de services Web ; enfin, StyleVision, le meilleur logiciel de conception graphique de feuilles de style. En outre, l'édition Entreprise comprend des outils de gestion de schémas XML et de web sémantique, ainsi que des programmes de différenciation compatibles XML.

Réduisez le montant de votre achat! **Download the Altova XML Suite today: www.altova.com**



***Offre spéciale:** jusqu'au 24 décembre 2006 pour tout achat ou mise à niveau de la suite XML d'Altova, recevez la nouvelle version de DatabaseSpy, l'outil d'interrogation et de création de bases de données, GRATUITEMENT !

Sous réserve de certaines conditions.
Consultez le site Web pour plus de détails.

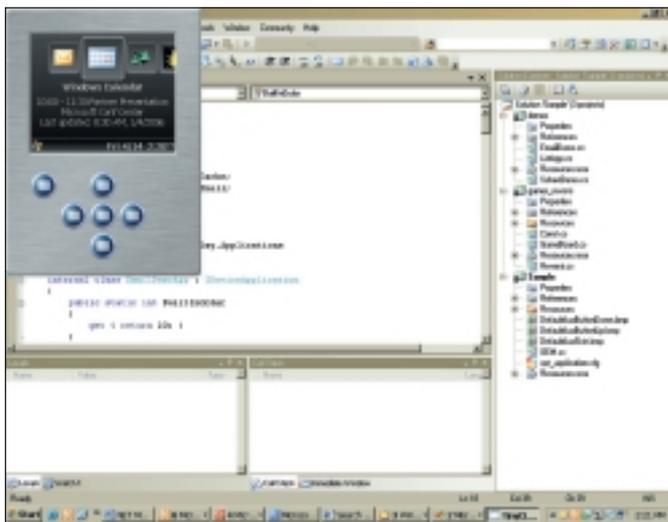
CONFÉRENCE

Eclipse Now You Can : une première à Paris !

Mi-octobre se tenait à Paris, une journée entièrement dédiée à Eclipse, une grande première en France. Plutôt orientée " Eclipse verticalisée ", la journée fut l'occasion de constater toute l'importance de la plateforme dans les environnements verticaux (santé, industrie, aéronautique, etc.). Organisée activement par Tni software, la journée rassembla une centaine de personnes. La matinée fut consacrée aux conférences alors que l'après-midi proposait des ateliers systèmes embarqués et applications business. Ce fut l'occasion de rappeler les apports d'Eclipse dans le monde du développement. Une bonne occasion de voir concrètement Calisto et le projet Jazz d'IBM. Ralph Müller, directeur Europe de la fondation Eclipse rappelait aussi que le véritable concurrent d'Eclipse n'était pas NetBeans mais Visual Studio et .Net. Et l'intégration de Mono dans Eclipse pour prendre en compte .Net ? " Techniquement c'est possible. Ce serait (même) une bonne chose " note Ralph. Cependant, une telle intégration se décide au niveau du comité Eclipse. On retiendra de cette journée l'importance, pour les industriels, de mettre en place un environnement collaboratif ouvert pour pouvoir communiquer, échanger avec l'ensemble de son écosystème à partir d'une unique plate-forme technique, grâce à un socle commun. Pour 2007, les organisateurs espèrent bien renouveler la journée avec une dimension technique, en plus.

FRAMEWORK

Microsoft prépare le micro .Net framework



Alors que l'éditeur travaille à sortir .Net 3.0 et à préparer la future version, provisoirement numérotée 3.5, du framework prévue pour 2007, Microsoft met au point en ce moment, une nouvelle édition de .Net, le micro .net framework. Il s'agit de créer un framework (très) simplifié nécessitant une mémoire la plus faible possible, afin de l'embarquer dans des terminaux ayant peu de ressources matérielles. La version bêta est disponible depuis fin septembre dernier. La programmation se fera via Visual Studio. Le micro framework complète l'offre embarquée de Microsoft qui comprend XP Embarqué, Windows CE / Mobile et le

Compact Framework. Le micro framework est un sous-ensemble du framework et nécessitera environ 300 Ko de mémoire. Pour le développement, un émulateur est disponible pour tester ses applications. Site : <http://www.aboutnetmf.com>

WEB

Contribute en version 4.0

Adobe vient de rendre disponible la 4e version de Contribute, l'environnement pour créer et déployer rapidement du contenu web. Contribute permet de mettre à jour rapidement le contenu, texte, images, vidéo, Flash... sur des sites et blogs, sans risque de compromettre le code de ses pages web. L'auteur a seulement accès au périmètre que l'administrateur veut bien lui accorder. Pour proposer Contribute à l'échelle d'une entreprise, on peut mettre en place Contribute Publishing Server. Celui-ci centralise l'administration de Contribute et permet une gestion fine des autorisations de mise à jours des sites via Contribute. Site : www.adobe.com

ARCHITECTURE

BEA renforce son SOA

Avec la sortie de Web Logic Platform 9.2, BEA veut poser des fondations SOA robustes. Platform 9.2 inclut WebLogic Integration et surtout SOA 360°. Ce dernier permet de déployer du SOA autour d'une architecture unifiée de bout en bout. Pour ce faire, SOA 360° repose sur Tuxedo, WebLogic et AquaLogic. Côté architecture, BEA met en place MicroService Architecture. Cette solution middleware se veut à la fois légère et standard (implémentations des standards SOA actuels). BEA mixe dans 360° des solutions ouvertes et commerciales. À noter, la disponibilité de BEA Kodo 4.1 assurant la persistance en Java (JPA pour être plus précis). Il supporte les EJB 3 de Java EE 5. Cette version de Kodo sera prise en charge par le prochain WebLogic Server qui doit sortir d'ici la fin de l'année. En même temps, BEA a mis à jour son outil de développement Workshop en supportant Eclipse 3.2.1. Site : www.bea.com

PHP fait salon à Paris les 9 et 10 novembre 2006

Le forum PHP 2006 se tiendra les 9 et 10 novembre à Paris. Des intervenants de renom tels que le créateur de

**FORUM PHP
Paris 2006**



PHP Rasmus Lerdorf, Google et Yahoo! seront présents. Ce forum s'adresse à tous ceux qui développent des applications Web. La première journée est axée sur le domaine fonctionnel, la deuxième est orientée technique. De nombreux domaines seront abordés : les bonnes pratiques en PHP, l'intégration PHP / Java, les motifs de conception, le futur de PHP. Le forum PHP organisé par l'Association Française des Utilisateurs de PHP (AFUP) est une occasion unique de rencontrer la communauté et les professionnels de PHP. Plus d'informations sur <http://www.afup.org>

Le premier service pack pour Visual Studio 2005...

Presque un an après la sortie de Visual Studio 2005, Microsoft vient d'annoncer la proche mise à disposition d'un service pack 1. En fait il ne s'agit pas que d'un seul service Pack mais d'un ensemble adapté à chacune des différentes éditions de la gamme Visual Studio 2005. Tout le monde aura donc droit à ses correctifs et à ses améliorations, de la simple édition Express au Serveur Team Foundation.

Le service pack est aujourd'hui disponible en téléchargement pour tous dans sa version Bêta 1, en 7 versions : une pour chacune des 5 éditions Express de Visual Studio 2005 (Visual Basic, Visual C++, Visual C#, Visual Web Developer, Visual J#), une pour les versions complètes (Professional, Standard, les Team Editions, Team Explorer...) et une pour les composants serveurs (Team Foundation Server, Team Build Server, Team Foundation Server Proxy).

Coté Team Foundation, plusieurs problématiques apparues avec les dernières technologies Microsoft seront prises en compte :

- Utilisation possible de Microsoft Office 2007 Project et Excel pour se connecter aux serveurs et utiliser le suivi d'éléments de travail
- Compatibilité des projets utilisant le nouveau modèle " Web Application Project " avec le gestionnaire de configuration

En plus de corriger divers problèmes, le SP1 ajoute un lot de nouvelles fonctionnalités :

- Prise en compte et support de l'authentification Basic et Digest pour se connecter au serveur Team Foundation, afin de faciliter les connexions au serveur dans les contextes hors réseau privé virtuel(VPN).
- Possibilité de personnaliser les types d'éléments de travail (Work Items) avec des formulaires personnalisés. De base, juste quelques champs étaient disponibles : simple champ, calendrier, champ multi lignes..., il sera maintenant possible d'ajouter vos propres contrôles.
- Nouvelle gestion de l'historique pour la partie " fusion de branche ". Il sera maintenant possible de reconstruire complètement l'historique d'évolution d'un fichier dans le gestionnaire de configuration, y compris au travers des branches et des fusions.
- D'un point de vue installation, il sera maintenant possible d'installer la partie Data Warehouse de SQL Server 2005 (utilisée pour la génération de rapports) sur un serveur différent de celui de la base relationnelle, permettant ainsi d'accroître les performances en séparant la charge de travail.

Pour terminer, quelques problèmes issus de la première version seront corrigés : 15 autour d'Office, 8 côté génération de rapports, 8 d'un point de vue administration, 38 pour le gestionnaire de configuration et 13 concernant le suivi d'éléments de travail.

■ Florent Santin



Une étude IDC montre que 30 % des logiciels sont utilisés de façon illégale. Les conséquences? Vous perdez des bénéfices, vous devez surinvestir en R&D et vous exposez inutilement votre propriété intellectuelle. Êtes vous certain que toutes les copies de votre logiciel ont été payées? Avec HASP HL vous pouvez en être sûr! HASP HL d'Aladdin protège votre logiciel, votre propriété intellectuelle et vos revenus tout en vous fournissant des modèles flexibles de distribution adaptés à tous vos besoins.

Rejoignez le numéro 1, comme les 30 000 sociétés les mieux protégées à travers le monde l'ont déjà fait.

Pour plus d'information, livres blanc, kit de développeur gratuit, rendez-vous sur: www.Aladdin.fr/hasp

Aladdin Western Europe Ltd, 205 av Georges Clémenceau, 92024 Nanterre Cedex
Tel : +33 1 41 37 70 30 • Fax : +33 1 41 37 70 39

North America | UK | Germany | France | Benelux | Spain | Italy | Israel | China | Brazil | Japan

Aladdin.fr/HASP

HASP[®] HL
REINVENTING SOFTWARE PROTECTION & LICENSING



Aladdin[®]
SECURING THE GLOBAL VILLAGE

Borland fait renaître Turbo !

Petit rappel : le 8 février dernier, Borland annonçait la mise en vente de tous ses environnements de développement pour pouvoir se concentrer sur les outils de cycle de vie. Ne pouvant pas assurer les deux marchés de front, l'éditeur faisait un choix historique.

Pour préparer cette cession, l'éditeur créait une filiale spécialisée sur les IDE, le Developer Tools Group. La 3e étape fut la présentation des IDE Turbo, disponibles depuis quelques semaines.

Turbo rappellera, pour les plus nostalgiques, Turbo Pascal. Cette nouvelle gamme reprend le concept des versions gratuites personnelles que Borland avait par le passé mis à disposition des développeurs. À l'image de Microsoft et de sa gamme Express, la gamme Turbo propose les versions gratuites des langages : Delphi, Delphi for .Net, C++ et C#. Un support de 1er niveau est offert (installation et configuration). Le niveau supérieur demeure payant. Aucun Turbo Java n'est disponible pour le moment et rien ne semble prévu avant un certain temps. Si on souhaite utiliser un JBuilder gratuit, Borland propose toujours en ligne JBuilder Foundation.

L'objectif de Turbo est de proposer à tout le monde "un socle commun, une offre gratuite pour démarrer dans la programmation" précise Gilles Dumontier, le responsable français. Si le développeur veut passer à la vitesse supérieure, il doit accéder aux versions payantes de Turbo ou acquérir Borland Developer Studio, environnement multi langage de Borland.

La philosophie de Turbo est de revenir à des environnements plus simples, plus épurés, évitant d'investir des centaines, voire, des milliers d'euros. " Il s'agit d'offrir une réelle alternative " précise G. Dumontier. Sur le marché Windows, Borland est coincé au

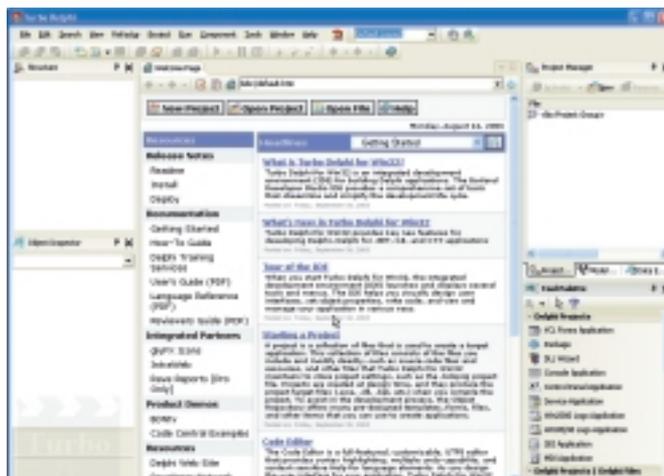
côté des outils Microsoft et sur Java, Eclipse écrase tout ou presque, au point que le prochain JBuilder utilisera Eclipse.

Un seul Turbo

Si l'initiative redonne corps à la stratégie IDE de Borland, en attendant la vente définitive à des investisseurs et non à un éditeur (la vente pourrait se finaliser très rapidement), il ne s'agit que d'outils légers avec des fonctionnalités limitées. Ainsi, il ne sera pas possible d'importer des projets Delphi sur Turbo Delphi, par contre, l'inverse ne causera aucun souci. De plus, il ne sera pas possible de coupler Turbo à un outil de modélisation de type Together. L'élément le plus surprenant est l'impossibilité d'installer plusieurs versions de Turbo sur un même poste. Il faudra à chaque fois désinstaller la première version de Turbo pour installer un nouveau langage. À l'heure actuelle, nous ne savons pas si cette limitation sera levée dans les prochains mois.

Un plan pluriannuel

Le Developer Tools Group travaille sur le long terme et pas uniquement sur la gamme Turbo. Ainsi, dans la foulée des IDE langages, la filiale de Borland annonçait InterBase 2007, la base de données multi-plate-forme et embarquée de Borland. Cette version propose les éditions serveur et desktop. Parmi les nouveautés, on trouve le support d'Unicode et de nouveaux mécanismes de protections de données. Le Group proposera aussi une édition pour Borland Developer Studio.



Sur les produits, le Group met en place un ambitieux plan pour les prochaines années :

- JBuilder : IDE Java historique, JBuilder a beaucoup souffert d'Eclipse. Pour reprendre du terrain, le Group va sortir à la fin de l'année (si tout va bien), JBuilder 2007 (alias Peloton) basé sur Eclipse avec le support de Java EE 5, des EJB 3, un designer SWT et Swing, un module de travail P2P. Derrière, les équipes prévoient déjà JBuilder 2008 et 2009. Les focus seront un développement web intégré via un RAD, l'implémentation de frameworks et outils open source, du SOA, une meilleure intégration au cycle de vie.

- Delphi : l'un des focus sera le support de .Net 2 et du 64 bits. Ceci est prévu dans Delphi Highlander (sortie début 2007 ?). Derrière, ce sera Delphi Vista avec le support des frameworks de .Net 3 (pas avant le 2e semestre 2007). En 2008, on attend un Delphi/C++ pour Windows 64... Par contre, toujours aucune indication sur un retour de Linux !

- Sur InterBase, la prochaine étape sera fin 2006 avec la présentation de Trailblazer, puis pour les prochaines années Vault et Voyager incluant des fonctions (hypothèses) : support multi langage de développement pour les procédures stockées, du 64-bit.

Disponibilité

Les versions de Turbo sont disponibles immédiatement en téléchargement sur le site de Turbo Explorer (www.turboexplorer.com). Les versions professionnelles seront vendues à 399 dollars US. Le prix éducation est fixé à 100 dollars US.

Même si physiquement les IDE seront séparés de Borland dès la vente (l'éditeur ne souhaite pas qu'ils tombent aux mains de concurrents), des liens très forts demeurent entre les outils actuels de Borland et les IDE maison. Turbo est une première étape pour redorer le blason Borland !

Site : www.turboexplorer.com

■ François Tonic

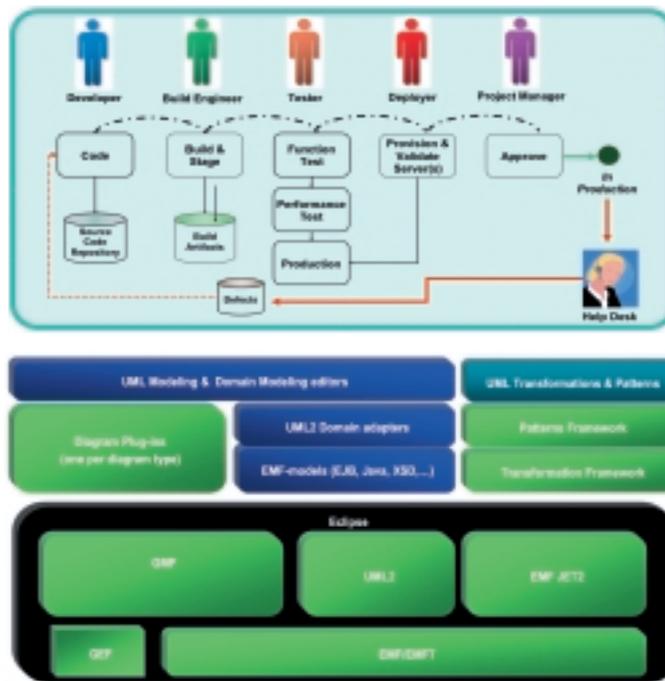
IBM complète Rational v7 avec Caspian

En juin dernier, IBM présentait la plate-forme Rational version 7, alias Baltic. La seconde étape de cette nouvelle plate-forme s'appelle Caspian. Cet ensemble complète la nouvelle offre logicielle Rational, sur la partie outils et modélisation.

Après Atlantic, IBM reste sur l'eau avec Baltic et Caspian. L'axe de Baltic est centré sur l'équipe de développement, la gestion de projet et comment accélérer et globaliser la livraison des applications. Caspian, prévu pour décembre prochain, se concentrera sur les développeurs et les tests. La problématique de Baltic est très importante car il s'agit de pouvoir lier les exigences entreprises avec l'outil informatique. Il faut donc de la réactivité, de l'efficacité et une maîtrise des coûts, tout en gérant les risques techniques et métier. La gouvernance devient donc l'élément central de Rational 7. Grâce à l'utilisation d'un processus métier unique, on gère le changement, la gestion de configuration, la qualité logicielle, le portfolio projet, les exigences, le développement, la maintenance, le déploiement, etc.

Équipe, flexibilité, gestion

Baltic propose trois grandes nouveautés : la gestion de la livraison d'applications avec Rational Build Forge qui automatise la fabrication et la livraison, une meilleure gestion des tests avec ClearQuest et une globalisation mieux prise en compte avec la notion de traçabilité introduite dans ClearQuest ainsi qu'un meilleur support des langues dans les outils. L'un des défis de la plate-forme Rational 7.0 est de simplifier l'ingénierie logicielle permettant un meilleur contrôle et une souplesse dans le modèle. Il faut repenser la modularité et la granularité du logiciel, s'inspirer du modèle communautaire de l'open source, obtenir une bonne gouvernance en centralisant, mais en laissant la place aux décisions indi-



viduelles. La notion de réutilisation de codes devient un de credo de Rational v7 tout en préconisant les méthodes agiles et la qualité du livrable. Avec Baltic, il s'agit de faire de la collaboration entre les différents rôles, cela passe par une meilleure intégration du processus métier de développement, des builds intégrés, ainsi que les tests et le déploiement. Au-delà de la version 7, IBM travaille à un cycle de vie amélioré, un environnement distribué plus performant (réseau WAN et client), une meilleure sécurité ainsi qu'à l'amélioration du coût de possession. Et cela passera notamment par l'usage de la technologie Jazz, sur laquelle nous reviendrons très rapidement.

Caspian et MDD

La prochaine étape de Rational v7 se nomme Caspian. Le modèle MDD (Model Driven Development, développement piloté par modèle)

constitue l'axe le plus important de Caspian qui se concentre sur les outils de développement. Rational n'est pas novice dans le modèle. L'éditeur avait longuement travaillé sur Rose et XDE. Avec Caspian, l'idée forte est d'apporter une flexibilité au MDD, tout en ayant une bonne intégration avec l'ensemble du processus de développement, une bonne intégration avec les différents domaines (business, technique, etc.) ainsi qu'une automatisation des tâches de développement répétitives. L'autre nouveauté se nomme True RTE. Le RTE (Round Trip Engineering) permet de construire à partir d'un code le modèle correspondant (à ce code). La difficulté est d'obtenir un modèle fidèle au code et de maintenir ensuite la synchronisation entre modèle et code. Au-delà du True RTE, Caspian apporte des nouveautés dans la modélisation : support de Java 5, l'utilisation des spéciali-

sations UML 2 (version 2.1), la disponibilité des diagrammes objets, ainsi qu'une meilleure prise en compte du MDD dans l'ensemble du processus de développement. Si la part belle est donnée à C++ et Java dans la v7, IBM annonce un MDD pour .Net après cette version. Les équipes de Rational ont aussi beaucoup travaillé sur la prise en main et l'ergonomie de l'outil. Cela passe aussi par une installation et un déploiement de la solution plus souple, plus claire. Il sera possible de créer un environnement pré-configuré (par l'administrateur), mais pas dans l'immédiat.

Au niveau architecture, Rational s'appuie largement sur Eclipse et les projets Eclipse autour de la modélisation, ce qui n'était que partiellement le cas dans la version précédente.

Rational v7 : une solution complète

Avec Baltic et Caspian, IBM Rational met à niveau la partie gestion de projet, collaboration et le développement. Si C++ et Java constituent les langages phares, .Net continue d'être une plate-forme à prendre en compte même s'il y aura un décalage de disponibilité. IBM confirme aussi la génération du "développement piloté par". Et là, on a le choix : MDA, MDD, MDE, etc. La difficulté de ces modèles est de garantir leur intégrité et la parfaite synchronisation entre le code et le modèle et, derrière, avec l'ensemble des éléments du processus de développement. Nous sommes là de plain pied dans l'industrialisation du développement logiciel.

■ François Tonic

EMBARQUÉ

Oracle met à niveau Berkeley DB

Oracle a sorti la version 4.5 du SGBD embarqué Berkeley DB. Elle fait partie intégrante de la stratégie Oracle sur le marché des SGBD embarqués. Cette version de Berkeley DB se focalise sur les performances et la disponibilité, tout en simplifiant le modèle de développement. La gestion de l'accès concurrent a été améliorée, tout comme la réplication avec un framework de réplication. Ce SGBD est disponible en deux licences : l'une open source, l'autre commerciale. Oracle a par ailleurs annoncé le rachat du Français Sunopsis, un éditeur spécialisé dans l'intégration des données. Ce rachat doit renforcer l'offre Oracle Fusion Middleware, notamment sur l'intégration des données hétérogènes.

Site : <http://www.oracle.com/technology/products/berkeley-db>

OUTILS

Toad pour DB2 version 2

L'éditeur Quest Software lance Toad pour DB2, sur systèmes Linux, Windows, Unix et z/OS (nouveau venu). Toutes les plates-formes supportées pour DB2 le sont par Toad, simplifiant le travail du DBA. La version 2 ajoute un module d'analyse des données, un filtre objet pour aider à la navigation dans les bases, une personnalisation de l'interface utilisateur par l'administrateur et l'intégration dans Toad de DB2 Personal Connect. L'outil est disponible dès maintenant (à partir de 727 euros par utilisateur).

Quelques questions à David Axmard, co-fondateur et vice-président de MySQL

Programmez! : Pensez-vous que le marché des bases de données a changé depuis l'apparition des SGBD open source et notamment de MySQL ?

David Axmard : Certainement ! Les SGBD open source sont pleinement reconnus sur de nombreux marchés. Switcher sur MySQL peut réduire les coûts de 90 % ! Notre SGBD (gratuit) est téléchargé plus de 100 millions de fois et la version 5, plus de 5 millions !

PI : Pour vous, quelle est la philosophie de MySQL ?

D.A. : Offrir un outil alliant robustesse, rapidité et simplicité d'utilisation.

PI : Depuis les débuts de MySQL, le modèle de développement a-t-il évolué ?

D. A. : Eh bien, notre équipe a considérablement grossi. Il y a une grande différence entre avoir un noyau de 2 personnes et 50 personnes.

PI : Oracle et Microsoft se livrent une course à la fonctionnalité. Qu'en est-il de MySQL ?

D. A. : Au risque de me répéter, je redirais ceci : robustesse, rapidité, simplicité d'utilisation. Cela a toujours été nos objectifs. Il est important d'implémenter des fonctions dont les utilisateurs actuels vont se servir, au lieu de juste ajouter de plus en plus de fonctions. Bien entendu, nous innovons aussi. Regardez, par exemple, notre concept de moteur de stockage qui est interchangeable.

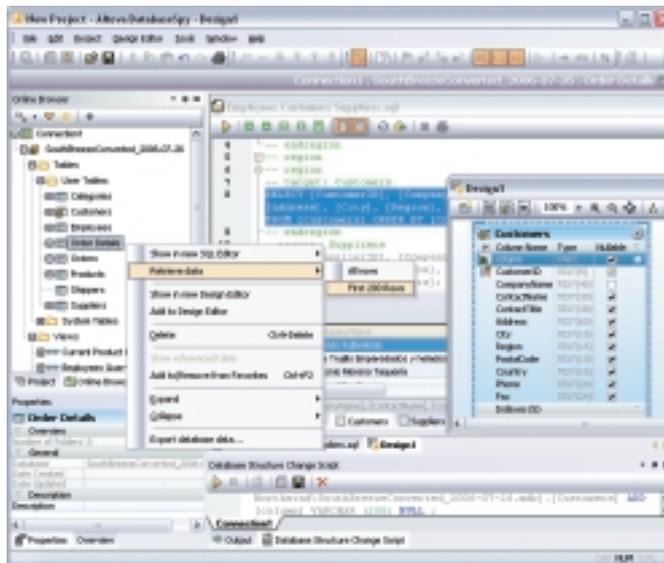
PI : Pour vous, quelles seront les évolutions des prochaines années ?

D. A. : Nous travaillons à rendre MySQL toujours plus modulaire ("plug-gable") permettant à des projets extérieurs de s'intégrer à MySQL avec un minimum de travail. Avec les plug-ins, il est possible d'améliorer l'innovation autour de MySQL.



ADMINISTRATION

Altova aide les données



L'éditeur Altova vient d'annoncer la sortie de son outil DatabaseSpy 2007. Il permet de gérer les données, de prendre en charge les requêtes et de créer des connexions avec les principales bases de données. La philosophie de DatabaseSpy est de simplifier et d'aider le DBA et le développeur notamment sur la création de requêtes SQL, le design d'une base et la gestion d'une base de données (SQL Server, Oracle, DB2, MySQL, Sybase). Sur la connectivité, il supporte ADO et ODBC. Sur cette partie, on dispose d'un assistant de création de connexion. XML n'est pas oublié. L'outil peut vous aider à transformer les données d'un SGBD vers une application XML. Il est possible d'utiliser l'outil maison XMLSpy. Il est disponible dès maintenant (à partir de 129 dollars US). Une version limitée est également disponible.

Site : www.altova.com

CRITIQUE

Sybase renforce les risques

L'éditeur a dévoilé début octobre la version 3 de Risk Analytics Platform. Il s'agit d'une plate-forme orientée marché financiers pour déployer des applications critiques d'analyse en temps réel. Cette version bénéficie d'une nouvelle console de gestion des données workflow, de synchronisations et de monitoring. L'éditeur a ajouté de nouveaux adaptateurs de données pour étendre la prise en charge des données de diverses sources. L'outil repose, pour les fondations, sur Adaptive Server Enterprise 15 et Sybase IQ. Pour la modélisation, on bénéficie de PowerDesigner. À noter aussi l'amélioration des fonctions OLAP.

D'autre part, Sybase a aussi complété Anywhere Suite avec Unwired Accelerator 8. Cette solution aide à déployer des applications mobiles pour une utilisation locale ou connectée. L'outil prend en charge les dernières versions de Business Objects, Crystal Reports, Lotus Notes, SAP... Pour ces outils mobiles, l'éditeur utilise maintenant Eclipse, et le support de SOA fait son apparition.

Site : www.sybase.com

VIRTUALISATION

OpenVZ en nouvelle version

En pleine ébullition, le marché des outils de virtualisation accueille une nouvelle édition du projet OpenVZ. Cet outil fonctionne uniquement sous Linux (Red Hat, Debian, Mandriva). Soutenu par SWSOft, OpenVZ permet maintenant d'indiquer le nombre de processeurs, de la prise en charge de l'ethernet virtuel. " Nous continuons à sortir nos logiciels à un rythme élevé, pour mettre à la disposition des utilisateurs une technologie sans cesse meilleure et toujours plus polyvalente. ", déclare Kir Kolyskin, responsable du projet OpenVZ. Site : www.openvz.org

OUTIL

X-develop 2.0 disponible

L'éditeur multi-langage X-develop est disponible depuis peu en version 2.0. Il prend en compte ASP.net, C#, HTML, Java, VB, XML, supporte Mono et fonctionne sous Windows, MacOS X et Linux. Parmi les nouveautés de cette version, on notera des améliorations sur les supports des différents langages : par exemple de Cassini et Mono XPS sur la partie serveur web (ASP.Net), une édition du code Ajax avec vérification syntaxique, complétion, indentation, etc. Site : www.omnicore.com

INSOLITE

Un peu plus de MacOS X sous Windows Vista

Un développeur .Net, Simon on the (dot)Net, a décidé de créer un clone de la fonction Exposé de MacOS X sous Windows Vista. Et la mission a été remplie ! Il nécessite Windows Vista RC1 et le mode Aero Glass. À voir et à télécharger !

Site : <http://blogs.labo-dotnet.com/simon/archive/2006/09/12/11116.aspx>

IDE

RainingData propose Omnis Studio 4.2



L'atelier logiciel Omnis Studio 4.2 est téléchargeable depuis quelques semaines. Cette version fonctionne maintenant sur Mac Intel en mode universel. Le Web Service, via un plug-in, est supporté en serveur et en client et pour le client léger, on dispose

d'un serveur http. Firefox est maintenant supporté par le client web Omnis. Dans le développement, il y a des améliorations diverses dans le langage et l'éditeur (templates html, apparition des sous-types dans une liste ou une variable rangée, de nouvelles préférences de la JVM...). Et si vous devez utiliser des objets .Net, Omnis Studio vous propose des astuces pour réaliser cette opération. Pour mieux faire connaître Omnis Studio et le montrer en action, des webinaires sont disponibles, sur différents thèmes. L'outil fonctionne sur Windows, Linux et MacOS X. Site : <http://www.omnis.net>

Perforce

Gestion rapide de configuration logicielle (GCL)



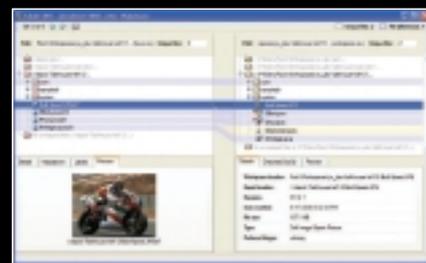
La fonctionnalité Folder Diff, un atout de productivité du système de GCL Perforce.

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

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

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

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



Folder Diff de Perforce

PERFORCE

S O F T W A R E

Téléchargez sans conditions une copie gratuite de Perforce sur www.perforce.com. Vous bénéficiez d'une assistance technique gratuite pendant votre période d'évaluation.

SGBDO, SGBDRO, SGBDR : l'objet au cœur de la donnée ?

Voilà une question qu'il faut pouvoir comprendre et éventuellement trancher. On parle bien de deux types de base de données. Peut-on considérer un SGBDR orienté objet ? La tendance actuelle est bel et bien de mettre des notions objets dans les bases relationnelles. L'arrivée du XML natif est-il un élément supplémentaire poussant vers l'objet ?

Basiquement, un SGBDO permet de manipuler des collections objets directement dans la base et les tables. On évite ainsi d'utiliser des outils de mapping relationnel objet qui s'occupent de faire le lien entre la donnée relationnelle et l'objet. Dans un SGBDRO, on rajoute des notions objets à la base relationnelle, avec stockage ou non des objets dans des tables.

Passer par des couches indépendantes ou dans le SGBDR ?

Un SGBDRO intègre directement dans son environnement une gestion de l'objet. Mais si l'objet est directement intégré, la base de données demeure un classique

SGBDR au niveau des données, contrairement au SGBDO qui est réellement objet de bout en bout. L'avantage d'un SGBDO est que l'on travaille en objet du début à la fin, sans nécessité du moindre mapping. Dans le cas d'un SGBDRO, nous sommes dans un modèle hybride, car les données demeurent relationnelles. Le mapping se fait dans la base et non à l'extérieur. Il ne faut pas oublier que l'objet et le relationnel sont deux mondes différents. Quand on utilise un SGBDR et que l'on développe en objet, il faut créer une passerelle entre les deux, d'où les outils de mapping relationnel / objet.

Utiliser un mapping externe à la base, bien que cela rajoute une couche technique, offre une souplesse d'utilisation appréciable, dans le sens, ou vous pouvez changer de mapping (quitte à re-développer le code), et avoir une souplesse d'adaptation selon les besoins et les demandes. Cette démarche s'avère particulièrement adaptée pour les SGBDR en place et ayant un existant fort. Car passer d'un modèle SGBDR à un modèle SGBDRO nécessite des développements, une migration. Alors que le couple SGBDR + mapping se veut moins intrusif, plus souple.

" Il existe une course à la fonctionnalité. SQL Server est plus un "suiveur" qu'autre chose. En complexité, on atteint les limites. Chez les clients il y a un besoin de simplification, de performances, de stabilité. Il faut que l'outil de développeur s'adapte bien à la base de données en dessous. Il n'y a pas la même culture entre le développeur et l'exploitation. Les clients s'y perdent un peu. Dans 80 % des cas, ils ont des besoins "simples". En face, on a des réponses compliquées", précise **Serge Frezefond**, consultant **MySQL**.

SGBDR, SGBDRO, SGBDO ou autre chose ?

Utiliser un SGBDRO peut avoir un avantage, surtout si on possède une ancienne édition uniquement relationnelle. Cela évite de tout porter, de tout recoder. Mais on est lié à une plate-forme. L'option, comme vue plus haut, d'un SGBDR couplé à un middleware de mapping possède des avantages : indépendance relative, codage limité même si le code généré nécessitera vérification et optimisation. Vous pouvez aussi opter pour une base de données de type **InterSystems Caché**, une base dite post relationnelle. Dans ce cas, on manipule des objets directement dans la base. On modélise des classes qui sont ensuite générées en classe (objet) et en table, le tout pour Java, EJB ou .Net.

SGBDR, SGBDRO, SGBDO ou autre chose ?

" Les utilisateurs viennent vers nous pour la facilité de développement, les performances, quand les SGBDR sont trop consommatrices de ressources. La base de données doit répondre aux exigences de l'utilisateur. De plus, faire du mapping est synonyme de consommation de temps de développement même si le code

est (en partie) généré, il apporte une certaine complexité. Il faut surveiller le code, l'optimiser, lorsque le modèle évolue, il faut faire évoluer le mapping ", commente **Patrick Oliviero** d'InterSystems. Effectivement, il est à noter que souvent, la couche objet d'un SGBDR est avant tout une sur-couche.

SGBD ou plate-forme de données ?

Voilà une question importante, la base de données serait-elle autre chose qu'une simple base de données ? C'est ce que Microsoft souhaite faire passer comme message avec **SQL Server 2005**. " Notre vision est de pouvoir disposer d'un repository afin de gérer le cycle de vie de la donnée. Il y a une forte intégration entre SQL Server et .Net. SQL Server n'est plus une base de données mais une plate-forme, capable de gérer les données d'une entreprise. Il s'agit d'apporter une flexibilité, une souplesse. On fait aussi de l'interopérabilité en environnement hétérogène, le support XML démontre notre volonté en ce domaine. Le XML natif est pertinent ", commente **Angelica Reyes-Matiz**, responsable de la plate-forme applicative de **Microsoft France**.



Avec l'accroissement du volume des données, les applications de plus en plus orientées données, le SGBD se place au cœur de l'environnement applicatif. Mais cela oblige à avoir une base de données la plus transparente possible pour le développeur et l'utilisateur. " La base devient réellement un service. Il faut l'oublier, elle doit être transparente "; conclut Patrick Oliviero.

■ **François Tonic**



Une application multilingue
avec plus de 100 vues de 20 types différents,
en DHTML/Ajax, en Swing ou en plugin Eclipse,
connectée à un SGBD et un bus JMS.
Il vous faut combien de temps pour la réaliser ?



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

L'IHM en toute simplicité.

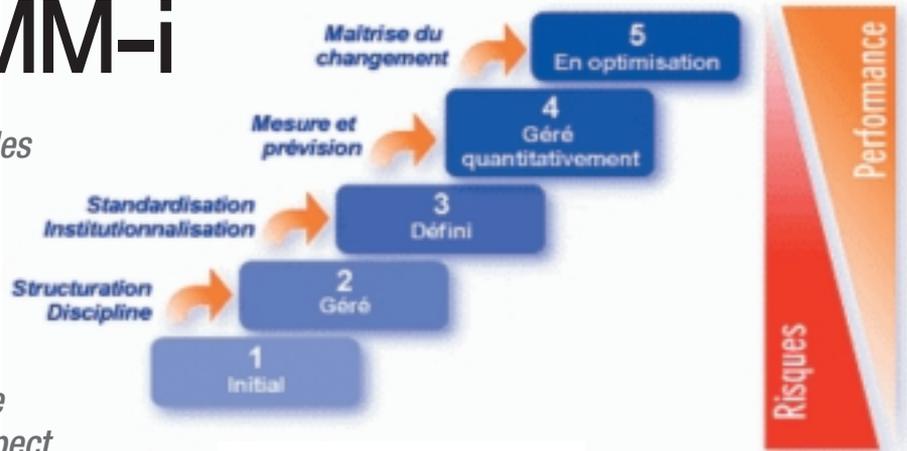
www.lyria.com



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.

Le modèle CMM-i

Aujourd'hui, la tendance au sein des organisations qui réalisent des développements logiciels est la mise en place de démarches d'amélioration de leur activité. L'objectif annoncé par les directions se traduit en terme de rentabilité, de productivité, de respect des engagements... Et les équipes opérationnelles sur le terrain ont parfois des difficultés à percevoir l'incidence de cette amélioration sur leur travail au quotidien.



© GROUPE SQLI 2003
CMM-i : les niveaux de maturité

Le modèle CMM-I est l'un de ces modèles d'amélioration, nous allons l'utiliser pour illustrer cet impact. Pour rappel, CMM-I (Capability Maturity Model Integrated) est un modèle qui procède d'une logique d'amélioration progressive sur cinq niveaux de maturité, qui permet d'évaluer la maturité des processus et des pratiques d'une organisation par rapport à ce qu'il faudrait faire idéalement dans les projets, afin d'avoir un fonctionnement le plus prédictif possible. Chaque niveau du modèle couvre différents domaines d'activité (en tout, il y en a 25) qui sont utilisés dans la réalisation ou le support à la réalisation de produits logiciels.

Les premiers niveaux

Le niveau 1 correspond à une maturité initiale dans laquelle se trouvent toutes les équipes qui arrivent à délivrer un produit logiciel. Et cela, quels qu'en soient les retards, les écarts de fonctionnalités... constatés dans le produit final. Bonne nouvelle, vous êtes probablement tous de maturité 1. Dans ce contexte, vous avez des projets qui se déroulent de manière différente selon l'expérience des intervenants (chef de projet, expert, développeur). Chaque projet possède ses propres outils de travail (les environnements de modélisation, de développement, sont différents ou paramétrés différemment sur chaque projet), le suivi de l'avancement est hétérogène (l'un suit le planning, l'autre le temps passé, un troisième uniquement le nombre d'anomalies après chaque livraison), le formalisme des livrables produit par le projet est spécifique à chaque

contexte (l'un fait des maquettes, l'autre des spécifications détaillées qui incluent les modèles de données, d'autres des modèles objets...).

Dans ces conditions, le développeur change régulièrement ses habitudes de travail à chaque changement de projet.

Le niveau 2 apporte une discipline dans le projet. Pour cela, il est demandé de mettre sous surveillance un certain nombre d'activités pour s'assurer que le déroulement du projet est conforme au planning initialement défini. Les conséquences de ce niveau semblent faibles au quotidien pour les développeurs. En effet, les activités directement concernées touchent le domaine de la gestion de projet. Les chefs de projets sont donc fortement mis à contribution pour maîtriser le périmètre de leur projet (domaine de Gestion des exigences – REQM), déterminer un plan pour l'atteinte de ce périmètre (domaine de Planification de projets – PP) et suivre l'avancement du projet par rapport à ce plan (domaine de Surveillance des projets - PMC) et traiter les écarts constatés (dérive de charge, dérive de délai, évolution du périmètre). La direction informatique donne en général des directives sur les activités d'ingénierie (spécification, conception, réalisation, test) à planifier et à suivre sur les projets. Il reste toutefois quelques écarts sur la manière de réaliser ces activités. Les outils, les modèles de document, la règle de constitution des données de suivi restent parfois hétérogènes. Cet effort sur la discipline se traduit pour les développeurs par une meilleure visibilité sur l'état actuel et à venir

du projet, à l'ensemble de l'équipe : les développeurs connaissent leurs tâches sur les 2 ou 3 prochaines semaines, la date de livraison finale du produit, l'objectif du projet pour les utilisateurs finaux.

Deux domaines d'activité viennent compléter le niveau 2. L'une, l'assurance qualité (domaine PPQA – Process and Product Quality assurance) qui vient vérifier régulièrement au sein des projets que les directives de l'organisation sont bien appliquées. L'autre, la gestion de configuration (domaine CM – Configuration Management), plus proche des développeurs, qui se traduit par la mise en place d'une arborescence projet standardisée sur tous les projets, la mise en place d'un outil de gestion des versions (des codes sources en particulier ou des documents du projet), que ce soit Subversion, CVS, Visual Source Safe, PVCS, StarTeam...), un outil de gestion des changements, qui dans un premier temps est simplement un outil de gestion des anomalies. L'ensemble de ces outils s'accompagne de procédures d'utilisation qui décrivent en particulier les responsabilités des intervenants du projet : " Qui décide du moment et par qui une anomalie sera corrigée ".

Le niveau 3 renforce l'homogénéité du suivi, mais également sur la manière de faire les choses. C'est donc à ce stade que l'ensemble des activités d'ingénierie commence à être homogénéisé (domaine OPD – Organizational Process Definition). Des modèles de documents, des guides de rédaction (guide de conception, guide de développement, ...), des guides d'utilisation des environnements de dé-

veloppement, des guides et des outils pour supporter les tests sont mis en place. La "Software Factory" de l'organisation se met en place à ce moment (domaine TS – Technical Solution). L'élaboration de la stratégie de test suit une démarche structurée (domaine PI – Product Integration, VER – Verification, VAL – Validation). Les formations pour acquérir les compétences nécessaires à l'utilisation de tout ou partie de cette usine logicielle sont planifiées et dispensées auprès des équipes (domaine OT – Organizational Training).

Un vocabulaire commun est utilisé au travers de l'ensemble des projets : les termes d' "Analyse", de "Spécification", de "Conception", de "Test unitaire", de "Test d'intégration", ... ont le même sens pour tous. Chacun sait ce qu'ils représentent et comment cela se traduit concrètement dans ses activités de tous les jours. Par exemple :

- Le résultat de la conception est un modèle physique de données ou un modèle de classe,
- Le test unitaire est l'utilisation d'une liste standardisée de points à vérifier (et uniquement ceux là) par le développeur à la fin de ses tâches de développements.

Cette uniformisation permet de collecter des retours d'expérience lors de bilans auxquels les différents intervenants sont conviés, afin d'identifier les bonnes et mauvaises choses qui ce sont déroulées durant le projet.

Les informations capitalisées sont principale-

ment qualitatives, mais s'appuient sur des données quantitatives (nombre d'anomalies, écart entre la charge prévue et le réalisé...). Elles concernent à la fois les processus de l'organisation, mais également les produits réalisés. Cela se traduit donc par de la capitalisation technique et de la réutilisation de composants, d'outils ou de solutions. L'efficacité de cette collecte est d'autant plus importante, que les bilans sont réalisés sur des contextes similaires (même environnement de développement, même technique de test,...). Les retours exploitables et comparables sont de plus en plus nombreux au fil des projets et permettent à chacun, y compris les développeurs, d'améliorer leur travail.

Une organisation plus efficace

Les niveaux 4 et 5 se focalisent principalement sur la mise en place d'une organisation efficace pour l'exploitation des enseignements et l'accélération dans l'optimisation des processus et des outils mis en place dans les précédents niveaux.

L'effet constaté par les équipes de développement est une prise en compte, au sein des processus standard, des retours d'expérience et une recherche systématique des axes d'optimisation. Ces derniers peuvent également passer par l'innovation, voire la rupture technologique, dans un contexte maîtrisé. Les innovations répondent à des besoins réels de changement dont le retour sur investissement

est totalement connu. Cela permet de ne faire évoluer les processus que dans le cas d'un gain réel.

Ces deux derniers niveaux s'appuient beaucoup sur des analyses chiffrées pour déterminer quelles innovations étudier, et pour décider ou non de leur déploiement. Cela semble à nouveau éloigné des équipes de développement alors qu'en réalité, dans une organisation à ce niveau de maturité, tous les collaborateurs de l'organisation (développeur, chef de projet, concepteur, manager, ...) ont conscience des enjeux de l'amélioration permanente pour la stratégie de l'entreprise et y contribuent pleinement. Ces niveaux ne sont donc pas une fin en soi, mais le début de la concrétisation de la recherche d'optimisation sur sa pratique de réalisation des logiciels.

Dans cet exposé, il apparaît que la mise en place de CMM-I procède d'une démarche de recherche d'amélioration permanente au sein d'une organisation. Les exemples cités plus haut sont là pour donner des idées de ce qu'il est possible de faire, toutefois chaque organisation doit choisir la déclinaison des objectifs des différents domaines de CMM-I pour ses besoins propres. En clair, CMM-I n'est pas une méthode, mais un cadre pour la définition et la mise en place d'une méthode au sein d'une organisation.

■ Christian Hartz

Directeur de projet, en charge du programme CMM-I Groupe SQLi

EN BREF... EN BREF... EN BREF... EN BREF... EN BREF...

Borland renforce son cycle de vie

Borland lance sa solution de qualité tout au long du cycle de vie, Borland LQM. L'objectif est de qualifier le projet dès la définition des besoins. Cela peut permettre d'éviter retards et problèmes de conception. Cette solution fait partie d'un ensemble de 4 solutions pour favoriser l'industrialisation du logiciel. On y retrouve : Gauntlet (racheté par Borland au printemps dernier) pour les tests et détection de défauts, SilkCentral Test Manager pour l'administration de la qualité logicielle, une meilleure intégration avec CaliberRM et Caliber DefineIT ainsi qu'avec StarTeam. " Les conséquences d'une médiocre qualité des logiciels sont bien connues... Pourtant, il s'agit souvent d'une préoccupation traitée a posteriori, en aval du cycle de développement, et dont seul le département d'assurance qualité – lorsqu'il existe - est tenu pour responsable " commente Erik Frieberg, Vice-Président du marketing et de la stratégie produit de Borland Software. " Borland permet aujourd'hui aux entreprises de faire de la qualité une préoccupation partagée par toutes les équipes et tout au long du cycle de vie. Il s'agit d'une étape significative de notre vision stratégique de l'optimisation des livraisons logicielles au sein de laquelle la gestion du cycle de vie qualité joue un rôle critique". Site : www.borland.com

Nuxeo passe à Java

L'éditeur Nuxeo a annoncé il y a quelques semaines sa migration de Zope (en Python) à Java pour son outil de gestion de contenus, Nuxeo CPS. Rebaptisée Nuxeo 5, cette nouvelle version s'appuiera sur Java EE 5. Nuxeo explique cette transition par les nouveaux aspects open source de Java. Il s'agit surtout de pouvoir s'appuyer sur une forte communauté Java et la simplification de Java EE 5.

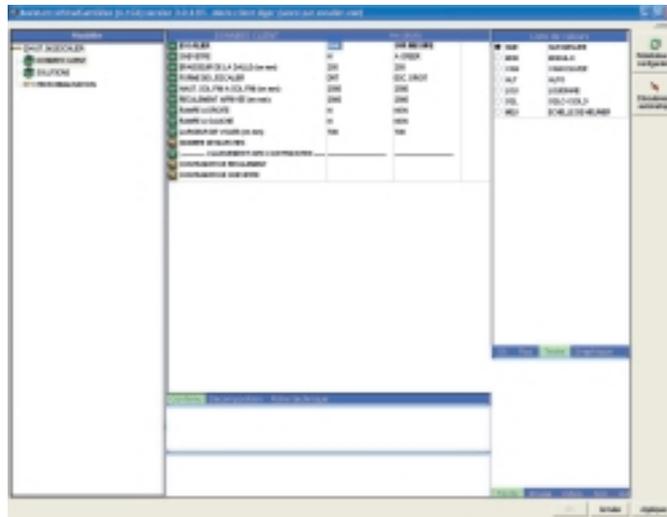
Du point de vue marché, Java est bien perçu par les entreprises alors que Zope, malgré une percée sensible, demeure peu connu. Et le passage à Zope 3 posait des soucis de développement. Dans le mouvement de Nuxeo 5, l'éditeur a dévoilé le projet Nuxeo RCP. Il repose sur le projet Apogée qui fournit des composants et surtout une infrastructure logicielle s'appuyant sur Eclipse.

D'ailleurs, Nuxeo RCP repose sur Eclipse RCP. Nuxeo ajoute ces composants et surtout Nuxeo Core Engine. Nuxeo RCP s'intègre à Nuxeo 5. Le projet inclut aussi Apache Jackrabbit. Nuxeo Core inclut les services de bases de gestion de contenus (les API, le moteur de requêtes, les schémas, le repository...).

Site : nuxeo.org

Caméléon : Red Hat déploie Linux chez Lapeyre

Red Hat a permis de faire évoluer le logiciel de ventes et de commandes Caméléon de Lapeyre. Déployé sous Linux dans un environnement Oracle 9i, il est aujourd'hui présent dans l'ensemble des magasins du groupe.



L'application Caméléon

Le logiciel Caméléon est indispensable aux vendeurs Lapeyre : il leur permet de personnaliser chaque commande en introduisant les variables que le client va adapter en fonction de ses besoins. Fenêtres, portes, escaliers ou volets sont passés au crible par le client et son conseiller, et, selon les mesures indiquées, des simulations sont possibles. Ainsi, le catalogue des différents produits est consultable; des schémas en 3D et des animations permettent de visualiser et de comparer les choix.

Au cœur du système de vente, Caméléon permet donc de dérouler des scénarios pour une solution sur-mesure, de les virtualiser sur écran, et d'en étudier la faisabilité technique. Autre avantage : la commande ferme est chiffrée en direct et envoyée aux usines chargées de fabriquer les matériaux.

Cette expertise sur la personnalisation de produits adaptés à chaque client a permis à Lapeyre de se forger sa réputation. Au moment où la décision de faire évoluer le logiciel a été prise, plusieurs pistes s'ouvraient, sachant que Caméléon devait rester opérationnel et accroître ses fonctions pour demeurer un adjuvant commercial indispensable. D'un côté, les solutions propriétaires, étant donné que Lapeyre est une filiale de Saint-Gobain, fortement lié à Microsoft. De l'autre, la volonté de jouer la carte du moyen et du long terme en se penchant sur Linux.

" Une consigne : ne pas perturber le système de vente... "

Philippe Rennes, responsable système et bases de données chez Lapeyre, n'a pas fait d'appel d'offres : le choix de Red Hat, leader



Philippe Rennes, responsable système et bases de données chez Lapeyre

sur le marché Linux, s'est naturellement imposé, en amont. " Nous travaillions sporadiquement avec eux et connaissions leurs compétences. De plus, nos ingénieurs avaient une forte connaissance de Unix ; aussi, Linux nous a semblé la solution idéale. En effet, nous souhaitons nous affranchir du mode propriétaire sans perdre les avantages d'un support régulier et pérenne. Il importait également que nous puissions disposer d'un matériel certifié, performant et stable. Enfin, il ne fallait pas perturber le système de vente " indique-t-il.

Du côté de Red Hat, on s'apprête à relever le défi : une personne a travaillé pendant un mois sur le projet. " La première étape a été la construction d'un master avec un consultant, à partir des pré-requis et des configurations du système Caméléon qui existait déjà, souligne Franz Meyer, directeur de Red Hat pour l'Europe du sud, le Moyen-Orient et l'Afrique. Cette étape était essentielle car il n'était pas question d'envisager de tout refondre et de former les conseillers de clientèle en magasin. Ceux-ci devaient pouvoir continuer à travailler. "

L'équation gagnante : Red Hat Network Satellite et Oracle 9i

L'enjeu est double : faire évoluer Caméléon avec déploiement massif sur les serveurs des 150 magasins, sans doubler la facture par les

interventions physiques des techniciens de Red Hat sur les sites Lapeyre. La solution retenue est donc la migration vers une base de données Oracle 9i, l'ensemble étant supporté par l'ensemble Red Hat Enterprise Linux. Lapeyre se dote également du système Red Hat Network Satellite, qui lui permet de télécharger et d'installer automatiquement les mises à jours du système d'exploitation, les patches et, pour gérer l'ensemble de l'infrastructure, Linux. Enfin, le support Red Hat est également ajouté pour pallier les éventuels problèmes techniques.



Franz Meyer, directeur SEMEA de Red Hat

Concrètement, l'ensemble du déploiement prend moins d'un semestre : " dans chaque magasin, un serveur spécialement conçu pour répondre aux besoins de Lapeyre a été installé, raconte Philippe Rennes. Il a fallu en moyenne un quart d'heure aux techniciens de Lapeyre pour connecter le serveur, brancher les appareils et déployer le master. " Chaque magasin est équipé d'un Dell PowerEdge 1850 ou 2850 avec une base Oracle 9i.

Les données transitent de Caméléon en mode client-serveur jusqu'aux usines, en toute transparence.

Résultat : Caméléon est plus que jamais au centre du réseau Lapeyre, les nouveaux magasins en sont équipés avec les versions évoluant au fur et à mesure et les relations entre Lapeyre et Red-Hat sont au beau fixe !

■ Claire Padych

Avec AMEN, détendez-vous.
Nous avons créé la solution
d'hébergement 100% pro, 100% zen.

NOUVEAU



**SATISFAIT
OU
REMBOURSÉ !**

Un espace d'hébergement accru (5 Go), nouveau datacenter 100% sécurisé,
infrastructure en cluster avec sauvegarde,
haute disponibilité (99,9%), assistance technique 6j/7⁽¹⁾.

016 AMEN RCS PARIS, B. 421 527 787, IN.WEB WE TRUST - Nous croyons au web. * Voir conditions sur le site www.amen.fr. Conditions Générales de Vente sur www.amen.fr. Prix au 01/01/2006, modifiables sans préavis. * Sous 10 jours, hors achat du nom de domaine. (1) par email.



- nom de domaine : .fr, .eu, .com, .net, .org ...
- hébergement web dynamique de 5 Go,
- 10 bases MySQL 4, PHP 5, Perl 5, Python, Ruby,
- messagerie de 1000 comptes POP/IMAP avec 25 Go de stockage,
- anti-virus/anti-spam,
- + 5 applications pré-installées offertes au choix : osCommerce, forum, FAQ, blog, album photo, gestion de contenu + l'outil de création de site en ligne "Website Creator".



**70,80 €
HT/AN**

(Soit 84,68 € TTC/AN)

www.amen.fr

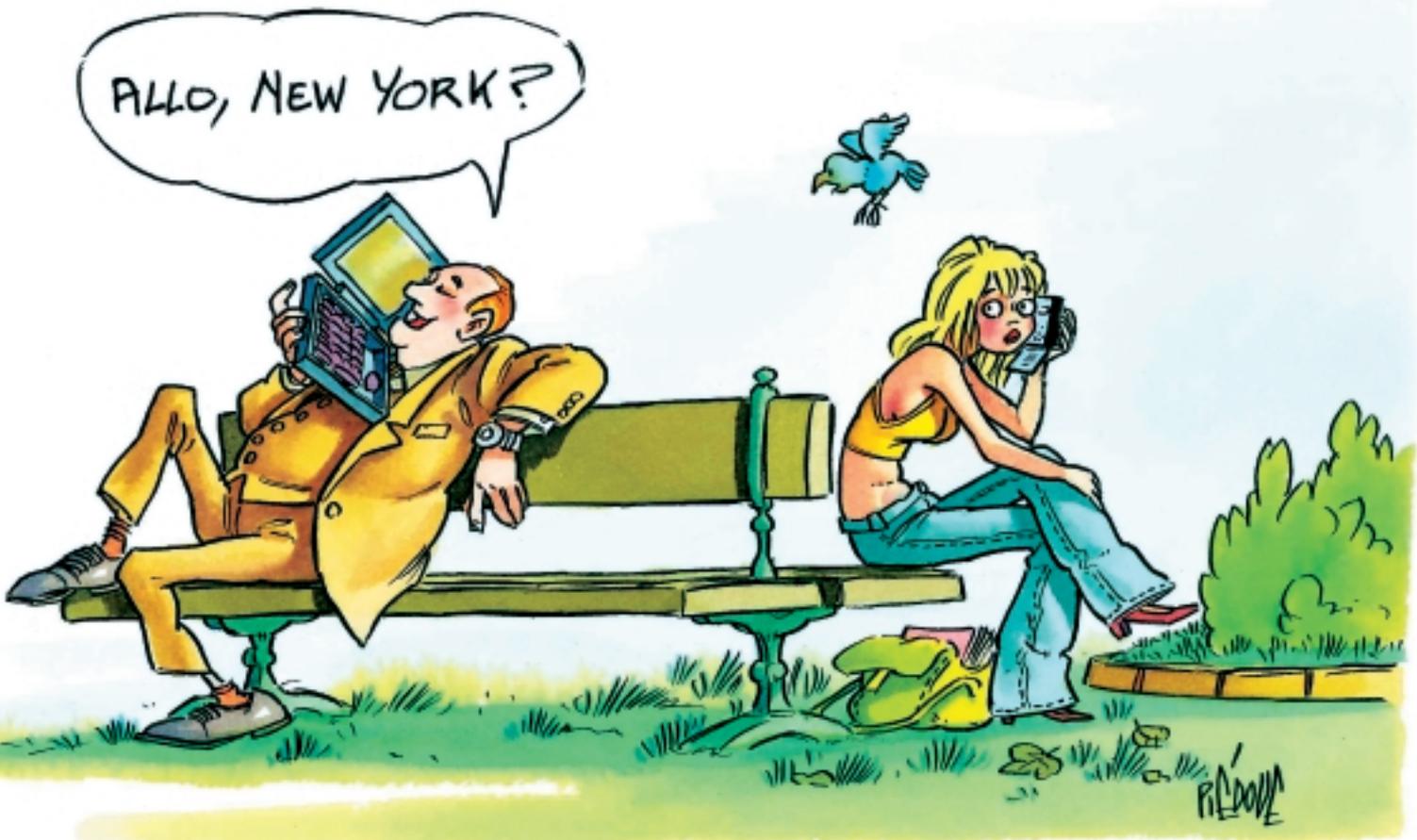
0892 55 66 77

[0,34 € TTC/mn depuis la France 9H - 19H]


amen
IN WEB WE TRUST



VoIP Mettez de la voix dans vos applications



La téléphonie a connu plusieurs révolutions : les téléphones portables, internet, l'ADSL, la fibre optique, le satellite. Si Skype a bouleversé le marché en 2005, une révolution bien plus profonde se met en place : la voix sur IP (ou VoIP). Et il n'est pas rare de voir des téléphones portables l'utilisant. En entreprise, le réseau VoIP n'est plus une bête curieuse mais un véritable outil au quotidien. À la maison aussi, cette technologie remplace parfois l'opérateur télécom traditionnel.

Les offres estampillées VoIP se multiplient dans le matériel mais aussi dans le logiciel. Skype a été l'un des fers de lance de cette nouvelle manière de communiquer. Que ce soit dans l'open source ou le commercial, les applications se multiplient et intéressent les grands éditeurs : Microsoft avec Communication Server, Live Messenger, Windows Mobile 5, Yahoo, Google, etc. Pour le développeur aussi, VoIP étoffe sa panoplie de bibliothèques et API pour implémenter et coder sa voix sur IP. Depuis le printemps dernier, le consortium Eclipse travaille sur ce domaine avec ECF Call API.

Asterisk PHP permet d'écrire des applications Asterisks en PHP. En Python, on utilisera le framework Shtoom ou encore JMF si on est en Java, etc.

Cependant, VoIP apporte aussi son lot de problèmes, d'incompatibilité, de choix techniques et technologiques. Il existe de nombreux standards pour établir la communication (ex. : SIP et IAX). Que l'on soit dans un environnement ouvert ou fermé, les choix ne seront pas forcément identiques. Ensuite, il y a la notion de qualité de services de la VoIP. Et même si globalement la qualité s'améliore grandement, il demeure des facteurs de dégradation de services qu'il faut identifier et résoudre.

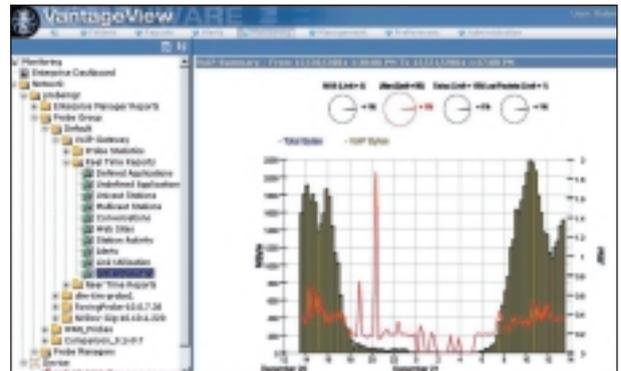
Dans ce présent dossier, nous avons abordé deux thèmes : le marché et les problèmes de qualité et l'implémentation. Pour la partie codage, nous vous montrerons comment créer simplement et rapidement un client IAX en C avec la bibliothèque IAXClient ainsi qu'une implémentation Java de VoIP avec Java Media Framework. Bonne écoute !

■ François Tonic



Voix sur IP : cap sur la qualité !

La voix sur réseau IP (VoIP ou parfois nommé ToIP, T pour Téléphonie) consiste à faire transiter une communication sur un réseau IP. La voix est numérisée et compressée afin d'être transmise par paquets IP. Le destinataire décompresse et restitue la voix. VoIP repose sur des protocoles réseau et de communication. Mais si VoIP apporte une réelle souplesse d'utilisation et une baisse du coût de téléphonie, la question de la qualité devient primordiale.



La voix sur IP, comme la téléphonie traditionnelle, constitue un service vital pour l'entreprise, en quelque sorte, une application critique. Il faut se montrer prudent en entreprise et qualifier son réseau. " On peut patienter quelques secondes devant un écran, en téléphonie, non. Les utilisateurs ne comprennent pas la dégradation de services et de qualité. Il peut alors y avoir rejet si cela ne fonctionne pas ! " précise Gilles Portier (responsable Avant-vente Performance chez Compuware). " Il peut y avoir de nombreux problèmes avec les applications passant par le réseau. Le problème ne se règle pas en mettant plus de bande passante ! On peut par exemple perdre des paquets TCP, avoir un mauvais temps de transit. Il faut, tout d'abord, mesurer si un déploiement se fera convenablement sur le réseau et, en cas de problème, découvrir où est l'obstacle ", poursuit Gilles Portier.

Il ne faut pas oublier que VoIP passe par un réseau IP. Si d'autres applications échangent sur ce réseau, il peut y avoir des problèmes de bande passante. Le rajout de bande ne résout pas le problème de trafic.

Et la sécurité ?

À l'instar des réseaux wifi, un réseau de téléphonie IP n'est pas à l'abri de piratage ou de capture des conversations. Une des attaques possibles est de convertir les flux de voix en fichier audio audible pour un simple lecteur. Des outils comme vomit (voice over misconfigured internet telephones) ou SiVus, et tout autre scanner VoIP, permettent ces attaques. Bref, la VoIP est une nouvelle source d'insécurité ou d'espionnage. Une des recommandations de base est d'utiliser des protocoles sécurisés

(ex. : IPSec). Cependant, il faut que les API utilisées, les softphones et le matériel VoIP soient compatibles avec ces mécanismes de sécurité. Soyez donc très vigilant !

Quelques lectures indispensables sur le sujet : http://actes.sstic.org/SSTIC05/ILTY__Im_Listening_to_you_via_VOIP/SSTIC05-article-Bareil-VOIP_Projet_ILTY.pdf#search=%22VOMIT%20voip%22

- <http://www.hsc.fr/ressources/presentations/rs06-voip-sec/rs06-voip-sec.pdf#search=%22VOMIT%20voip%22>

- <https://www.securite.org/presentations/voip/>

Quelques API pour implémenter la VoIP

Le développeur qui voudrait créer son softphone ou simplement intégrer des fonctions VoIP pourra utiliser le protocole SIP, aujourd'hui très répandu. Les API SIP sont relativement nombreuses et présentes sur la majorité des plates-formes.

Sous .Net

SIP .Net 1.0 d'Independentsoft (<http://www.independentsoft.com/>) est l'une des API les plus connues sous Windows. Elle fonctionne aussi bien sur le framework .Net que le Compact Framework. Elle s'utilise en C# et VB. Il existe des composants prêts à l'emploi pour vous faciliter la vie sous Windows (surtout si vous utilisez Telephony API) :

- TAPIx ActiveX Xontrol de HotWind Software
- KDTele Tools de KAED System
- HbTAPI Components de Hager+Busch Gbd
- NrComm Lib de DeepSoftware

Vous pouvez opter pour le SDK complet. La société Conaito fournit VoIP SDK. Il s'utilise en C++, Javascript, VB, C#... Il permet de créer

un softphone VoIP avec des fonctions de sécurité. La version standard permet uniquement la téléphonie P2P, les autres éditeurs (Extended, Entreprise) permettent de réaliser des environnements de téléconférence, du client/serveur, etc. Côté Microsoft, vous pouvez passer par Live Communications Server 2005 SDK pour créer des applications SIP et gérer le serveur Live Communications. Côté client, il existe RTC Client API.

Open source SIP / VoIP

L'une des piles les plus connues est oSIP (<http://www.gnu.org/software/osip/osip.html>). Il s'agit d'une implémentation SIP sous licence LGPL. La librairie est écrite en C et ne possède aucune dépendance (excepté la librairie C). Elle s'utilise pour un softphone desktop ou embarqué. Pour faciliter son usage, il existe le projet Exosip. Sipx (<http://www.sipfoundry.org/>) se présente comme un concurrent d'Asterisk. SipX se découpe en plusieurs projets, par exemple : sipXpbx pour créer un SIP PBX d'entreprise, sipXtapi pour créer un client SIP, sipXmediaLib, orienté purement VoIP. Il existe aussi OPAL, pile sous licence Mozilla, utilisant le standard H323.

Et côté Java

En Java, il existe plusieurs JSR ayant un rôle dans VoIP :

- JAIN SIP (JSR 32) : spécification pour échanger des informations entre client et un serveur SIP. Existe aussi une version allégée : JAIN SIP Lite (JSR 125). La partie serveur est assurée par SIP Servlet API (JSR 116)
- Java Telephony (JTAPI)

■ François Tonic

Un client IAX avec la librairie IAXClient



Le protocole IAX (Inter-Asterisk EXchange) est le protocole natif du PBX logiciel Asterisk. Il est souvent utilisé pour créer des ponts entre deux processus Asterisk distants afin de rediriger les appels de l'un à l'autre. Mais ce protocole est également utilisable par des applications clientes.

Il en existe déjà un certain nombre, tels que Kiax, iaxComm ou DIAX. Dans cet article, le terme IAX fait référence à la version 2 du protocole (également nommée IAX2). La version 1 est désormais obsolète et n'est en pratique plus utilisée. Le protocole IAX possède quelques caractéristiques qui le démarquent d'autres protocoles tels que SIP ou H.323. La principale différence avec ces derniers est l'utilisation d'un unique port pour toutes les transmissions de paquets (contrôle et données). Par défaut, il s'agit du port UDP 4569. Le fait de n'utiliser qu'un port UDP permet en particulier de pouvoir traverser les routeurs NAT et les firewalls avec beaucoup de facilité, sans l'intervention d'applications de réécriture de paquets.

Une autre caractéristique intéressante est la nature binaire du protocole (contrairement à un protocole tel que FTP dont les commandes sont des chaînes de caractères telles que USER, PASS, LIST, etc.). Ceci permet de réduire un peu l'utilisation de la bande passante. Lors de l'utilisation du mode ``trunk``, les données peuvent être rassemblées dans un même paquet IP afin d'éviter la réplication des en-têtes IP autant de fois qu'il y a de paquets. Ceci n'est cependant jamais utile pour un client. Le protocole a également été conçu pour pouvoir transférer virtuellement tout type de données (en particulier la voix, la vidéo et l'image), même si pour le moment, il est spécialisé dans le transfert de la voix et des formats et codecs associés (GSM, G.711 a-law, G.711 mu-law, G.729, etc...).

La RFC du protocole est disponible à l'adresse : <http://www.cornfed.com/iax.pdf>

La librairie IAXClient

Description

Il existe quelques librairies pour utiliser le protocole IAX. La libiax gère seulement IAX 1 et n'est plus maintenue. La libiax2 semble aussi ne plus être mainte-

nue depuis quelque temps. Les développeurs de IAXClient ont donc repris la libiax2, et ont ajouté plusieurs choses, dont une librairie audio portable, et une API au-dessus de libiax2 facilitant beaucoup l'intégration d'IAX dans une application, en gérant directement les entrées/sorties audio et l'encodage/décodage des données transférées.

Cela signifie qu'une application n'a plus qu'un contrôle abstrait à effectuer (enregistrement, appels, réception, choix des codecs, volume, etc.). La gestion d'évènement se fait grâce à une fonction callback (appelée par la librairie lorsqu'elle doit communiquer un évènement à l'application).

Cette librairie est malheureusement assez jeune, et manque cruellement de documentation. Lire le fichier iaxclient.h ainsi que le code source est parfois très utile pour comprendre certaines fonctions.

Utilisation

Dans cette partie, nous verrons donc comment utiliser la librairie pour écrire un petit softphone en C capable d'émettre un appel. Maîtriser l'API des threads POSIX est vivement recommandé. Vous pouvez également utiliser le code source à l'adresse URL comme exemple complet reprenant les points abordés dans cet article.

Premièrement, n'oubliez pas :

```
#include <iaxclient.h>
```

Les fonctions de la librairie sont préfixées par iax_. La première fonction à appeler est une fonction d'initialisation :

```
iax_initialize(AUDIO_INTERNAL_PA, 1);
```

La macro AUDIO_INTERNAL_PA indique à la librairie d'utiliser PortAudio (la librairie audio portable énoncée plus haut). Sous

GNU/Linux, ceci revient à utiliser OSS, donc si vous utilisez ALSA, n'oubliez pas de charger le module snd-pcm-oss. Le deuxième paramètre indique le nombre d'appels maximum. Une fois la librairie initialisée, on doit lui indiquer la fonction callback à utiliser pour la gestion des évènements :

```
static int
event_callback(iaxc_event event)
{
    ...;
}

iaxc_set_event_callback(event_callback);
```

Une fois que la fonction callback est fixée, on peut ``démarrer`` la librairie :

```
iaxc_start_processing_thread();
```

Le code suivant indique les codecs à utiliser :

```
iaxc_set_formats(IAXC_FORMAT_GSM, (IAXC_
_FORMAT_GSM
    | IAXC_FORMAT_ULAW
    | IAXC_FORMAT_ALAW));
```

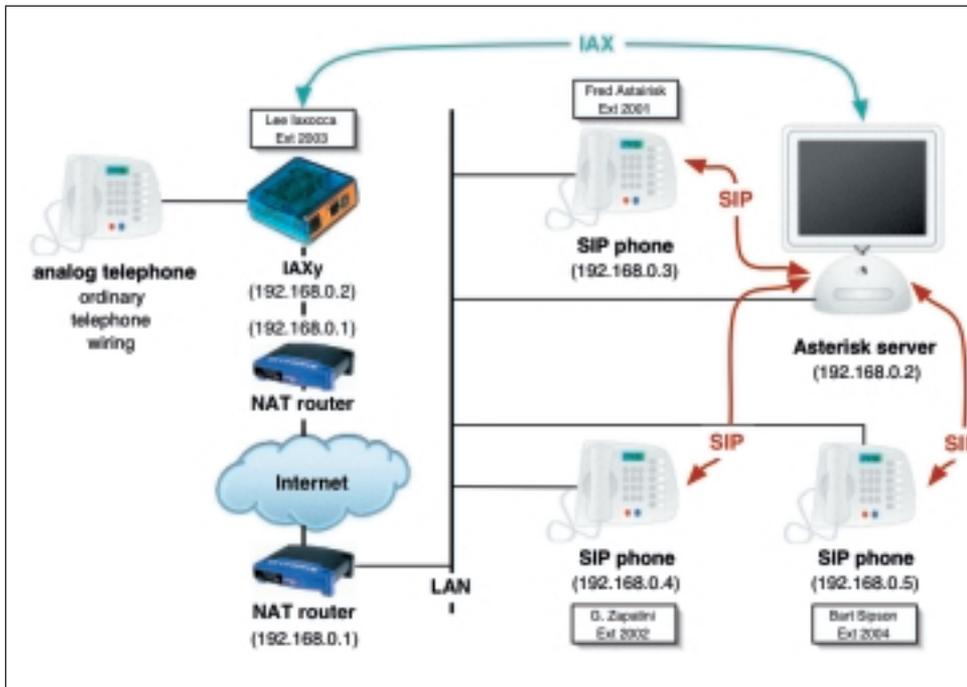
Le premier paramètre est le codec préféré, le second est la liste des codecs supportés. La librairie se charge automatiquement de la négociation du codec.

La librairie est prête. Il ne reste plus qu'à s'enregistrer puis émettre un appel :

```
iaxc_register(user, pass, host);
```

Cette fonction envoie la demande d'enregistrement à l'hôte indiqué.

Elle est asynchrone (comme toutes les fonctions de la librairie), ce qui signifie qu'elle n'indiquera pas en retour si l'enregistrement a été effectué ou pas. Cette action là sera traitée par la fonction callback dans un autre



◀ Fonctionnement d'un réseau VoIP utilisant Asterisk en serveur, le tout en protocole IAX et des matériels SIP et analogiques.

thread. Le développeur est responsable de la synchronisation entre les threads pour correctement récupérer le résultat de la demande d'enregistrement. En pratique, votre softphone est une GUI inactive qui attend des événements (comme la réponse de l'hôte, ou un click de souris).

Ce modèle asynchrone est donc plutôt adapté. Néanmoins, l'accès aux variables doit se faire de manière atomique, en utilisant des mutex. Si votre application est conforme à POSIX, utilisez l'API des threads POSIX. Si vous souhaitez la rendre davantage portable, utilisez une autre API.

Par exemple, Kiax utilise les mutex de Qt. Assurez-vous cependant que l'API que vous utilisez est compatible avec celle utilisée par IAXClient.

Une fois enregistré, le client peut émettre un appel. Faites attention au format de la destination (user:pass@host/number) :

```
dest_length = snprintf(NULL, 0, "%s:%s@%s/%s", user, pass, host, number);
dest = malloc((dest_length + 1) * sizeof(char));
sprintf(dest, "%s:%s@%s/%s", user, pass, host, number);
iaxc_call(dest);
```

Tout comme `iaxc_register()`, cette fonction ne fait qu'envoyer une demande d'appel. La gestion à proprement parler de cet appel se fera dans la fonction `callback` :

```
static int
event_callback(iaxc_event event)
{
    switch (event.type)
    {
        case IAXC_EVENT_LEVELS :
            return handle_levels(&event.ev.levels);

        case IAXC_EVENT_REGISTRATION :
            return handle_registration(&event.ev.reg);

        case IAXC_EVENT_STATE :
            return handle_call_state(&event.ev.call);
    }

    return 0;
}
```

Lisez `iaxcclient.h` pour connaître en détail les macros et structures utilisées dans cet exemple.

La valeur de retour de la fonction `callback` peut prendre 3 significations : si strictement inférieure à 0, elle indique une erreur. Si strictement supérieure à 0, elle indique un succès. Si égale à 0, elle indique un succès, et demande à la librairie d'effectuer son action par défaut. En pratique, retourner 0 a pour effet l'affichage de messages indiquant l'état d'un appel (exemple: "IAXCLIENT: Call 0 answered").

Dans le cas d'un événement `IAXC_EVENT_REGISTRATION`, il faut analyser les champs `id` et

`reply` de `event.ev.reg`. Un ID est attribué pour chaque client.

Le champ `reply` indique le résultat de l'enregistrement. Il peut valoir `IAXC_REGISTRATION_REPLY_ACK` en cas de réussite, `IAXC_REGISTRATION_REPLY_REJ` si l'enregistrement a échoué ou `IAXC_REGISTRATION_REPLY_TIMEOUT` si l'hôte ne répond pas.

L'évènement `IAXC_EVENT_LEVELS` permet de connaître les niveaux de volume des flux entrants et sortants (champs `input` et `output`).

L'évènement `IAXC_EVENT_STATE` permet de connaître l'état d'un appel. Le champ `format` indique le codec utilisé. Le champ `callNo` indique le numéro de l'appel. La valeur 0 indique un numéro inconnu. Le champ `state` est un ensemble de drapeaux indiquant l'état de l'appel, tels que `IAXC_CALL_STATE_ACTIVE`, `IAXC_CALL_STATE_OUTGOING`, `IAXC_CALL_STATE_RINGING`, etc.

Lorsque `IAXC_CALL_STATE_ACTIVE` n'est pas présent dans le champ `state`, l'appel a été fermé par le correspondant. En revanche, pour initier la fin de l'appel, utilisez la fonction `iaxc_select_call()` pour indiquer à la librairie quel appel est l'appel courant, puis la fonction `iaxc_dump_call()` pour le terminer.

La fonction `iaxc_shutdown()` termine tous les appels en cours et stoppe la gestion des périphériques audio. Il ne reste plus qu'à appeler `iaxc_unregister()`.

Lexique

IAX : protocole de voix sur IP. Il fut conçu par le projet open source Asterisk. Permet la communication entre le client et le serveur et entre serveurs. Non standard, mais connaît un succès croissant.

Liens annexes

- Asterisk** : <http://www.asterisk.org>
- VOIP** : <http://www.voip-info.org>
- Proformatique** : <http://www.proformatique.com>
- Proformatique communauté** : <http://www.proformatique.org>

■ **Richard Braun**
 Développeur C
rbrun@proformatique.com
 Proformatique - <http://www.proformatique.com>

La VoIP avec Java Media Framework

Nous tendons à mieux communiquer, et dans notre quotidien, nous sommes entourés par le multimédia : qu'il s'agisse de télévision, de radio, de visioconférences ou tout simplement de téléphone. Notre exigence naturelle manifeste des besoins de plus en plus affinés concernant la gestion de l'information. JMF répond à ce besoin ... Il s'agit d'une API extensible permettant de manipuler des médias temporels. De nombreuses fonctionnalités sont destinées à faciliter la transmission et le traitement de la vidéo et du son.

Le JMF n'étant pas une API standard, il sera nécessaire de le télécharger puis de l'installer sur votre poste. Nous utiliserons ici la dernière version à ce jour : la version 2.1.1e (disponible à l'adresse suivante : <http://java.sun.com/products/java-media/jmf/downloads>).

Concept du JMF

1 Lecture d'un flux audio : Le Player dans tous ses états

La classe principale pour lire un fichier audio est la classe Player. Afin de lire un fichier audio, le Player passera par différents états :

- **Unrealized** : Etat initial après instanciation du Player
- **Realized** : Reconnaissance des ressources à utiliser. Il s'agit d'une phase d'initialisation permettant de rassembler les informations nécessaires au lancement d'une ressource.
- **Prefetched** : Acquisition des ressources. Elles sont réservées et prêtes à être utilisées.
- **Started** : Lancement de la ressource.

Les états Realized et Prefetched sont des étapes nécessitant un certain temps de traitement. Aussi une étape ne peut être exécutée que si l'étape précédente a été validée. C'est pourquoi des états intermédiaires sont présents afin de mieux suivre le déroulement de l'opération :

- Realizing, entre Unrealized et Realized
- Prefetching, entre Realized et Prefetched

Un gestionnaire d'événement permet de suivre les changements d'état. A chaque changement d'état, la méthode controllerUpdate() de l'interface ControllerListener est appelée. En nous abonnant à un ControllerListener, il nous sera donc possible de capturer les étapes afin d'effectuer les opérations voulues.

ce qui permet de grandement simplifier son utilisation. Pour notre plus grand bonheur, un Manager est destiné à la gestion des transmissions de flux par RTP : Le RTPManager. La création d'un RTPManager nécessitera 2 informations : la source et une référence vers l'objet externe qui va communiquer avec le RTPManager.

Contexte

Nous allons travailler avec 3 fichiers sources :

- Sender.java contenant le code lié à l'émetteur. Il s'occupera de la capture de la voix et de l'envoi du flux audio vers le récepteur.
- Receiver.java contenant le code lié au récepteur. Il sera en charge de recevoir le flux et d'effectuer une lecture du flux audio.
- Constants.java qui contiendra toutes les variables globales de l'application.

Paramètres globaux de l'application

```
public class Constants
{
    public static String SENDER_IP = "192.168.0.1";
    public static int SENDER_PORT = 2006;

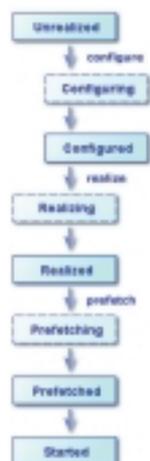
    public static String RECEIVER_IP = "192.168.0.1";
    public static int RECEIVER_PORT = 6002;
}
```

SENDER_IP : adresse IP de l'émetteur.

SENDER_PORT : port d'écoute de l'émetteur.

RECEIVER_IP : adresse IP du récepteur.

REVEICER_PORT : port d'écoute du récepteur.

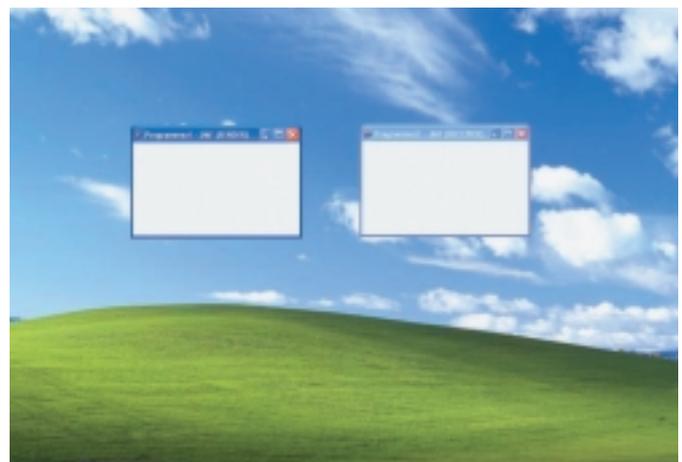


2 Capture vocale : le Processor

La capture d'un vocal avec un microphone s'effectue à l'aide de la classe Processor qui hérite directement de Player. A la différence de Player, Processor apporte 2 états supplémentaires situés entre Unrealized et Realized : Configuring puis Configured. Ces états permettent notamment de traiter le démultiplexage des flux et de définir le format des données.

3 Transmission du flux : RTPManager

La transmission du flux se base sur le protocole RTP. JMF dispose d'un ensemble de classes Managers,





Lire un flux audio

La classe Receiver

```
import javax.media.*;
import javax.media.rtp.*;
import javax.media.rtp.event.*;

import javax.swing.JFrame;
import java.net.InetAddress;

public class Receiver implements ReceiveStreamListener {

    public Receiver() {}

    public void buildGUI() {
        JFrame mainFrame = new JFrame();

        mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        mainFrame.setTitle("Programmez! - JMF (RECEIVER)");
        mainFrame.setSize(300, 200);

        mainFrame.setVisible(true);
    }

    public void init() {
        /* Création du gestionnaire de flux RTP */
        RTPManager manager = RTPManager.newInstance();

        try {
            /* Configuration de l'adresse locale */
            SessionAddress localAddress = new SessionAddress(InetAddress.
getByName(Constants.RECEIVER_IP), Constants.RECEIVER_PORT);
            manager.initialize(localAddress);

            /* Configuration de l'adresse de l'émetteur */
            SessionAddress sourceAddress = new SessionAddress(Inet
Address.getByName(Constants.SENDER_IP), Constants.SENDER_PORT);
            manager.addTarget(sourceAddress);
        }
        catch (Exception e) {
            e.printStackTrace();
        }

        /* Abonnement aux evenements de reception de flux */
        manager.addReceiveStreamListener(this);
    }

    public void update(ReceiveStreamEvent event) {

        if (event instanceof NewReceiveStreamEvent) {
            ReceiveStream stream = event.getReceiveStream();
            try {
                /* Création d'un Player pour lire le flux */
                Player player = Manager.createRealizedPlayer(stream.get
DataSource());
                player.start();
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}

public static void main(String[] args) {
    Receiver receiver = new Receiver();

    receiver.buildGUI();
    receiver.init();
}
}
```

1. Création de l'interface graphique : buildGUI()

Pour représenter notre application, nous allons créer une interface graphique symbolique. Il nous permettra simplement de gérer l'ouverture et la fermeture de l'application.

2. Initialisation du gestionnaire de flux : init()

L'instanciation du RTPManager s'effectue à l'aide de la méthode RTPManager.newInstance().

Avant que le RTPManager puisse gérer la transmission du flux, nous devons l'initialiser en lui indiquant son adresse locale à l'aide de la méthode initialize()

```
SessionAddress localAddress = new SessionAddress(InetAddress.get
ByName(Constants.RECEIVER_IP), Constants.RECEIVER_PORT);
manager.initialize(localAddress);
```

Une fois cela effectué, la méthode addTarget() nous permettra de lui indiquer l'adresse source à partir de laquelle va être envoyé le flux en question.

```
SessionAddress sourceAddress = new SessionAddress(InetAddress.get
ByName(Constants.SENDER_IP), Constants.SENDER_PORT);
manager.addTarget(sourceAddress);
```

Enfin, nous nous abonnons au gestionnaire d'événement ReceiveStreamListener qui nous permettra de capturer chaque arrivée de flux et d'effectuer un traitement spécifique, en l'occurrence la lecture audio.

3. Lecture du flux audio transmis : update()

À chaque réception de flux, la méthode update(), héritée de ReceiveStreamListener, est exécutée.

On obtient la nature de l'opération par l'intermédiaire du typage de l'événement ReceiveStreamEvent. Nous souhaitons récupérer l'arrivée d'un nouveau flux, et le typage associé est NewReceiveStreamEvent.

```
if (event instanceof NewReceiveStreamEvent)
```

Une fois l'événement capturé, nous récupérons le flux en question et effectuons la lecture à l'aide d'un Player.

```
ReceiveStream stream = event.getReceiveStream();
try {
    Player player = Manager.createRealizedPlayer(stream.getDataSource());
}
```

Gros plan

```
player.start();
}
```

Le Player a la possibilité de s'abonner à un événement lui permettant de suivre de façon plus précise les différents états de lecture. Il s'agit du gestionnaire d'événement ControllerManager. Nous utilisons ici la méthode createRealizedPlayer() qui gère de elle-même les étapes de transition en s'assurant que chaque étape d'initialisation du Player a correctement été réalisée avant de pouvoir effectuer la lecture.

Capter un flux audio

La classe Sender

```
import javax.media.*;
import javax.media.rtp.*;
import javax.media.protocol.*;

import javax.swing.JFrame;
import java.net.InetAddress;

public class Sender implements ControllerListener {

    private Processor processor = null;
    private SendStream stream = null;

    public Sender() {}

    public void buildGUI() {
        JFrame mainFrame = new JFrame();

        mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        mainFrame.setTitle("Programmez! - JMF (SENDER)");
        mainFrame.setSize(300, 200);

        mainFrame.setVisible(true);
    }

    public void init() {
        try {
            /* Récupération du périphérique */
            CaptureDeviceInfo captureDevice = CaptureDeviceManager.getDevice("JavaSound audio capture");

            /* Création d'un Processor pour capturer le son */
            processor = Manager.createProcessor(captureDevice.getLocator());

            /* Configurer le périphérique */
            while (processor.getState() < Processor.Configured) {
                //Configuration du Processor
                processor.configure();
            }

            /* Configuration du type de données à traiter (Fichier WAV) */
            processor.setContentDescriptor(new ContentDescriptor(ContentDescriptor.RAW_RTP));
        }
    }
}
```

```
/* Abonnement aux evenements de lecture du son */
processor.addControllerListener(this);

/* Initialisation de la ressource */
processor.realize();
}
catch (Exception e) {
    e.printStackTrace();
}
}

public void initializeOutput() {
    /* Récupérer la source à envoyer */
    DataSource source = processor.getDataOutput();

    RTPManager manager = RTPManager.newInstance();

    try {
        /* Configuration de l'adresse locale */
        SessionAddress localAddress = new SessionAddress(InetAddress.
        getByName(Constants.SENDER_IP), Constants.SENDER_PORT);
        manager.initialize(localAddress);

        /* Configuration de l'adresse du receveur */
        SessionAddress destAddress = new SessionAddress(InetAddress.
        getByName(Constants.RECEIVER_IP), Constants.RECEIVER_PORT);
        manager.addTarget(destAddress);

        /* Envoi du flux RTP */
        SendStream stream = manager.createSendStream(source, 0);
        stream.start();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}

public void controllerUpdate(ControllerEvent e) {

    if (e instanceof RealizeCompleteEvent) {
        initializeOutput();
        processor.start();
    }
}

public static void main(String[] args) {
    Sender sender = new Sender();

    sender.buildGUI();
    sender.init();
}
}
```

1. Création de l'interface graphique : buildGUI()

De même que pour la classe Receiver, une interface graphique similaire sera associée au Sender.

2. Préparation de la capture : init()

L'instanciation d'un Processor nécessite une ressource d'origine. Souhaitant capturer une voix, nous devons obtenir un MediaLocator permettant de référencer diverses ressources : un fichier audio, des protocoles de transfert ou, dans notre cas, un appareil de capture tel qu'un microphone.

Vous pourrez trouver la liste de tous les périphériques reconnaissables par JMF à l'aide du programme ci-dessous.

```
public class ListDevices {  
  
    public static void main(String[] args) {  
  
        /* Parcourir la liste des peripheriques */  
        Vector devices = (Vector)CaptureDeviceManager.getDeviceList(null);  
        for (int i=0; i < devices.size(); i++) {  
  
            /* Afficher un périphérique */  
            CaptureDeviceInfo currentDevice = (CaptureDeviceInfo)devices.get(i);  
            System.out.println("\n>> " + currentDevice.getName());  
        }  
    }  
}
```

Ainsi, il vous sera possible de renseigner la méthode CaptureDeviceManager.getDevice() renvoyant un CaptureDeviceInfo qui représentera le matériel de capture. La méthode getDevice() demande un nom symbolique de périphérique. Souvent "JavaSound audio capture" suffira et vous permettra de référencer votre matériel de capture par défaut.

A partir du CaptureDeviceInfo, nous obtenons un MediaLocator qui nous permettra de récupérer notre Processor.

```
processor = Manager.createProcessor(captureDevice.getLocator());
```

L'étape de configuration du périphérique permet d'initialiser les informations liées au traitement audio tel que le format. L'appel de la méthode configure() n'est pas bloquante, ce qui explique pourquoi une boucle est nécessaire afin de vérifier que le Processor est correctement configuré avant de passer à l'étape suivante.

```
while (processor.getState() < Processor.Configured) {  
    //Configuration du Processor  
    processor.configure();  
}
```

Le Processor doit être associé à un type de flux afin de définir le type de données qui lui sera envoyé. Par exemple, dans le cas d'une sauvegarde d'une capture dans un fichier WAV, nous utiliserons plutôt un FileDescriptor(FileTypeDescriptor.WAVE).

```
processor.setContentDescriptor(new ContentDescriptor(ContentDescriptor.RAW RTP));
```

Enfin, nous nous abonnons au gestionnaire d'événement ControllerListener permettant de capturer les états de lecture et initialisons le Processor à l'aide de la méthode realize().

3. Capture des états du Processor : controllerUpdate()

Le gestionnaire d'événement ReceiveStreamListener que nous avons vu dans le Receiver, fonctionne de façon similaire au ControllerListener. En effet, le typage de l'événement ControllerEvent va définir l'état du Processor.

L'état qui nous intéresse ici est RealizeCompleteEvent qui sera envoyé lorsque toutes les ressources du Processor seront disponibles et exploitables. Nous lancerons alors l'initialisation du gestionnaire de flux pour l'envoi, et ensuite lorsque tout sera prêt, nous débuterons la capture vocale sur le microphone.

```
public void controllerUpdate(ControllerEvent e) {  
  
    if (e instanceof RealizeCompleteEvent) {  
        initializeOutput();  
        processor.start();  
    }  
}
```

4. Initialisation du gestionnaire de flux : initOutput()

L'initialisation du RTPManager s'effectue de la même façon que précédemment. Nous utiliserons donc initialize() pour renseigner l'adresse locale et addTarget() pour indiquer l'adresse de destination. Une fois le RTPManager disponible, nous récupérons un SendStream permettant de démarrer l'envoi du flux audio.

```
SendStream stream = manager.createSendStream(source, 0);  
stream.start();
```

Enfin, nous démarrons le Processor qui marquera le début de la capture vocale du microphone.

```
processor.start();
```

Autres possibilités

L'API JMF offre de nombreuses possibilités de traitement de flux multimédia. Nous avons choisi de traiter ici un flux audio cependant, JMF étant une API générique destinée au média, le traitement d'un flux vidéo ne diffère pas de façon significative. JMF est fidèle à la volonté de Sun, qui est de simplifier au mieux la tâche du développeur. Il propose d'ailleurs un ensemble de classes à cet effet, nous avons pu en voir un exemple à travers le RTPManager. Nous pouvons également citer le StateHelper permettant de faciliter la manipulation des états d'un Processor ou encore des méthodes générant des interfaces et permettant de lire un flux vidéo.

D'autres fonctionnalités du JMF permettent de sauvegarder des fichiers sur le disque ou encore d'étendre les possibilités du JMF lui-même, via le développement de ses propres plug-ins.

Références

JMF, page d'accueil du site officiel

<http://java.sun.com/products/java-media/jmf/>

JMF, Guide officiel de Sun

<http://java.sun.com/products/java-media/jmf/2.1.1/guide/>

■ Eric CHAYSINH - echaysinh@devutil.net

Ajax : les outils pour le web 2.0

On ne le dira jamais assez, le web 2.0, avec AJAX, est une vraie révolution pour le développement web. Mais si le gain est appréciable pour l'utilisateur, en termes de confort et d'ergonomie, il n'en va pas de même pour le développeur. Le code réclamé par ce type d'applications est souvent très verbeux, voire confus, et dans le cadre d'un projet important, cela peut devenir très difficile à gérer. Certains produits sont aujourd'hui disponibles pour faciliter la création d'applications web 2.0, avec plus ou moins de réussite.

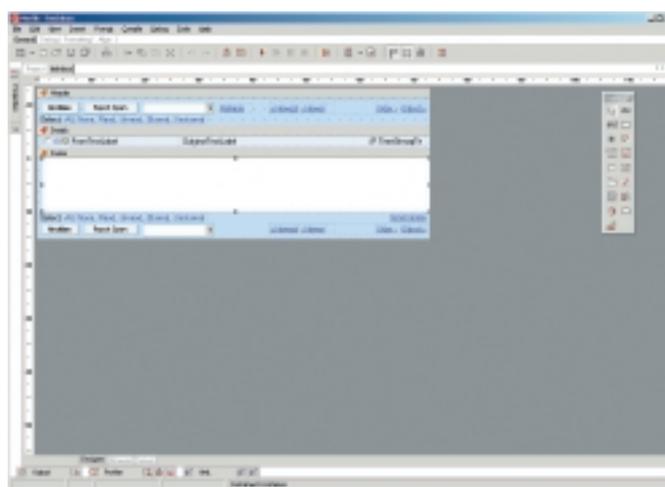
La mode, et la logique devrait-on dire, est au développement suivant les règles du pattern MVC, pour Model – View – Controller. Cette méthode permet de scinder efficacement la description des objets et règles métiers (Model), l'affichage de ces objets (View) et les interactions entre les deux (Controller). Si les développeurs web aspiraient à mettre en pratique ce pattern depuis longtemps, cela leur était rendu difficile par la pauvreté des environnements disponibles, ainsi que certaines limitations induites par le protocole HTTP. Ou du moins par la manière dont ils s'en servaient. La révolution Ajax, c'est précisément une nouvelle manière d'appréhender le protocole HTTP, en étant en mesure de dialoguer librement, sur demande, avec le serveur depuis le client, sans forcément devoir réafficher une page complète.

Ce faisant, le pattern Ajax a permis la mise au point d'environnements MVC capables de dissocier les données de leurs conteneurs bien plus efficacement qu'auparavant. En effet, dans de nombreuses pages web, on pouvait estimer que le volume de données transmises sur le réseau pour le seul squelette d'une page pouvait représenter jusqu'à 80% du volume total transféré. Et ce, à chaque "écran" de l'application. Ajoutez à cela les très faibles débits Internet qui étaient encore majoritaires il n'y a pas si longtemps, et vous comprendrez

pourquoi les applications web ont peiné à convaincre ces dernières années, malgré la présomption d'un potentiel énorme.

En faisant procéder à des échanges ponctuels avec le serveur, Ajax a permis de réduire drastiquement cette proportion, notamment en évitant de réafficher l'intégralité d'une page pour simplement mettre à jour les données affichées. Un exemple typique : l'affichage d'une liste composée de nombreux enregistrements. Lorsque l'affichage requiert plusieurs pages, Ajax permet de ne mettre à jour que le contenu des cellules du tableau qui affiche les données, plutôt que l'intégralité de l'écran.

En outre, Ajax reposant sur JavaScript, son avènement a remis au goût du jour ce langage, trop longtemps déconsidéré par les développeurs. Désormais, il est abondamment utilisé dans la réalisation d'applications web 2.0, de façon à transférer une partie de la charge de traitement des données (comme les tris ou le formatage) sur le poste client, déchargeant d'autant le serveur. Mais tout ceci ne se fait pas sans douleur. Si JavaScript a été si longtemps boudé, ce n'est pas que par snobisme. C'est également parce que c'est un langage assez inconfortable, et pour lequel il n'existe que très peu d'outils professionnels de développement. Son emploi au sein d'un projet web 2.0 peut parfois prendre des allures de cauchemar. Voici ce que les



environnements dont nous allons parler maintenant sont censés vous épargner.

Les différents types d'outils

Selon les éditeurs, l'approche du problème que représente la production d'applications web 2.0 a été envisagée sous des angles radicalement différents. Certains ont pensé favorable de proposer au développeur un IDE complet, intégrant l'ensemble des outils nécessaires au développement. C'est le cas par exemple de SweetDev Studio, qui repose sur la plate-forme Eclipse. De fait, le langage serveur associé est Java (avec JSP bien sûr). D'autres vont encore plus loin, comme Morfik Web OS, qui propose non seulement un environnement, mais également son propre langage de développement. Dans le cadre de ce comparatif, nous avons retenu ces deux outils, plus deux autres : Tibco General Interface et Backbase. Ce dernier est le seul à

ne pas disposer de véritable IDE, mais offre en revanche des outils d'aide au développement suffisamment évolués pour que l'on puisse malgré tout le rattacher à cette catégorie. Backbase est également le plus souple des quatre, mais nous en reparlerons ultérieurement.

Les produits

Sur ce marché encore récent, aucun consensus n'est établi. En effet, si deux tableurs aujourd'hui se ressemblent très fortement, il n'en va pas de même pour les environnements de développement web 2.0.

Morfik WebOS AppsBuilder

Morfik est l'un des plus ambitieux des quatre logiciels. Comme nous l'avons dit, il propose un environnement très complet et très cohérent, intégrant son langage propre. Malgré ce parti pris, les créateurs de Morfik ont voulu aider la transition pour les déve-

loppeurs déjà habitués à un langage, en proposant le leur en quatre syntaxes différentes, inspirées de C#, Java, Pascal et Basic. Ce langage de haut niveau est la pierre angulaire de Morfik. Il intègre la technologie brevetée et âprement protégée JST : JavaScript Synthesis Technology. Il s'agit en réalité d'un compilateur qui traduit le code Morfik en JavaScript. Le résultat de cette compilation est un exécutable qui contient à la fois le serveur web et les sources générées HTML/JavaScript de l'application. Une base de données intégrée permet également de procéder à la conception intégrale de son application au sein de l'environnement Morfik. Cette base de données exploite le moteur Firebird. Souvent comparé à des RAD du type d'Access, Morfik a soulevé un certain enthousiasme lors de sa sortie il y a quelques mois. Certains sont allés jusqu'à comparer l'ampleur de l'innovation proposée par ce produit avec ce que fut Visual Basic 1.0 en son temps. Ce qui est probablement exagéré. S'il est vrai que la cohérence de l'environnement a son charme, les concepts mis en oeuvre sont assez éloignés de ce à quoi le développeur web est habitué. Aussi la transition risque-t-elle d'être assez difficile. En revanche, quel qu'un ayant déjà travaillé avec un RAD devrait se sentir plus à l'aise. Pour le moins en théorie, car si l'idée générale est séduisante, et que certaines des innovations technologiques apportées par Morfik sont dignes d'intérêt, le produit est encore très jeune. Trop jeune pour envisager son utilisation en production en tout cas.

Appréciation de 1 à 5

Installation : 5 R.A.S.
Utilisation : 2 Interface buggée, compilateur JST très lent
Déploiement : 2 Le serveur web embarqué est une idée très discutée...

Documentation : 2 Très incomplète ; projets d'exemple

Général : 2
 Malgré un concept séduisant, cette bêta de Morfik ne convainc pas. La réalisation n'est pas à la hauteur des ambitions affichées. En outre, à trop vouloir simplifier le développement Ajax, qui par essence est complexe car trop peu formalisé, Morfik s'est largement éloigné de l'intérêt intrinsèque de cette technologie. De plus, la conception des écrans de l'application, imposant des zones "Headers", "Details" et "Footer", est fastidieuse, et l'organisation générale des éléments graphiques permet difficilement la réutilisation d'éléments. Une copie à revoir.

Licence : Commerciale
Tarif : N/C

Tibco General Interface

Ce produit concurrent, offrant également une interface de type RAD, a fait des choix radicalement différents de Morfik. La première surprise concerne la nature de l'IDE, qui est lui-même une application web ! Autant l'avouer, ceci est très surprenant, d'autant plus lorsque l'on a constaté combien cet environnement est complet. Proposé pour Internet Explorer sous forme d'HTML Application (fichiers .hta) et sous forme de fichiers HTML classiques pour Firefox, la partie Builder de Tibco General Interface constitue en soi un exemple très parlant des capacités de ce framework. Les widgets sont nombreux et de qualité. Si bien sûr une copie d'écran ne permet pas de faire la différence avec une application lourde, l'utilisation ne le permet guère plus. Les différents menus, fenêtres, grilles et autres boîtes à onglets sont globalement bien réactifs, et d'une qualité graphique irréprochable. De ce point de vue, TGI Builder est une véri-

NOUVEAU !

- ▶ Une méthode d'apprentissage révolutionnaire pour améliorer rapidement vos pages et sites web avec Ajax.
- ▶ Un guide pour intégrer efficacement Ajax à vos divers développements.
- ▶ 80 hacks pour doper vos applications Web.
- ▶ Tout ce qu'il faut savoir pour concevoir des pages web performantes et aux normes. En bref, la bible de tout concepteur Web !
- ▶ Guide d'apprentissage pratique pour maîtriser rapidement Ruby on Rails.
- ▶ 337 solutions adaptables et exemples pour développer en Ruby.
- ▶ 100 trucs et techniques pour dynamiser vos sites web.
- ▶ Une méthode d'apprentissage révolutionnaire et incroyablement efficace pour créer rapidement des pages web modernes et vraiment professionnelles.
- ▶ Jennifer Tidwell vous révèle à travers des modèles commentés, l'art de créer des interfaces utilisateurs ergonomiques et efficaces pour tous types de supports : Web, CD-ROM, PDA, ou téléphones mobiles.

Recevez notre nouveau catalogue sur simple demande à info@oreilly.fr

www.oreilly.fr

Ouvrages en vente dans toutes les librairies !

L'INFORMATIQUE À LA SOURCE
O'REILLY®

O'REILLY® News
 Abonnez-vous à notre e-newsletter sur www.oreilly.fr et recevez tous les mois des informations sur notre actualité.

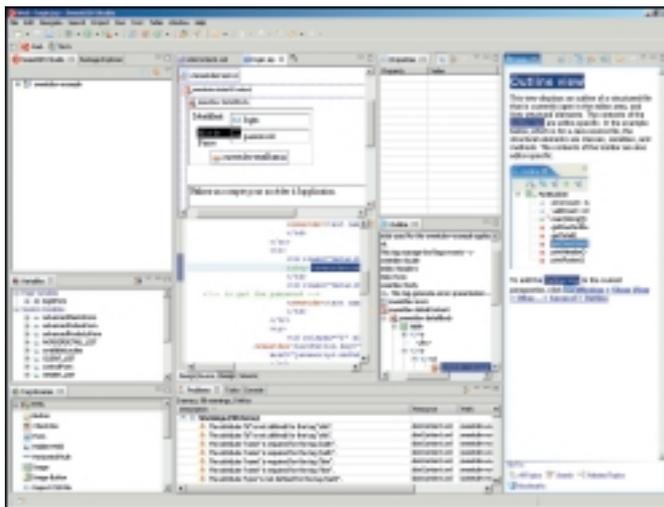
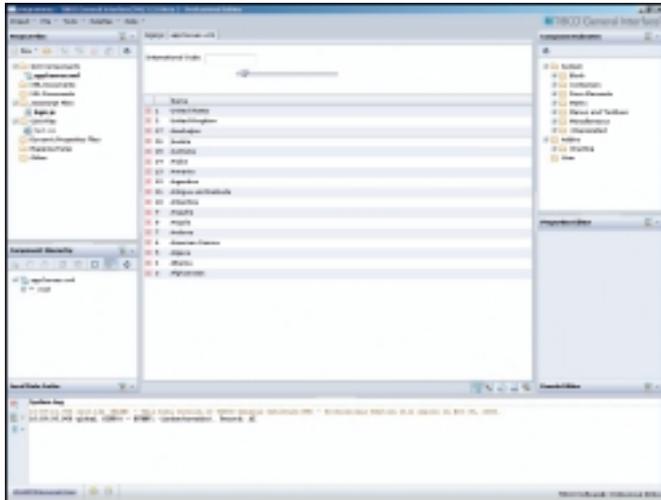


table réussite. Technologiquement, l'orientation est plus conventionnelle. Les langages employés sont le JavaScript, natif, pour la logique métier, et le XML pour la couche de présentation, ainsi que, le cas échéant, le stockage et/ou l'échange de données. Ce choix permet d'envisager aisément l'intégration de n'importe quel type de données, qu'il soit capable ou non de s'exporter nativement en XML. En effet, rien de plus simple que de générer un arbre XML à partir d'un jeu d'enregistrement provenant, par exemple, d'une base MySQL. Un autre point très intéressant, la scission entre la partie "Builder", c'est-à-dire l'IDE à proprement parler, et la partie "Framework". Le framework, composé exclusivement de fichiers JavaScript,

peut être déployé par simple copier/coller des fichiers en question sur n'importe quel serveur web. L'administrateur est ainsi libre d'opter pour la solution d'hébergement qui lui convient le mieux, et le développeur, pour le langage côté serveur qui a sa préférence. La stratégie de Tibco pour faciliter la mise au point d'applications Ajax passe donc par XML concernant la présentation. Les widgets sont décrits dans un dialecte XML propre au framework, qui permet la génération à la volée du code HTML/JavaScript nécessaire à l'exécution de l'application écrite. Ceci permet de réaliser aisément des composants (simples fichiers XML), qui peuvent être réutilisés au sein d'une même application, aussi bien que dans un autre projet.

Appréciation de 1 à 5

Installation : 5

R.A.S. (une archive à dézipper)

Utilisation : 3

Une note due à la lenteur d'exécution du Builder dans le cas d'un projet important. Sinon, le seul regret est que l'on ne puisse pas positionner les éléments à la souris. Autre déception, le Builder étant en mode web, il ne permet pas la coloration syntaxique des sources en mode édition.

Déploiement : 4

La copie des sources sur le serveur suffit à déployer le framework. Un assistant facilite la création de fichiers d'initialisations des applications.

Documentation : 4

Assez fouillée, et suivie, la documentation est complétée par un site dédié, avec forum communautaire sur lequel interviennent régulièrement des développeurs du framework.

Général : 4

L'un des produits les plus intéressants de ce comparatif, grâce à son universalité (notons entre autre les efforts produits pour supporter parfaitement Firefox 1.5), sa rigueur et la richesse de sa bibliothèque (l'add-in de génération de graphique, fourni, est vraiment fameux). Son modèle MVC bien implémenté permet en outre de bien encadrer les développeurs.

Licence : BSD

(depuis la bêta 3.2)

Tarif : gratuit

SweetDev Studio

Proposé comme plug-in pour la plateforme Eclipse, cet IDE annonce d'emblée son orientation. Pour IDEO Technologies, le développement sera Java ou ne sera pas. A l'instar de Tibco GI, SweetDev se compose d'un framework et d'un IDE. Si le framework SweetDev RIA (Rich Internet Applications) est disponible de manière autonome (et gratuite), y

recourir sans utiliser l'IDE de l'éditeur reste une gageure.

En effet, le framework est lui-même dérivé de Struts, framework de création d'applications web en JSP, lequel a été modifié en vue de son intégration aux éléments propres à SweetDev, qui enrichissent la plateforme de plusieurs dizaines de modules Ajax en JSP prêts-à-l'emploi. De plus, ont été intégrés à la dernière version du Studio le support de Hibernate et JSF.

Parmi les fonctionnalités particulièrement remarquables de ce produit, par rapport à sa concurrence directe, il faut notamment citer l'environnement d'édition JSP, qui est une vraie réussite. Non seulement l'éditeur permet une composition aisée des pages par glisser/déposer des composants, mais il offre en plus une synchronisation parfaite avec la source générée à la volée. Il est de plus possible de déboguer les scripts JSP pas à pas. L'édition des JSP comme des XML dispose d'un mécanisme d'auto-complétion très efficace, et agréable à utiliser. Il est regrettable que les tags HTML ne bénéficient pas du même support. Grâce au moteur AppXRay embarqué dans l'éditeur, SweetDev maintient en permanence une table de liaisons entre les éléments composant un projet, et peut ainsi indiquer au développeur en temps réel (sans attendre la compilation et l'exécution) les éventuelles erreurs présentes dans le code, épargnant ainsi à ce dernier de fastidieuses occurrences de cycles développement/debuggage.

Appréciation de 1 à 5

Installation : 3

Nécessite la présence d'Eclipse et d'un serveur d'applications Java EE.

Utilisation : 3

Les utilisateurs habitués à Eclipse seront tout à fait à leur aise. Les nombreux assistants et la synchronisation de l'édition vi-

suelle et des sources rendent le produit plutôt confortable. Malheureusement, de nombreux stack overflow viennent contrarier l'expérience générale.

Déploiement : ②

Déployer une application Java EE n'a jamais été le plus simple sur le web. SweetDev souffre donc naturellement de cet héritage.

Documentation : ②

Assez incomplète, malgré un tutorial de démarrage intéressant.

Général : ④

SweetDev Studio permet de piloter Struts comme le framework SweetDev depuis une interface riche, des outils de contrôle puissants et une bibliothèque de qualité. Faire du JavaScript en Java, voilà qui devrait ravir les développeurs amoureux du langage de Sun. Les autres risquent fort de ne pas changer d'avis.

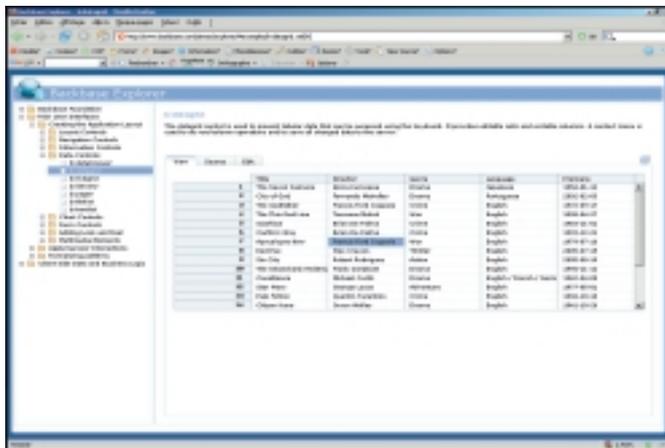
Licence : *Commerciale*

Tarifs : à partir de 995 € H.T. par développeur et par an

BackBase

Nous revoici donc face à BackBase, dont nous avons déjà parlé il y a quelque temps dans Programmez. Il s'agit d'un des pionniers en matière de framework Ajax, et probablement aujourd'hui l'un des plus aboutis. Bien qu'il ne s'agisse pas à proprement parler d'un IDE, BackBase propose des outils de développement qui vont bien au-delà des frameworks classiques, c'est pourquoi nous l'avons inclus dans ce comparatif.

Une simple consultation du "BackBase Explorer" sur le site officiel permet de constater la richesse et la qualité de ce framework. Qu'il s'agisse des éléments de présentation (cadres, fenêtres, menus...), de formulaires (sélecteur de date, combo box...) ou encore de fonctionnalités de traitement (tris, validations...), les composants sont nombreux, cohérents graphiquement et



efficaces. Techniquement, la plupart des éléments d'une application BackBase s'écrivent en HTML ou avec le dialecte XML propriétaire du framework : BXML. Tout comme c'est le cas pour Tibco GI, le choix du XML permet de forcer une structure rigide, et donc sans erreur possible à l'exécution, mais d'un autre côté, cela implique un code assez verbeux. Dans certains cas, on est très surpris du nombre de lignes nécessaires à l'obtention de composants somme toute assez simples. Pour pallier à ce défaut, et pour séduire une palette de développeurs souvent attachés à leur environnement favori, BackBase est proposé en trois versions : Client (tout HTML/BXML), Java et .Net. Ces deux dernières offrent des bibliothèques qui peuvent être embarquées sur les serveurs d'applications respectifs des plates-formes, et ainsi générer les composants BackBase à partir d'objets natifs de son langage favori. Il faut préciser cependant que rien n'interdit de recourir à un langage tiers, dans le cas de l'édition "Client", pour générer le code BXML. Mais cela implique en ce cas de produire du code déjà disponible en Java ou .Net.

Appréciation de ① à ⑤

Installation : ⑤

Comme il s'agit d'un framework simple, seules les sources sont

nécessaires sur le serveur.

Utilisation : ④

Dans la version Client, les seuls outils disponibles pour le développeur sont les outils de debugage, très complets et vraiment impressionnants. Dans les versions Java et .Net, des plug-ins sont disponibles respectivement pour Eclipse et Visual Studio, pour un confort optimal.

Déploiement : ⑤

La version " Client " ne demande que les sources sur l'espace web, et l'inclusion dans les scripts du JavaScript boot.js. Pour les éditions Java et .Net, pas de difficultés majeures, pour peu que l'on maîtrise le serveur d'application de sa plate-forme.

Documentation : ⑤

Fouillée et quasi exhaustive, la documentation des widgets est excellente. On regrettera simplement qu'elle ne soit fournie qu'au format PDF. Sinon, l'Explorer permet de tester et comprendre

aisément le fonctionnement des différents composants du framework.

Général : ④

BackBase est une valeur sûre. Un framework éprouvé, complet et bien documenté. La possibilité de créer ses propres widgets, les outils d'aide au développement ainsi que la riche documentation permettent une productivité vraiment améliorée pour le développement d'applications web 2.0.

Licence : *Commerciale*

Tarifs : *N/C (une version communautaire est disponible gratuitement pour les sites non-commerciaux et non-institutionnels)*

Conclusion

Malgré sa très rapide généralisation, le développement web 2.0 n'en est qu'à ses tous premiers pas. Et les outils qui lui sont dédiés également, bien sûr. Aussi, on ne peut que saluer les efforts déjà réalisés par les divers éditeurs et communautés qui travaillent à la réalisation d'outils plus performants, confortables et efficaces. Et bien que ceux-ci ne soient pas forcément aujourd'hui à la hauteur de nos attentes, voire parfois de nos besoins, la rapidité avec laquelle ils ont été mis au point nous laisse à penser que dès l'année prochaine, de très grands progrès auront encore été réalisés.

■ **Gauthier Delamarre**

Ne pas confondre applications web et composants Ajax

Les logiciels abordés dans ce comparatif ont pour vocation première d'encadrer le développement d'applications web, qu'il ne faut pas confondre avec des sites web, même si dans certains cas (boutiques en ligne, centrales de réservation, etc.), les problématiques se rejoignent. Pour aborder ce deuxième cas, celui des sites, d'autres environnements spécifiques sont disponibles, tels que Ruby On Rails, ou encore CakePHP (portage de RoR en PHP). Nous aurons l'occasion prochainement de vous parler de ces frameworks.



Vista et .Net 3

2^e partie

Peut-on encore utiliser Win32 ?

Changements et innovations

Avec la sortie imminente de Windows Vista et du framework .Net 3, nous continuons à explorer les nouvelles fonctions de ces deux ensembles.

Pour le développeur, les possibilités graphiques de Windows Presentation Foundation sont immenses, notamment couplées avec les fonctions graphiques de Vista. Nous lèverons aussi un voile sur ce qui vous attend courant 2007 autour d'ADO.Net avec le futur framework ADO.Net Entity.

Pour le court et moyen terme, le développeur s'interrogera : comment migrer / porter son existant sous Vista ? Son application actuelle fonctionne-t-elle sous Vista ? Quid du code .Net 2 sous .Net 3 ? À l'heure où nous écrivons, toutes les réponses ne sont pas connues, gageons que Microsoft en saura plus dans quelques semaines.

Pour une application 100 % code managé, le passage à .Net 3 ne posera guère de problème, mis à part pour les éventuels appels natifs, aussi bien sous Windows XP que sous Vista. Avec .Net 3, on se rend indépendant du système. Cependant, les nouveaux mécanismes de sécurité et de privilèges (en particulier UAC) peuvent surprendre le développeur. Il ne faut pas oublier qu'ils n'existent pas sous XP. Il faut

donc procéder à des tests et auditer le code pouvant être affecté. Pour ce qui est des dll et API, le passage de XP à Vista posera peu de problèmes (hormis pour le 64 bits dans certains scénarios). La compatibilité ascendante existe (à quelques détails près, voir article sur la Win32 dans ce numéro). Par contre, une application native Vista ne fonctionnera pas sous XP. Pour les applications web et notamment les sites, IE 7 se révélera plus rigoureux dans les standards, le parsing. Il faut prévoir des tests approfondis et une mise à jour du site pour être compatible Vista / IE 7.

Tout ne fonctionnera donc pas nécessairement. Le développeur devra réaliser des tests, des audits de code. Malheureusement, à l'heure actuelle, il n'y a pas de procédures de tests de code Vista sous Visual Studio 2005. Peut-être dans Orcas, prévu courant 2007, mais dont la première bêta devrait voir le jour en janvier ?

Nous pourrions dans les prochains mois faire un premier bilan sur la transition vers Vista des applications. ■ François Tonic



"Pour être prêt sur Vista, il faut être prêt sur .net"

C'est la mobilisation générale chez Microsoft pour accompagner les développeurs dans la

migration vers Vista. Laurent Ellerbach, responsable du Marketing des outils de développement et des relations développeurs, explique: " Nous traitons du sujet Vista depuis un an et demi. Le thème des DevDays de l'année dernière était **Développez aujourd'hui pour être prêt pour Vista, demain** " La campagne d'information tout azimut inclut un " Centre de Développement " autour de Vista sur MSDN, qui représente environ 20% du trafic de MSDN France, (autant que VB), des évène-

ments et des webcasts spécifiques, des efforts de communication à l'intention des communautés.

Vista : une migration laborieuse ? Microsoft aide-t-il les développeurs ?

" Pour des applications bien écrites, il n'y a pas de problème, affirme Laurent Ellerbach. Pour être prêt sur Vista, il faut être prêt sur .net. Quand j'étais étudiant, en 1995, j'ai écrit une application sous Windows 95, avec C++, elle marche toujours sur Vista. Si on fait attention aux API, à la sécurité, aux privilèges -en les réduisant au maximum, on n'a pas de problèmes. Plus on est proche des drivers et des noyaux, plus c'est délicat.

Avec Vista, on programme un nouveau type d'interface homme-machine, on propose de nouveaux outils pour relier les interfaces graphiques avec le code. "

Chez Microsoft France, Laurent Ellerbach dirige désormais le marketing de la division développement. Il avait déjà la responsabilité des relations développeurs et enseignement. La division, qui comptait une dizaine de personnes à ses débuts, en compte désormais 45. La division a aussi en charge de présenter tous les produits à venir : " Technology Live ", Longhorn Server, etc. Laurent fêtera dans quelques mois ses dix ans de Microsoft: il y est rentré en 1997, pour y effectuer son stage de fin d'études!

Windows Vista : la compatibilité Win32 en question

A cette question, Microsoft a répondu "nous encourageons les développeurs à écrire du code managé pour le Framework .NET ". Néanmoins, pour les projets à court terme il reste envisageable d'écrire du code avec les API Win32. Malheureusement, certaines applications ne peuvent s'écrire en code managé, et d'autre part certaines fonctionnalités de Vista ne fonctionnent pas en code géré.

Avec Vista, les programmeurs vont-ils tourner définitivement la page de la programmation Win32 ? Non, évidemment : et si vous avez pu lire dans une (mauvaise) presse que le noyau de Windows Vista est désormais "fondé sur .NET en code géré", c'est inexact. Le noyau a été revu, et si de nombreuses fonctionnalités annexes s'appuient il est vrai sur du code managé, certaines d'entre elles comme le TCB* (Trusted Computing Base), ou encore la prise en charge de l'annulation des E/S, pour ne citer que ces deux caractéristiques, ne sont pas intégrées au code managé.

() TCB est une des briques de "l'informatique de confiance" que vise Vista. Cette base est constituée d'un ensemble de mécanismes de protection visant à renforcer la politique de sécurité d'un système Windows (par signature d'un objet informatique (logiciel, document), et désignation d'un tiers de confiance qui vérifiera que l'objet manipulé est autorisé à être utilisé sur le système informatique local).*

Une application écrite sous Windows XP et faisant appel directement aux API win32 tournera sans problème sous Vista. Ceci dit, il est nécessaire de nuancer. D'abord Microsoft précise que l'application tournera sans problème sous Vista à condition que les appels à ces API ne soient pas exotiques (autrement dit les appels utilisés doivent avoir été documentés au préalable par Microsoft). Entre parenthèses, les applications MS-DOS continueront aussi à fonctionner sous Windows Vista sans aucune modification. Ensuite, ce n'est pas parce que .net 3 est intégré à Vista qu'un service win32 est nécessairement mappé vers .net (et inversement).

Par exemple l'API win32 DeleteFile qui permet d'effacer un fichier existant est mappable vers System.IO.Directory.Delete, System.IO.DirectoryInfo.Delete, System.IO.File.Delete, ou encore System.IO.FileInfo.Delete. Microsoft a d'ailleurs établi une carte à destination des

développeurs pour connaître d'un coup d'oeil les mappages possibles (<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndotnet/html/win32map.asp>). Maintenant, certains remarqueront que toutes les API Win32 ne connaissent pas d'équivalent sous .net. C'est le cas par exemple des API natives, mais aussi par exemple des API prenant en charge l'annulation des E/S. Ce dernier point constitue justement une des nouveautés de Vista.

Annulation des E/S

Pour mettre au point Vista, Microsoft s'est penché sur les désirs des utilisateurs. Un des points souligné par les études menées a été que l'interface homme/machine ne peut rester inactive plus de 3 secondes. Au delà de ce délai, l'utilisateur pense que l'application a planté, et bien souvent, il décidera tout bonnement de redémarrer sa machine ! Il faut dire que parfois ce comportement est justifié : les entrées / sorties en cours ne peuvent se réaliser et l'application ne peut être relancée (car le processus est devenu un processus "zombie" et ne peut disparaître de la liste des processus du système).

D'où la décision de rendre le système d'exploitation plus autonome à ce niveau. En effet, le développeur qui maîtrise bien son application ne pense pas toujours aux situations créées par un mauvais comportement, et qui sont le fait d'utilisateurs ignorant les conséquences de leurs actes. Par exemple, sous Windows XP un utilisateur pourrait avoir l'idée de charger par mégarde un fichier de plusieurs centaines de Mo... Il y a fort à parier que le développeur n'a pas prévu ce geste qui figera l'interface utilisateur et qui ne pourra être annulé (le thread courant est bloqué en attente d'une E/S). D'autres cas plus fréquents peuvent se présenter, comme l'encombrement

d'un réseau qui pourra ralentir considérablement une application jusqu'à la rendre inutilisable. Sous Vista, l'arsenal des API win32 s'est doté de CancelloEx qui vise l'annulation de toutes les opérations d'E/S en attente pour le pointeur (handle) d'un fichier spécifié. Ou encore de CancelSynchronousIo qui permet d'annuler les appels d'E/S synchrones de fichier. Ce sera le cas, par exemple, lors de l'ouverture d'un fichier (OpenFile). Pour que tout ceci soit fonctionnel, il est nécessaire de programmer une application multithread. Le premier thread sera chargé de la gestion des appels bloquants (E/S), tandis que le deuxième prendra en charge l'annulation, ainsi que par exemple l'affichage d'un indicateur de progression des traitements.

Sous XP, si l'utilisateur est bloqué (demande E/S synchrone très lente), la seule manière qu'il aura d'en sortir sera de mettre fin à l'application ; ce qui peut échouer et entraîner la naissance d'un processus zombie. Les API Win32 fournies sous XP pour résoudre ce problème sont fort limitées. Il existe bien Cancello (sous VB : [DllImport("kernel32.dll")] private static extern Boolean Cancello(IntPtr hFile);) mais cette fonction ne permet pas de stopper une opération d'E/S initiée par un autre thread...

Sous Vista, le programmeur doit d'abord se demander si l'opération d'entrées/sorties qu'il s'apprête à employer est annulable. Par exemple CopyFile ne l'est pas, tandis que CopyFileEx peut être annulée. Ensuite, il devra vérifier que l'opération d'entrées/sorties a été réalisée ou non, même si une demande d'annulation a été effectuée (car il est toujours possible que l'opération se déroule jusqu'au bout malgré la demande d'annulation).

L'exemple suivant, issu des exemples proposés par Microsoft, déclare une routine qui

accepte un délai d'expiration durant lequel il y a tentative de lecture. Si le délai a expiré, la routine tente d'annuler la lecture à l'aide de CancellableEx. Mais comme il n'y a aucune garantie que la lecture soit arrivée à terme de son travail, cet état doit être vérifiable (ERROR_OPERATION_ABORTED)

```

BOOL
DoCancelableRead(
    HANDLE hFile,
    LPVOID lpBuffer,
    DWORD nNumberOfBytesToRead,
    LPDWORD lpNumberOfBytesRead,
    LPOVERLAPPED lpOverlapped,
    DWORD dwTimeout,
    LPBOOL bCancelCalled
)
{
    BOOL result;
    DWORD waitResult;

    *bCancelCalled = FALSE;

    result = ReadFile( hFile,
                      lpBuffer,
                      nNumberOfBytesToRead,
                      lpNumberOfBytesRead,
                      lpOverlapped );

    if (!result) {
        if (GetLastError() != ERROR_IO_PENDING) {
            return result;
        }
    } else {
        return result;
    }

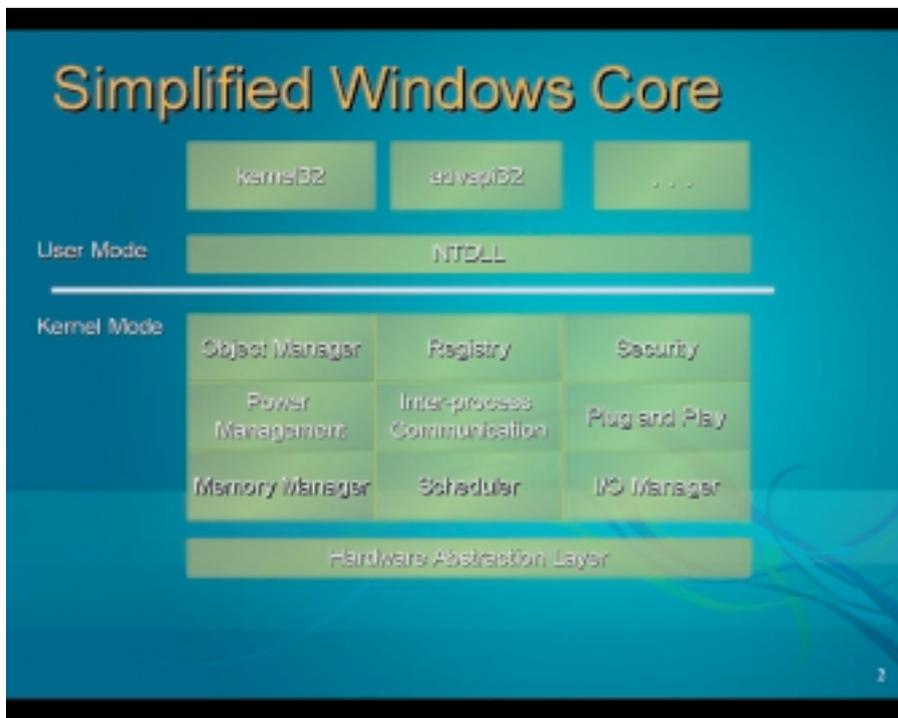
    // délai
    waitResult = WaitForSingleObject( lpOverlapped->hEvent,
                                     dwTimeout );

    if (waitResult == WAIT_TIMEOUT) {
        result = CancellableEx( hFile,
                               lpOverlapped );

        *bCancelCalled = TRUE;

        // attente de la fin de lecture
        waitResult = WaitForSingleObject( lpOverlapped->hEvent,
                                         0 );
    }
}

```



Schématiquement la structure noyau de Vista reste la même avec les API natives, en premier plan du mode utilisateur non privilégié.

```

return result;
}

```

API natives NTDLL.DLL

Vista, tout comme ses prédécesseurs, possède une bibliothèque spéciale dite d'API natives. Celles-ci réalisent des appels directs au système (affectant la mémoire / le matériel / les processus) mais officiellement ces API natives n'existent pas ou sont très peu documentées.

Du côté utilisateur, l'interface de ces API est la bibliothèque NTDLL.DLL. L'utilisateur qui fait appel à une de ces API provoquera une exception pour un passage en mode noyau en mode privilégié. Il existe ainsi quelques centaines d'API très puissantes (plus de 200) dont quelques dizaines sont mentionnées dans le kit de développement des pilotes Windows. Par exemple, la fonction Win32 de base ReadFile(...) est mappée vers la fonction native NtReadFile(...). Au final, si vous utilisez des fonctions natives vous gagnerez en performance mais aucune garantie n'est donnée par Microsoft que cette application tournera sous Vista et perdrera ensuite dans le temps.

```

NTSYSAPI NTSTATUS WINAPI NtReadFile(HANDLE FileHandle,
                                     HANDLE Event,

```

```

PIO_APC_ROUTINE ApcRoutine,
PVOID ApcContext,
PIO_STATUS_BLOCK IoStatusBlock,
PVOID Buffer,
ULONG Length,
PLARGE_INTEGER ByteOffset,
PULONG Key
);

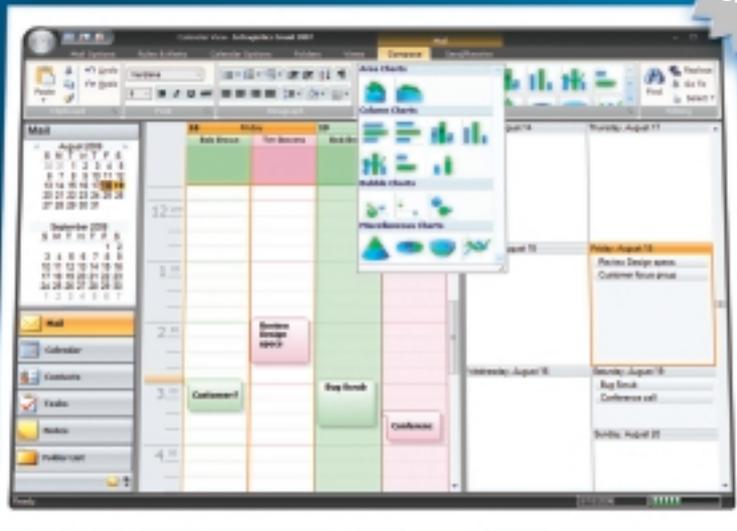
```

Le code non managé est parfois indispensable

Les programmeurs devant descendre dans les méandres du système doivent souvent faire du code non managé. Gerald Maffeo, un des développeurs responsable de l'implémentation de l'annulation des E/S dans Vista explique indirectement pourquoi : "nous avions à l'origine planifié d'inclure le code (d'annulation des E/S) dans Vista. Cependant, nous avons du changer nos plans en cours de route. La raison principale est qu'il nous a été demandé de conserver une compatibilité descendante du framework qui devait fonctionner sous Vista, mais aussi sous XP.

Or nous ne pouvions implémenter cette nouvelle technologie si, à la base, le noyau du système d'exploitation ne le supportait pas. Et il n'était pas possible d'implémenter ça sous XP/2003 sans casser l'intégration de certaines applications".

Maintenant avec
le look & feel
d'Office® 2007



Nouveau! Office® 2007 UI

Délivrez le style d'Office® UI
de demain, aujourd'hui.

NetAdvantage®

pour .NET 2006 Vol. 3

La boîte à outils idéale pour le design et le
développement de vos interfaces utilisateur.

Windows® Forms - Livrez la nouvelle interface utilisateur d'Office 2007 incluant le nouvel interface ruban, des calendriers, des galeries, des styles et plus!

ASP.NET - Appliquez facilement le style Office 2007 à tous les contrôles. Complétez l'expérience de vos utilisateurs avec les comportements AJAX intégrés ou profitez des capacités de rendering partiel grâce au nouveau composant WebAsyncRefreshPanel™ (WARP Panel™).

Charting - Actualiser l'affichage des données avec le nouveau style Office 2007 dans vos graphes. Configurez rapidement vos graphes pour modifier l'affichage des données d'une manière vivante.

Application Styling™ - Appliquez d'une manière globale le look & feel d'Office 2007, l'interface ruban y compris, visionnage des ressources, sélection de couleurs eye-dropper, une liste de ressources utilisées et bien plus encore.

Pour plus d'infos: infragistics.com/netadvantage

sales-europe@infragistics.com

 0800 667 307

Infragistics®
Powering The Presentation Layer

Copyright © 1996-2006 Infragistics, Inc. All rights reserved. Infragistics, NetAdvantage and the Infragistics logo are registered trademarks of Infragistics, Inc. Microsoft, Windows and Office are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks or registered trademarks are the property of their owners. *Microsoft® Office® 2007 is currently in beta and subject to change. Infragistics will continue to fine-tune and change Office® 2007 style points as beta builds are made available, and will finalize style points when RC builds are released.

Certaines fonctionnalités ne sont pas prises en charge par .net mais bien par les API win32 comme la génération de vidéos ou la possibilité de lire des flux vidéo. Un exemple parmi d'autres : votre application doit pouvoir détecter si elle tourne sous une machine virtuelle VMWARE ou non. En fait, il est possible de tester ceci de manière relativement simple en allant repêcher certains noms de périphériques, mais une méthode plus rigoureuse est de réaliser un test de bas niveau. Les processeurs x86 emploient des instructions d'entrées/sorties IN et OUT. Celles-ci sont des instructions privilégiées (non utilisateur). En mode non privilégié, c'est-à-dire lorsque vous exécutez votre application en dehors d'une machine virtuelle VMWARE, cet appel IN provoquera le déclenchement d'une exception (EXCEPTION_PRIV_INSTRUCTION). Par contre, si votre application tourne sous une machine virtuelle VMWARE vous allez pouvoir lire le port E/S qui vous renseignera sur la version du logiciel VMWARE présent.

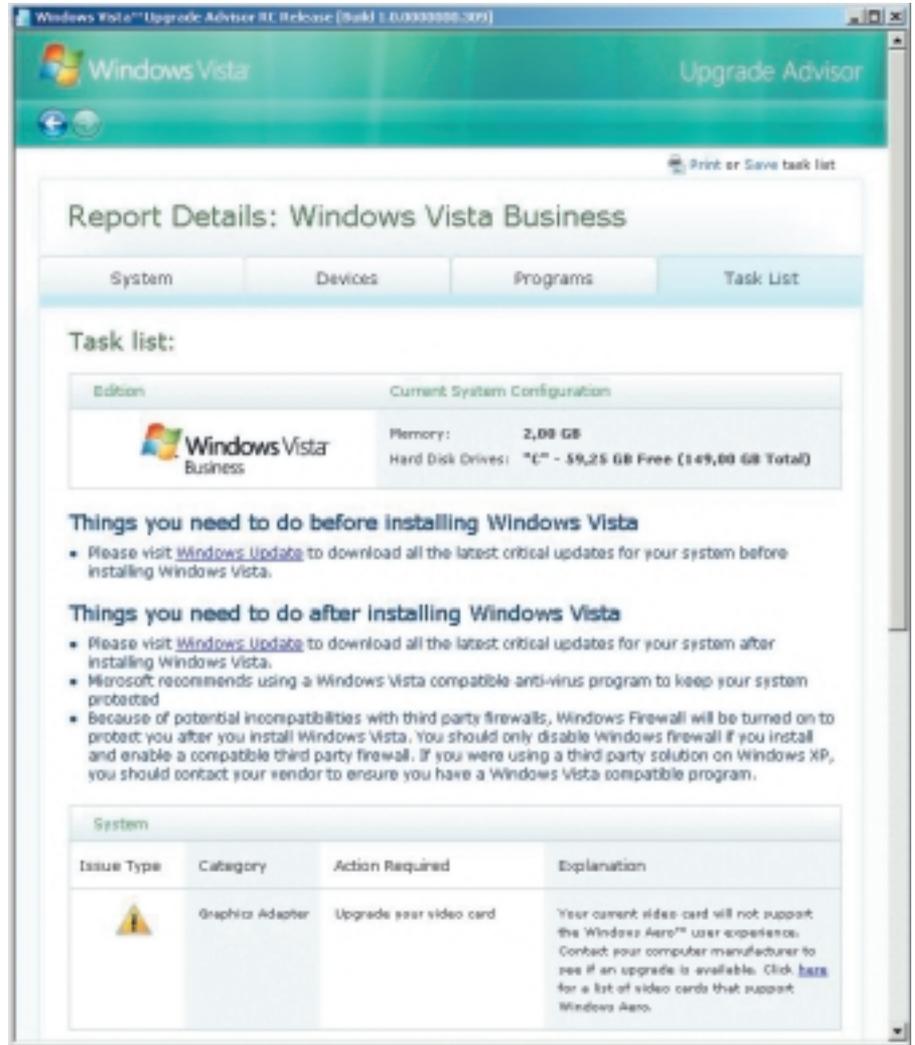
```
bool TestSiVMWare()
{
    bool rc = true;

    __try
    {
        __asm
        {
            push edx
            push ecx
            push ebx

            mov eax, 'VMXh'
            mov ebx, 0
            mov ecx, 10 // Fonction de recherche de la
            // version VMWare
            mov edx, 'VX' // numéro de port

            in eax, dx // lecture du port
            cmp ebx, 'VMXh' // test si VMWare
            setz [rc] // version

            pop ebx
            pop ecx
            pop edx
        }
    }
    __except(EXCEPTION_EXECUTE_HANDLER)
    {
        rc = false;
    }
}
```



Windows Vista Upgrade Advisor vous listera une série de tâches à effectuer pour rendre votre machine compatible Vista.

```
return rc;
}
```

Vérification du code non managé

Si vous avez décidé pour une raison x ou y de continuer à travailler avec des API Win32, natives ou non, vous devez savoir que Microsoft laisse à votre disposition un outil de diagnostic destiné aux applications en code non managé. Son nom : *Microsoft Application Verifier*. Il vous aidera à déceler des fuites mémoire ou des trous potentiels de sécurité en réalisant un monitoring de l'interaction de cette application avec l'OS : traçages des appels à la base des registres, au système de fichiers, aux API, à la pile (débordement), aux pointeurs de fichiers, etc. Concrètement, l'outil décèlera un appel non terminé à une API (Basics Verification), simulera un environnement avec l'utilisateur le moins privilégié pour

détecter des erreurs de droits (Limited User Account Predictor Verification), chargera l'application en tenant compte d'un environnement faible en ressources mémoire (Low Resource Simulation Verification).

L'outil peut s'exécuter de la ligne de commande (si vous testez une DLL vous ne pouvez le faire isolément du processus qui la sollicite). Voici comment activer les tests de base :

```
appverif /verify ApplicationAtester.exe
```

Pour sélectionner les tests dans un environnement à faibles ressources :

```
appverif /verify ApplicationAtester.exe /faults
```

L'application fonctionne en modifiant avant exécution chaque appel à des méthodes non managées. Chaque fonction est déviée avant exécution de la méthode réelle ("Function Hooking") pour pouvoir tracer l'appel.

Toutes les données créées durant l'analyse d'AppVerifier sont stockées sous le nom de fi-

chier %ALLUSERSPROFILE%\AppVerifierLogs dans un binaire. Ces logs peuvent être alors être convertis en données XML (via l'interface utilisateur ou la ligne de commande). Pour visionner ces données vous pouvez exécuter un navigateur, ou alors appliquer une transformation XSLT pour transformer le contenu brut XML. Le contenu XML peut aussi être importé dans Excel ou une base de données.

Tester votre passage de XP à Vista

Vous pouvez télécharger le logiciel Windows Vista Upgrade Advisor pour tester si votre machine est compatible avec Vista au niveau matériel et logiciel. L'application scanne le système et vous indiquera par exemple que votre carte graphique doit être mise à jour pour bénéficier de l'interface Aéro (si votre carte graphique a plus d'un an, la probabilité est grande que vous deviez en acheter une plus récente).

Si vous avez le choix, programmez en code managé

Comme nous l'avons vu, la programmation non managée faisant appel aux API Win32 comporte des inconvénients. D'abord une non

compatibilité descendante (comme l'annulation des E/S sous Vista non disponible sous XP), voire ascendante (les apis natives de NTDLL.DLL), ainsi qu'un risque sécuritaire. Rappelons en effet qu'un code C++ non managé aura un accès direct à la mémoire par le biais de pointeurs au contraire du code managé. Enfonçons le clou : C++ . Net permet d'activer en environnement runtime de débogage certains mécanismes de vérification, comme le "stack checks" et l'"uninitialized variable checks", ainsi que le repérage d'un débordement de tampon lors de l'exécution (l'option de compilation "/GS"). Ce dernier mécanisme est très intéressant, car il n'affecte que de très peu les performances (de l'ordre de 2%). Il ne corrige pas le code mais dresse une barrière difficilement franchissable.

Il est très tentant pour un développeur d'étendre les fonctionnalités de . Net en utilisant le mécanisme P/Invoke (*). L'API Win32 vous permet souvent de gagner du temps mais le revers de la médaille est le risque d'une instabilité (gestion mémoire), et une moindre portabilité (au travers de Mono en vue d'un portage Linux par exemple). D'un autre côté, on peut difficilement imaginer que

Microsoft abandonne un jour le support des fonctions API Win32 officiellement documentées. Par conséquent, les jours de la programmation API Win32 ne sont pas encore comptés même sous Vista.

(* Platform Invoke : passerelle permettant d'accéder aux fonctions, structures et rappels dans les DLL non gérées. Nous vous invitons à visiter un site entier consacré à ce sujet : <http://pinvoke.net/>.

Exemple de code P/Invoke C# :

```
[DllImport("Wininet.dll", SetLastError=true, CharSet=CharSet.Auto)]
static extern bool FtpCommand(
    IntPtr hConnect,
    bool fExpectResponse,
    [MarshalAs(UnmanagedType.U4)]
    int dwFlags,
    string lpszCommand,
    IntPtr dwContext,
    ref IntPtr phFtpCommand);
```

■ **Xavier Leclercq**

Xavier.Leclercq@programmez.com

Les outils Microsoft de compatibilité des applications

Avec l'arrivée de Vista, l'une des préoccupations majeure du développeur et de l'utilisateur sera de déterminer si son ancienne application est compatible. Comme par le passé, Microsoft a anticipé cette interrogation en préparant une nouvelle version 5.0 de son ACT (Application Compatibility Toolkit). La version 4.1 ayant été précédemment utilisée pour tester le passage au service pack 2 de XP. Cette cinquième mouture réalise l'inventaire des applications Windows présentes pour en tester la compatibilité avec Vista. Il vérifie également le fonctionnement des applications pour le comparer aux exigences d'un système fonctionnant sur une base de restriction des privilèges. Un des problèmes les plus fréquents est le blocage d'une application interagissant avec Internet Explorer, celui-ci évoluant dorénavant dans un cadre d'exécution protégé. Cet outil examine aussi les interfaces DCOM ainsi que les paramètres du pare-feu. Les nouvelles fonctionnalités d'analyse de cette version 5 permettent d'effectuer la création de rap-



ports et de les exporter. Pour tester au préalable vos applications sous Vista, nous recommandons aux développeurs la solution de virtualisation de microsoft, Virtual PC, car VPC Express est inclus dans Windows Vista sous forme d'option de licence. Virtual PC 2004 est depuis peu libre au téléchargement. Pour créer des applications compatibles, le développeur se référera au SDK dédié à Windows Vista (SDKv6). Microsoft recommande l'utilisation de PREfast, un outil d'analyse statique, capable de détecter certaines erreurs dans le code source et plus particulièrement dans le code des pilotes de périphériques (comme des pointeurs nuls ou des variables non utilisées).

Enfin, Microsoft a rédigé un livre blanc (Developer Best Practices and Guidelines for Applications in a Least Privileged Environment) recommandant des instructions et des méthodes pour permettre aux développeurs d'intégrer leurs applications dans la nouvelle infrastructure de sécurité de User Account Protection (UAP) sur Microsoft Windows Vista.

ACT : http://www.microsoft.com/downloads/details.aspx?FamilyId=4005DA7_9-933A-4CC8-BF86-FE2E28B792FD&displaylang=en

Virtual PC : <http://www.microsoft.com/windows/virtualpc/default.msp>

SDK Windows Vista : <http://windowssdk.msdn.microsoft.com/en-us/library/default.aspx>

PREfast : http://www.microsoft.com/whdc/DevTools/tools/PREfast_steps.msp

Developer Best Practices and Guidelines for Applications : <http://msdn.microsoft.com/windowsvista/reference/default.aspx?pull=/library/en-us/dnlong/html/AccProtVista.asp>

Le Framework ADO.Net Entity

Alors que .Net 3.0 est sur le point de sortir, plusieurs équipes travaillent d'ores et déjà sur la version suivante, appelée pour l'instant, vNext. Les travaux portent sur de nombreux points : création d'applications pour Vista, pour Office 2007 et d'une manière générale faire tout ce qu'il faut pour faciliter la vie des développeurs. A ce titre, deux nouvelles technologies risquent de particulièrement retenir l'attention : le Framework ADO.Net Entity et LINQ (pour Language-Integrated Query).

Développer une application accédant à des données demande aujourd'hui la mise en œuvre de compétences multiples et d'outils variés n'offrant pas nécessairement un bon niveau d'intégration. Un développeur doit à la fois maîtriser :

- un langage de développement (C# ou VB).
- une couche d'accès aux données (ADO.Net ou les Enterprise Libraries).
- le SQL (voire plusieurs variantes si l'application accède à des données dans des bases d'éditeurs différents).
- Plusieurs outils (Visual Studio, SQL Server Management Studio, outils fournis avec les SGDB d'autres éditeurs).

Et bien que Visual Studio permette d'intégrer des outils d'accès aux données pour essayer d'unifier l'expérience utilisateur, des actions en théorie aussi simples que le débogage simultané de la partie code et de la partie SQL peuvent s'avérer pénible à l'usage. Or, en théorie, pour accéder aux données relatives à un client, le code suivant devrait être tout ce que le développeur devrait avoir à écrire :

```
using(MaConnection maBase = new MaConnection()) {
    // récupération de tous les clients membres du club
    // et livrant à l'adresse de facturation
    var clients = from client in maBase.Client
                  where client.EstMembre == true
                  && client Adresse.Type == 1
                  select client;

    foreach(Client client in clients) {
        // Traitement métier...
    }
}
```

Ce type de code expose ce qu'une solution de correspondance entre objets et base de données (ou mapping OR) doit pouvoir permettre. Le développeur peut alors se concentrer sur son code sans s'occuper de la plomberie sous-jacente, quitte à pouvoir ensuite affiner lorsque des problèmes d'optimisation se posent. La suite de cet article va explorer comment, avec le Framework Entity et LINQ, ce type de code peut être mis en œuvre.

Le Framework ADO.Net Entity

Pour illustrer tous les exemples à venir, cet article va s'appuyer sur une version simplifiée d'une base de données destinée à recueillir les commandes de clients d'un site Web. Cette base contient donc une liste de clients. Chaque client peut être membre du club des utilisateurs du site (avec des réductions à la clé par exemple). Chaque client peut disposer

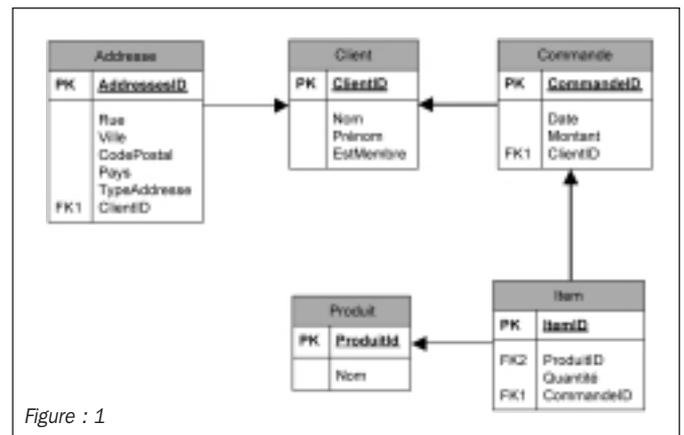


Figure : 1

de plusieurs adresses (facturation, livraison, etc). Et il a, bien entendu, la possibilité de passer plusieurs commandes :

Dans un développement classique, pour récupérer la liste de tous les clients ayant passé une commande depuis le début 2006 et qui sont membres du club, il faudrait commencer par écrire une requête SQL ressemblant à :

```
SELECT c.Nom, c.Prénom
FROM Client c
INNER JOIN Commande co ON co.ClientID = c.ClientID
WHERE c.EstMembre = 1 AND co.Date >= '2006-01-01'
```

Cette requête est encore assez simple : pour peu qu'il eût fallu aussi filtrer sur la nature des produits contenus dans la commande et le type d'adresse, plusieurs jointures auraient alors été nécessaires. Or la valeur ajoutée de ce type de requête est proche de zéro...

Le modèle de données Entity

Pour répondre à cette problématique, le Framework Entity introduit des éléments de modélisation permettant d'avoir une vue de plus haut niveau sur les structures de données. Ces éléments sont regroupés dans le modèle de données Entity (ou EDM pour Entity Data Model). En particulier, les concepts suivants sont définis :

- Entity, qui modélise une structure de données complexe. Les Entities peuvent être regroupées par Entity-Sets
 - Relationship, qui modèle une relation entre des Entities. Les Relationships peuvent être regroupées par Relationship-Sets.
- Ils permettent à l'EDM d'exprimer des notions comme :
- l'héritage (un ClientMembre est un type de Client).

- la gestion de type complexe (un Client a une adresse composée d'une rue, une ville, un code postal).

En pratique, le schéma logique peut alors être exprimé de la façon suivante (en se concentrant sur un client et ses commandes) : (Figure 2)

La requête SQL précédente peut alors être maintenant reformulée :

```
SELECT cm.Nom, cm.Prénom
FROM ClientMembre cm
WHERE cm.Commande.Date >= '2006-01-01'
```

Ici, la jointure est prise en charge par le Framework Entity grâce aux informations contenues dans le schéma EDM. A ce jour, la correspondance entre ce schéma et la structure de base de données sous-jacente se fait via l'utilisation de fichier XML. Des outils intégrés à Visual Studio seront fournis avec la version finalisée.

Le fournisseur de correspondance

Afin de pouvoir mettre en œuvre ces éléments dans une application .Net, un nouveau fournisseur de données a été créé. Ce dernier prend en paramètre les informations de correspondance et s'occupe alors de toute la plomberie. L'accès aux données depuis l'application peut alors se coder comme ceci :

```
using(MapConnection con = new
    MapConnection(Settings.Default.SiteWeb)) {
    con.Open();

    MapCommand cmd = con.CreateCommand();
    cmd.CommandText =
        "SELECT cm.Nom, cm.Prénom " +
        "FROM SiteWeb.SiteWebDB.ClientMembre AS cm " +
        "WHERE cm.Commande.Date >= @date";
    cmd.Parameters.AddWithValue("@date", DateDébut);

    DbDataReader r = cmd.ExecuteReader();
    while(r.Read()) {
        Console.WriteLine("{0}\t{1}", r["Prénom"], r["Nom"]);
    }
}
```

Ce code ne se distingue en rien de ce qu'aujourd'hui pourrait écrire un développeur utilisant ADO.Net 2.0. Mais à ce stade, l'accès aux données reste tabulaire. Une étape est encore nécessaire pour pouvoir récupérer des objets et travailler dessus.

La couche objet

Le Framework ADO.Net Entity est destiné à être accompagné de nombreux outils : création de schémas EDM, correspondance avec des structures de bases, etc.

Un de ces outils permet de générer des classes partielles à partir d'un schéma EDM : ainsi, le développeur peut ajouter du code métier (test de validité, calculs, autres) dans des fichiers distincts, sans craindre de perdre son code s'il génère de nouveau ces classes. Et elles permettent de réécrire le code précédent ainsi :

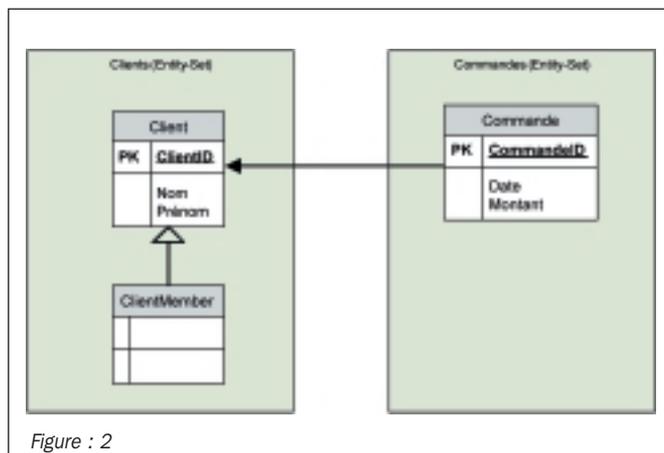


Figure : 2

```
using(MapConnection con = new
    MapConnection(Settings.Default.SiteWeb)) {
    con.Open();

   ObjectContext ctx = new ObjectContext(con);

    Query<ClientMembre> membres = ctx.GetQuery<ClientMembre>{
        "SELECT cm.Nom, cm.Prénom " +
        "FROM SiteWeb.SiteWebDB.ClientMembre AS cm " +
        "WHERE cm.Commande.Date >= @date",
        new QueryParameter("date", DateDébut);

    foreach(ClientMembre membre in membres) {
        Console.WriteLine("{0}\t{1}", membre.Prénom, membre.Nom);
    }
}
```

Les objets ObjectContext et Query remplacent ici les objets Command et DataReader utilisés habituellement. Mais l'outil de génération de classes partielles fournit aussi une classe particulière, spécifique au schéma EDM utilisé, qui masque une partie de cette plomberie. Le code devient alors :

```
using(SiteWebDB sw = new
    SiteWebDB(Settings.Default.SiteWeb)) {

    Query<ClientMembre> membres = sw.GetQuery<ClientMembre>{
        "SELECT cm.Nom, cm.Prénom " +
        "FROM SiteWeb.SiteWebDB.ClientMembre AS cm " +
        "WHERE cm.Commande.Date >= @date",
        new QueryParameter("date", DateDébut);

    foreach(ClientMembre membre in membres) {
        Console.WriteLine("{0}\t{1}", membre.Prénom, membre.Nom);
    }
}
```

Mais un point gênant reste : la nécessité de manipuler un langage de type SQL pour récupérer des objets .Net. Pour arriver au niveau du premier exemple de code de l'article, une dernière étape est nécessaire : l'utilisation de LINQ.

LINQ et Entities

Présentation de LINQ

LINQ était à l'origine un projet d'extension du langage C# devant permettre l'écriture de requêtes ensemblistes sans recourir à un autre langage. L'intégration de cette extension dans les différents langages du Framework permettra alors à ces langages de procéder à des sélections, des tris, des filtres sur différents types de données assimilables à une collection.

L'exemple de code suivant permet de récupérer dans un tableau d'entiers tous les éléments de valeur inférieure ou égale à 5, triés dans l'ordre croissant :

```
int[] numbers = new int[] {5, 7, 1, 4, 9, 3, 2, 6, 8};

var smallnumbers = from n in numbers
    where n <= 5
    orderby n
    select n;

foreach(var n in smallnumbers) {
    Console.WriteLine(n);
}
```

Comme indiqué précédemment, LINQ permet l'utilisation de requêtes ensemblistes sur de nombreux types de données (objets, XML). Il est aussi possible de requêter sur des DataSets en mémoire, sur des sources de données et sur des schémas EDM (exposés via le Framework Entity).

Utilisation conjointe avec SQL

Il s'agit probablement du mode d'accès aux données le plus simple. En effet, comme dans le cas du Framework Entity, un outil permet la génération de classes partielles à partir des structures de la base de données (tables, procédures stockées, vues et fonctions utilisateur).

Une des limitations majeures de ce mode d'utilisation est que, du fait de l'absence d'un schéma conceptuel intermédiaire, les classes générées reflètent directement la structure de la base : il n'est pas possible d'agglomérer deux tables dans une seule classe ou de générer des classes spécialisant une classe parent sur la base du contenu d'une colonne (comme la classe ClientMembre pour la classe Client dans les exemples précédents). Ce type de solution peut néanmoins parfaitement répondre aux besoins les plus simples.

Utilisation conjointe avec des DataSets

Dans certains cas, il peut être pertinent de cacher des données en mémoire, sans maintenir de connexion avec la base. Le Framework ADO.Net permet de satisfaire ce type de scénario depuis sa première version avec le type DataSet. Pour rappel, un DataSet permet la récupération en mémoire de données issues d'une base de données, tout en conservant une structure relationnelle (tables, relations, clés, index). Un langage de requête intégré permet même l'exécution de recherches simples.

Avec LINQ, il devient alors possible de travailler sur les données du DataSet :

```
DataSet ds = new DataSet();
ChargeClients(ds); // récupère les données depuis la base

DataTable clients = ds.Tables["Client"];

var query = from c in clients.ToQueryable()
    where c.Field<bool>("EstMembre") == true
    select new { ClientID = c.Field<Guid>("ClientID"),
        Nom = c.Field<string>("Nom"),
        Prénom = c.Field<string>("Prénom") };

foreach(var client in query) {
    Console.WriteLine("{0}\t{1}\t{2}",
        client.ClientID, client.Nom, client.Prénom);
}
```

Et chose intéressante : les requêtes sur jointure deviennent enfin possibles. Ainsi, pour rechercher tous les membres ayant passé une commande depuis début 2006, il faut écrire :

```
DataSet ds = new DataSet();
ChargeClients(ds); // récupère les données depuis la base

DataTable clients = ds.Tables["Client"];
DataTable commandes = ds.Tables["Commande"];

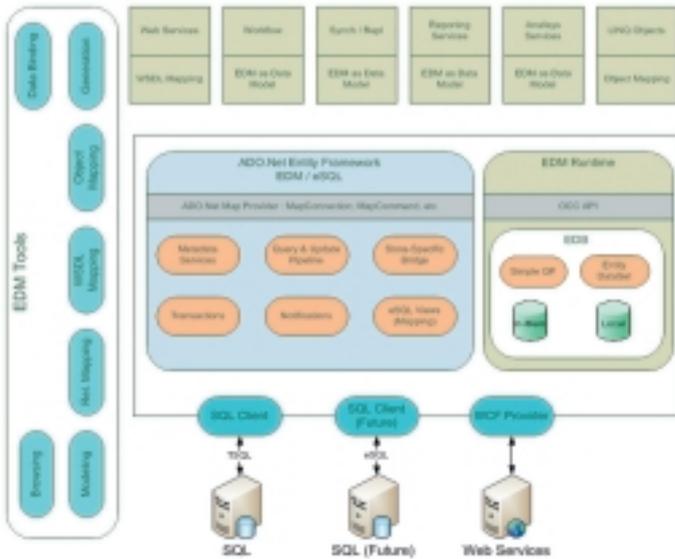
var query = from c in clients.ToQueryable()
    join co in commandes.ToQueryable()
    on co.Field<Guid>("ClientID")
    equals c.Field<Guid>("ClientID")
    where c.Field<bool>("EstMembre") == true
    && co.Field<DateTime>("Date") >= new DateTime(2006, 1, 1);
    select new { ClientID = c.Field<Guid>("ClientID"),
        Nom = c.Field<string>("Nom"),
        Prénom = c.Field<string>("Prénom") };

foreach(var client in query) {
    Console.WriteLine("{0}\t{1}\t{2}",
        client.ClientID, client.Nom, client.Prénom);
}
```

Mais ce type de code présente un inconvénient majeur : l'accès aux colonnes par leur nom rend le code moins lisible qu'il ne pourrait l'être et interdit toute vérification à la compilation. L'utilisation d'un DataSet typé permet de résoudre ce problème :

```
ClientDataSet ds = new ClientDataSet();
ChargeClients(ds); // récupère les données depuis la base

var query = from c in ds.Client
    join co in ds.Commande on co.ClientID equals c.ClientID
    where c.EstMembre == true
    && co.Date >= new DateTime(2006, 1, 1);
    select new { ClientID,
        Nom,
        Prénom };
}
```



```
foreach(var client in query) {
    Console.WriteLine("{0}\t{1}\t{2}",
        client.ClientID, client.Nom, client.Prénom);
}
```

Ce qui présente le double avantage d'être lisible et vérifiable à la compilation :

Utilisation conjointe avec le Framework Entity

Enfin, en utilisant LINQ au-dessus des classes générées à partir du schéma EDM, le code d'accès aux données peut enfin s'écrire de la manière suivante :

```
using(SiteWebDB sw = new
    SiteWebDB(Settings.Default.SiteWeb)) {
    var membres = from c in sw.ClientMembre
        // orderby c.Nom, c.Prénom, c.Commande.Date
        where c.Commande.Date > DateDébut
        select c;

    foreach(ClientMembre membre in membres) {
        Console.WriteLine("{0}\t{1}", membre.Prénom, membre.Nom);
    }
}
```

L'utilisation conjointe de LINQ et d'Entity a permis au final d'exprimer dans un seul langage, au sein d'un seul environnement, toutes les instructions nécessaires pour accéder en base aux données voulues. En résumé, ici :

- Entity a la charge de fournir le support pour un schéma conceptuel de données permettant une modélisation de plus haut niveau des structures en base. Il a aussi la charge de l'accès aux données (sélection, mise à jour, suppression). Enfin, il fournit des classes partielles permettant la manipulation des structures exposées par le schéma EDM dans un mode objet plutôt que tabulaire.
- LINQ a la charge de permettre l'expression de requêtes ensemblistes dans le langage utilisé par le développeur et non dans une variante de SQL. Il transforme ensuite ces requêtes en requêtes sur le schéma EDM sans que le développeur n'ait à intervenir sur ce processus.

Conclusion

De nombreuses questions restent néanmoins en suspens et demandent encore un peu de recul :

- LINQ to SQL et LINQ to Entities ne se recouvrent-ils pas ? L'absence d'une technique d'indirection (comme le schéma EDM d'Entity) semble limiter l'utilisation de LINQ to SQL aux cas les plus simples.
- Quel sera la place du langage SQL et des procédures stockées à l'avenir ? La tentation de ne plus utiliser que ce type de technologies peut être grande. Mais la problématique de l'optimisation de l'accès aux données, de la manipulation d'ensembles conséquents fait qu'un équilibre va devoir être trouvé entre la simplicité d'écriture et les besoins de performances.

Depuis le mois d'août, une version préliminaire est disponible sur MSDN pour tous ceux souhaitant commencer à tester ces technologies. Elle nécessite une version de Visual Studio, la version préliminaire de mai et une version de SQL Server (Express ou 2005) pour tester la persistance en base.

- La CTP de mai est téléchargeable sur <http://www.microsoft.com/downloads/details.aspx?familyid=1e902c21-340c-4d13-9f04-70eb5e3dceea&displaylang=en>
- La CTP d'août est téléchargeable sur <http://www.microsoft.com/downloads/details.aspx?familyid=b68f6f53-ec87-4122-b1c8-ee24a043bf72&displaylang=en>

Toutes les suggestions et remarques sont bien entendu les bienvenues :

■ Patrice Manac'h
Microsoft France



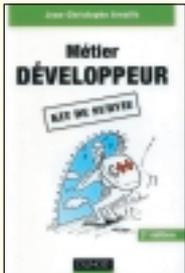
Offre Limitée **Abonnement 2 ans**

Abonnez-vous pour 24 mois et recevez le livre « Métier Développeur - Kit de survie », de J-Chr. Arnulfo (Dunod, 272 pages). Offre réservée à la France métropolitaine.

90€ au lieu de : • le magazine seul, au numéro : 130,90 €
• Métier Développeur : 19,90 €

Economisez 60 Euros

Abonnez-vous sur www.programmez.com ou coupon page 59



Introduction à Windows Presentation Foundation

2e partie

Windows Vista, le nouveau système d'exploitation de Microsoft, et son lot de nouveautés ont été largement présentés dans la presse. Tout le monde a déjà vu la nouvelle interface utilisateur faisant la part belle à la transparence et aux effets graphiques et aussi entendu parler des nouvelles fonctions liées à la sécurité apportées par Vista. Mais pour le développeur, qu'y a-t-il de nouveau ?

Pour illustrer d'autres mécanismes de WPF ainsi que l'utilisation d'Expression Interactive Designer, nous allons maintenant écrire une application simple mais complète : un lecteur de fichiers audio : il va afficher les albums présents dans le dossier "Musique" de l'utilisateur courant et permettre d'en jouer les différentes pistes. Voici le résultat fini (OK, il n'est pas particulièrement beau mais c'est à moi, et non à WPF, qu'on doit ce résultat graphiquement discutable) :

Je m'étais fixé une contrainte pour la réalisation de cette petite application : gérer complètement la partie présentation sous Microsoft Expression Interactive Designer sans écrire la moindre ligne de code. En effet, un des objectifs de WPF est d'améliorer l'ergonomie des applications en confiant cette tâche à des Designers et non plus des développeurs C#.

Il n'est cependant pas réellement possible de développer une application entière sans écrire la moindre ligne de code : dans le cas de ce lecteur audio, il faut, par exemple, parcourir une arborescence de répertoires pour dresser la liste des albums présents. Cependant, il est possible d'isoler complètement cette partie fonctionnelle de la partie graphique. Le projet WpfMediaPlayer doit donc être constitué de deux parties distinctes :

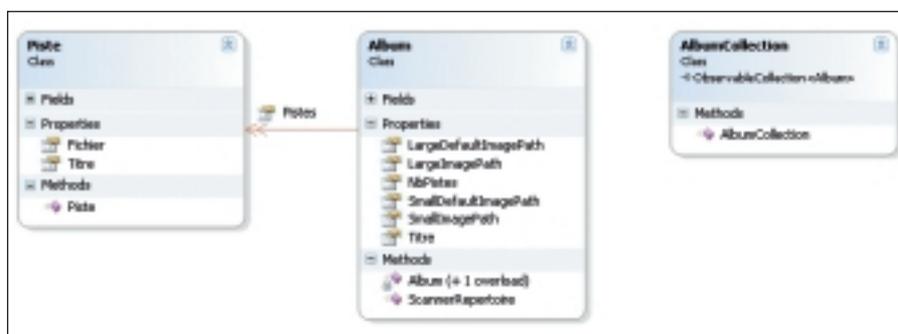
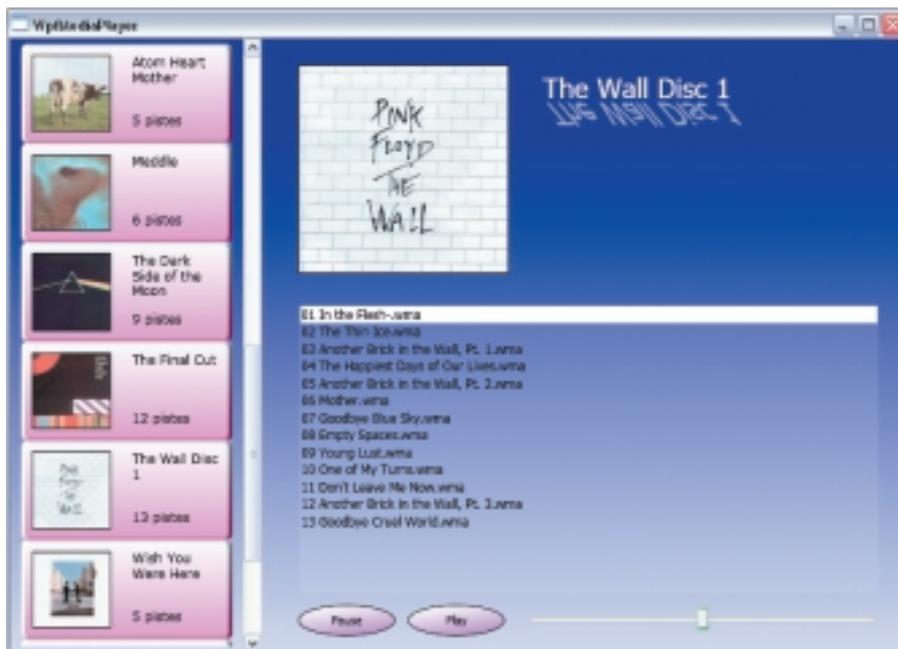
- Un assembly .NET (MediaEntites.dll) chargé de la récupération des données par parcours du disque et de leur présentation sous forme de classes manipulables sous EID.
- L'exécutable chargé de la présentation et entièrement développé en XAML.

Assembly " fonctionnel " MediaEntites.dll

Avant de passer à la description de l'exécutable et d'Expression Interactive Designer, jetons un coup d'œil à l'assembly.

Voici les trois classes exportées par cet assembly : (schéma ci-contre)

Dans la mesure où les objets représentant



l'arborescence des morceaux musicaux ne seront pas modifiés, une fois le parcours initial de cette arborescence effectué, les trois classes de l'assembly n'ont pas de contrainte d'implémentation particulière. Si ces objets pouvaient être modifiés à l'exécution (soit par appels de méthodes à la suite d'actions de l'utilisateur, soit par surveillance du répertoire de musique afin de détecter d'éventuelles modifications sur les fichiers audio), il serait nécessaire d'implémenter certaines inter-

faces .NET pour bénéficier complètement des mécanismes de Databinding de WPF.

Par exemple, une classe dont les propriétés peuvent être modifiées gagnera à implémenter l'interface *INotifyPropertyChanged* : lorsque WPF lie un composant graphique à un objet implémentant cette interface, il enregistre un délégué sur l'événement *INotifyPropertyChanged.PropertyChanged*, ce qui permet de mettre à jour l'interface utilisateur en cas de modification de la propriété,

sans avoir à écrire de code supplémentaire. L'implémentation de la propriété Titre de la classe Piste pourrait être implémentée de la manière suivante :

```
public class Piste : INotifyPropertyChanged {
    private string _titre;

    public string Titre {
        get {
            return _titre;
        }
        set {
            if (_titre != value) {
                _titre = value;
                NotifyPropertyChanged("Titre");
            }
        }
    }

    public event PropertyChangedEventHandler
    PropertyChanged;

    protected void NotifyPropertyChanged
    (string property) {
        if (PropertyChanged != null) {
            PropertyChanged(this, new PropertyChange
            dEventArgs(property));
        }
    }
}
```

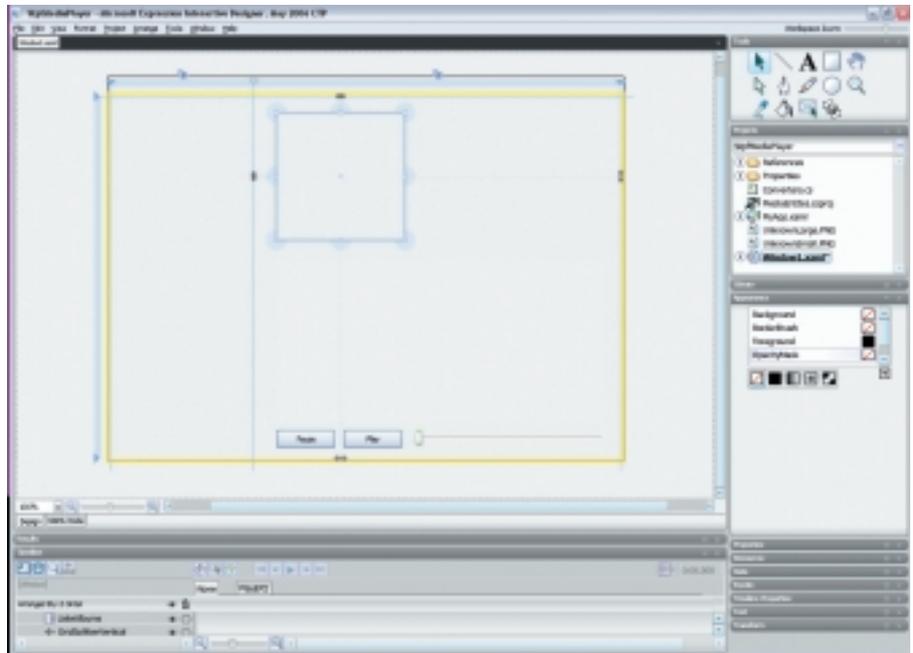
De la même manière, les listes doivent aussi publier leurs modifications pour une prise en compte automatique par le Databinding WPF ; une manière simple d'implémenter ceci est d'hériter de la classe ObservableCollection :

```
public class Album {
    ...
    public ObservableCollection<Piste>Pistes { ... }
    ...
}
public class AlbumCollection : Observable
Collection<Album>{
    ...
}
```

En effet, ObservableCollection implémente à la fois INotifyPropertyChanged et INotifyCollectionChanged.

Le fonctionnement de MediaEntites.dll est très simple :

- Le constructeur de la classe AlbumCollection récupère le dossier "Ma Musique" de l'utilisateur courant à l'aide de l'API .NET



Environment.GetFolderPath (Environment.SpecialFolder.MyMusic)et effectue un parcours récursif de celui-ci.

- Quand un répertoire contient un fichier desktop.ini avec une ligne " FolderType= MusicAlbum ", ce répertoire est considéré comme un album et l'ensemble des fichiers wma est ajouté à la collection des pistes.
- Les couvertures des albums sont des fichiers image nommés AlbumArt_ZZZ_small.jpg et AlbumArt_ZZZ_large.jpg.

Cette localisation de tout le code "fonctionnel" dans une DLL séparée de l'interface utilisateur n'est pas une exclusivité WPF : c'est une bonne pratique de programmation que de séparer le code lié à l'interface utilisateur du code métier, celui-ci est alors réutilisable dans différentes applications. En revanche, dans des applications plus complexes que ce petit exemple, ce code métier sera plus riche et offrira bien d'autres services qu'un simple accès en lecture à des données.

Réalisation de l'interface utilisateur

Afin d'exploiter complètement les possibilités offertes par WPF, il est nécessaire de faire intervenir des designers ou graphistes lors de l'élaboration de l'interface utilisateur des applications. Ceux-ci vont non seulement définir l'aspect général de l'application et ses principaux modes d'interaction, mais aussi participer au développement de celle-ci. En fonction des projets et des compétences disponibles, le schéma de développement d'une

application WPF variera, mais un certain nombre de grands principes resteront constants, en particulier l'utilisation de Microsoft Expression Interactive Designer comme outil principal de développement des interfaces WPF.

EID est un outil complet qui permet non seulement la manipulation du XAML définissant l'interface utilisateur d'une application, mais aussi celle du code associé à cette interface. EID permet ainsi l'exécution et la mise au point des applications WPF, accélérant ainsi le cycle de développement : si le développeur prépare convenablement les composants exécutant le code métier, le designer élaborant une interface utilisateur pourra s'appuyer sur ces composants.

L'interface utilisateur de EID est constituée d'une zone de travail présentant l'interface en cours d'élaboration entourée d'une multitude de boîtes à outils permettant de voir l'arborescence des composants, de sélectionner les

A retenir

Visual Studio .NET 2005 et Expression Interactive Designer travaillent sur les mêmes fichiers mais ne les verrouillent pas. Par ailleurs, ils détectent les modifications sur les fichiers ouverts lorsqu'ils passent en premier plan et en permettent le rechargement : il est donc possible (et fréquent) d'ouvrir un même projet dans les deux outils et de passer de l'un à l'autre.

composants à installer, d'en modifier les propriétés, etc. EID lui-même est développé avec WPF et exploite entièrement les possibilités de redimensionnement et zoom offertes par cette technologie : ainsi, deux curseurs dans l'interface de EID permettent non seulement de modifier de manière continue le facteur de zoom de la zone de travail mais aussi de le régler, toujours de manière continue.

La conception d'une fenêtre doit commencer par une réflexion quant aux options de redimensionnement de l'application. En effet, non seulement WPF est basé sur des graphiques vectoriels permettant de faire varier de manière continue la taille des objets, mais la plupart des composants WPF savent adapter leur taille et celle de leurs sous-composants en fonction de l'espace disponible. Le composant idéal pour gérer le redimensionnement d'une fenêtre est la Grid, qui s'apparente aux tables utilisées dans les pages Web pour gérer l'organisation de celles-ci.

La figure ci-dessous présente la fenêtre principale du WpfMediaPlayer : le composant principal est constitué d'une Grid, coupée en deux colonnes qui vont se redimensionner proportionnellement à la fenêtre globale (les cadenas sont ouverts).

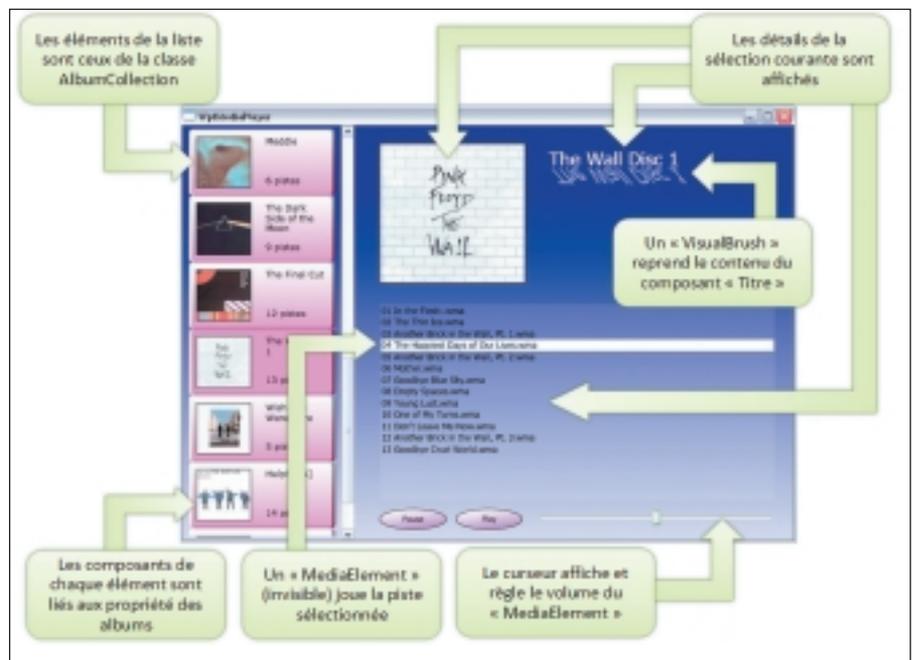
L'objet sélectionné est une image qui présentera la couverture de l'album courant. Les petits maillons reliés à cet objet par des lignes qui partent de chacun de ses côtés indiquent la manière dont l'image va se redimensionner lorsque la grille le sera aussi :

- Les distances aux bords gauche et haut seront fixes (maillons fermés),
 - Les distances aux bords droits et bas seront libres (maillons ouverts)
- L'image gardera donc une taille fixe et sera à une distance fixe du haut de la fenêtre et du bord gauche de sa colonne.

Styles et Templates

Pour le développeur, un bouton est un contrôle générant un événement " Click " et servant à déclencher une action. On a vu dans la première partie de cet article comment on pouvait composer le contenu d'un bouton pour dépasser les contraintes classiques (texte et/ou image). WPF permet d'aller plus loin en permettant par exemple d'avoir des boutons ronds : ceci se réalise facilement en XAML ou sous Interactive Designer :

- on définit dans les ressources de la fenêtre un Template centré (à l'aide d'une Grid) une ellipse et le contenu du bouton :



```
<ControlTemplate x:Key="TemplateBouton"
TargetType="{x:Type Button}">
<Grid x:Name="Grid">
<Ellipse Stroke="#FF000000" x:Name="Ellipse"
>... </Ellipse>
<ContentPresenter>...</ContentPresenter>
</ControlTemplate>
```

• et les boutons se voient appliquer ce template :

```
<Button Template="{DynamicResource TemplateBouton}" ...>
```

Le point important du template utilisé ici est l'élément ContentPresenter : un bouton sait afficher différents contenus comme on l'a vu dans la première partie de cet article : il aurait été dommage que les templates brident cette possibilité. Le ContentPresenter est fait pour cela : lorsque le template du bouton sera appliqué sur un bouton donné, le contenu de celui-ci sera inséré dans le template à la place de l'élément ContentPresenter de celui-ci. Il existe dans WPF d'autres éléments similaires au ContentPresenter : typiquement, un "ItemsPresenter" est utilisé dans tous les contrôles de type Liste pour préciser où doivent être affichés les éléments de la liste. Dans le cas d'une listbox, on peut utiliser la notion de template pour la liste elle-même, mais aussi pour chacun de ses éléments.

Le style d'un contrôle permet de modifier les propriétés de celui-ci : typiquement modifier la police ou la couleur de fond. Cependant, le style peut aller plus loin, puisqu'il peut pré-

ser le template à utiliser ! On peut donc avoir pour une même application un style "Arrondi" ou un style "Anguleux" ...

Databinding

La liste affichée dans la colonne de gauche doit afficher la liste des albums musicaux de l'utilisateur : elle va être remplie à l'aide du mécanisme de Databinding. WPF est capable d'effectuer du Databinding sur des objets .NET ou sur des sources XML. La boîte à outils "Data" permet de créer une source de données. Ensuite, le binding lui-même peut être effectué soit en modifiant le XAML manuellement, soit en déposant un objet représentant des données (la source ou un de ses composants) sur un objet graphique :

L'étape suivante est de préciser comment les items de la liste seront affichés. Ici, ce sont des objets .NET typés qui sont affichés dans la liste, des "Albums", il est donc possible de préciser leur mode d'affichage de deux manières :

- en donnant un " Template " d'affichage pour tous les objets de type " Album ". Cette option est intéressante quand des objets élémentaires sont affichés à de nombreux endroits dans une application.
- en précisant le Template pour les éléments de cette liste particulière.

Dans le cas du WpfMediaPlayer, le XAML résultant est celui-ci :

```
<ListBox ... ItemsSource="{Binding Mode=OneWay, Source={StaticResource AlbumCollectionDS}}" ItemTemplate="{DynamicResourceTem
```

```
plateAlbum}" ... />
```

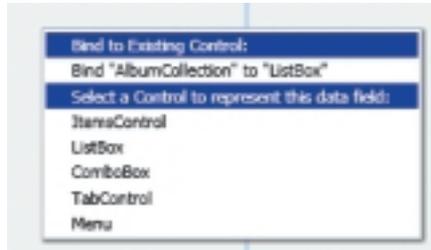
ItemsSource précise la source de données alimentant les éléments de la liste, ItemTemplate définit leur mode d'affichage. Le template de l'album est un assemblage de composants WPF qui utilise aussi le mécanisme de



Databinding pour afficher son contenu (typiquement le titre de l'album) :

```
Image :
<Image Source="{Binding SmallImagePath}"...
Titre :
<TextBlock Text="{Binding Titre}"...
```

Le databinding permet aussi une liaison dynamique entre différents composants de l'interface utilisateur, l'album sélectionné dans la liste de gauche est affiché en grand dans la partie droite de l'application. En fait, le Databinding est utilisé à de nombreux endroits dans une application WPF ; dans cet exemple, il est aussi utilisé :



- pour faire jouer automatiquement par un "MediaElement" la piste sélectionnée dans la liste des pistes de l'album courant,
- pour régler le volume de ce même MediaElement,
- pour afficher le reflet du titre de l'album sélectionné.

La figure ci-dessous met en évidence l'omniprésence du databinding dans cet exemple.

Conclusion

WPF est bien trop vaste pour être entièrement décrit dans un article comme celui-ci, cependant, ces petits exemples devraient permettre d'appréhender un peu les principaux concepts de WPF :

la richesse graphique et la séparation entre la présentation et le comportement qui s'appliquent aussi bien au niveau composant qu'au niveau application et permettent de nouveaux modes de développement des applications combinant les compétences classiques des développeurs à celles des graphiques et designers auparavant souvent cantonnés aux sites Internet.

Aujourd'hui, WPF est une technologie de type Smartclient, mais dont les applications peuvent néanmoins s'exécuter au sein d'Internet Explorer ou se télécharger via une technologie comme Click Once et s'appuyer sur des appels de services web pour réaliser leurs traitements.

Demain, avec WPF/E, Microsoft va proposer une solution permettant de toucher un encore plus grand nombre de clients (au-delà d'Internet Explorer, mais sans certaines fonctions comme la 3D, nldr).

■ **Alain Zanchetta** - *Consultant chez Microsoft France | Services*

Exécuter une application WPF dans Internet Explorer ?

WPF est une technologie de présentation pleine de possibilités et il serait dommage de la cantonner aux scénarios de type smart client avec un déploiement préalable bien que ceux-ci soient souvent la meilleure solution, tout au moins pour les applications métier. Une application WPF étant à la base une application .NET, il est bien évidemment possible de la déployer à partir d'une URL à l'aide de la technologie Click-Once ; cependant, il peut être souhaitable de voir une application s'exécuter au sein même d'Internet Explorer, typiquement dans tous les cas où on recherche une navigation fluide entre des pages HTML et des pages avec une interactivité plus riche. Deux technologies permettent de répondre à deux variations autour de ce thème :

- Si on cherche à toucher le plus grand nombre d'utilisateurs sur des plates-formes diverses, WPF/E sera la réponse adéquate. WPF/E sera disponible sur plusieurs plates-formes et pourra s'exécuter dans plusieurs navigateurs mais cette portabilité se fera au détriment de la richesse : seul un sous-ensemble (pas encore totalement défini) des fonctions de WPF/E sera disponible. Cette restriction a plusieurs justifications : non seulement la volonté de fournir un en-

semble de fonctionnalités communes à différentes plates-formes réduit nécessairement cet ensemble (or il est question de supporter WPF/E sur les plates-formes mobiles), mais puisqu'on parle de plates-formes où WPF n'est pas nécessairement présent a priori, il est souhaitable de limiter la taille du runtime qui devra être téléchargé et installé pour permettre l'exécution de WPF/E.

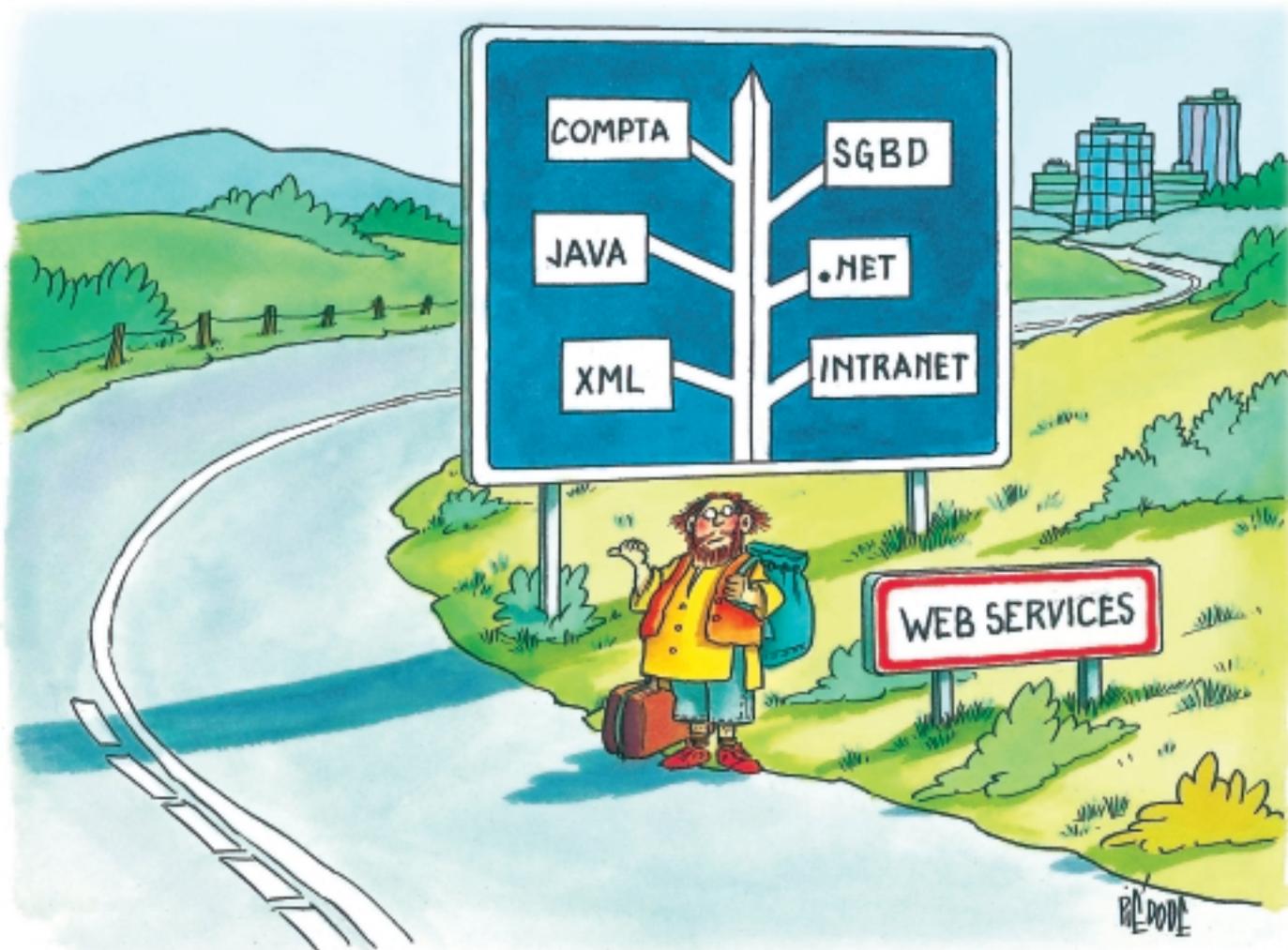
- Si on maîtrise l'environnement, ce qui est typiquement le cas des applications d'entreprises, et qu'on ne cherche à cibler que les plates-formes Windows où WPF est déjà installé, il est possible de développer des applications s'exécutant dans le navigateur et ayant accès à presque toutes les fonctionnalités de WPF et .NET. Cette possibilité existe depuis les premières versions de .NET: un " User Control " Winform peut en effet s'exécuter dans Internet Explorer. Avec WPF, il faut créer sous Visual Studio une "XAML Browser Application" et on peut alors développer une vraie application WPF dont le composant principal hérite de la classe Page. En termes d'expérience utilisateur, une telle application s'exécute au sein du navigateur et sans demander de confirmation à son lancement (tout au moins avec la configuration par

défaut d'Internet Explorer). Un tel comportement nécessite bien évidemment des précautions en termes de sécurité ; le modèle du " Code Access Security " de .NET trouve ici son plein usage : ces applications n'ont absolument pas un accès illimité aux ressources de l'ordinateur local, elles s'exécutent dans un environnement isolé et ne peuvent accéder aux fichiers locaux sans permission. Le modèle est néanmoins assez sophistiqué pour permettre à ces applications d'être utiles ! Par exemple, une telle application ne peut pas demander l'ouverture d'un fichier donné de l'ordinateur mais elle peut lire et écrire des fichiers en passant par les boîtes de dialogue standard de .NET... avec une précaution supplémentaire garantissant la protection de la confidentialité de l'utilisateur :

- Ouvrir une boîte de dialogue (classe OpenFileDialog) puis lire le fichier sélectionné par l'utilisateur à l'aide d'un objet Stream récupéré par la méthode OpenFile est accepté.
- Ouvrir la même boîte de dialogue puis essayer de récupérer le nom du fichier sélectionné par l'utilisateur à l'aide de la propriété FileName déclenche une SecurityException.

Web Services

comprendre et construire



Il y a environ 6 ans, les web services représentaient une véritable révolution, ils allaient tout changer ! Sans doute un peu trop optimistes, les éditeurs, promoteurs du concept, avaient oublié de créer les standards adéquats. Le manque de sécurité, l'interopérabilité douteuse, un choix technique pas toujours clair, ont contribué à rendre perplexes les développeurs et les entreprises. Depuis environ 2 ans, le web service rentre enfin dans les habitudes et se banalise. L'arrivée de nouveaux outils comme le bus d'entreprise (ESB), la disponibilité de standards, la vision plus pragmatique du service web, tout cela a permis de redonner envie de faire du web service. Sans parler de leur rôle dans les nouvelles formes d'intégration.

Malgré tout, si la situation s'améliore considérablement, tout n'est pas pour autant réglé. Il y a bien des standards, mais ceux-ci sont pléthoriques, se concurrencent parfois et ne sont pas toujours très... standard ! Si à la base, ils sont aujourd'hui à peu près fixés, au-dessus, le développeur devra naviguer entre deux eaux pour tenter d'utiliser un standard reconnu, ou tout le moins, utilisé.

L'interopérabilité, un des arguments du web service, constitue le maillon faible. Si un effort conséquent a été fait dans la sécurité, l'interopérabilité reste un sujet sensible, malgré l'esquisse de solutions. La multiplication des standards, qui a permis de combler les lacunes, contribue à perdre le développeur et les responsables informatique. Une fois les standards déterminés, il faut vérifier leur bonne implémentation et leur respect dans les outils de développement et les environnements de déploiement.

Pour bien comprendre et débiter dans l'univers du service web, ce dossier vous propose de commencer par rappeler ce qu'est un service web, son fonctionnement, ses standards de base. Le côté développement n'est pas oublié avec, comment concevoir rapidement un web service en Java, .Net ou PHP. La question de l'interopérabilité, comme vous le verrez, occupe une large part du dossier aussi bien sur la question des standards qu'entre web services de différentes plates-formes.

■ François Tonic

Web Services : une définition

On parle beaucoup des Web Services et de toute la galaxie de standards et de technologies qui gravitent autour. On oublie souvent au passage quelques questions essentielles : Pourquoi adopter les Web Services dans une application ? Quelle est la stratégie à suivre ? Quels sont les écueils à éviter avec SOAP ? Comment échanger de gros documents ou des données binaires ?

Au-delà de l'effet de mode, les Web Services apportent une réelle solution technique à un problème récurrent de l'informatique : faciliter l'interconnexion des applications.

Au fil du temps, 4 cas d'utilisation distincts et plus ou moins ambitieux sont apparus :

- Interconnecter des plates-formes hétérogènes ;
- Intégrer des applications existantes ;
- Client / serveur sur Internet ;
- Fournir des services à forte valeur ajoutée métier.

Interconnecter des plates-formes incompatibles

Initialement, les Web Services ont été utilisés pour interconnecter simplement des plates-formes qui auparavant communiquaient mal. La simplicité est apportée dans ce cas, plus par les outils, que par la technologie en elle-même. L'exemple typique est l'interopérabilité Java / .Net.

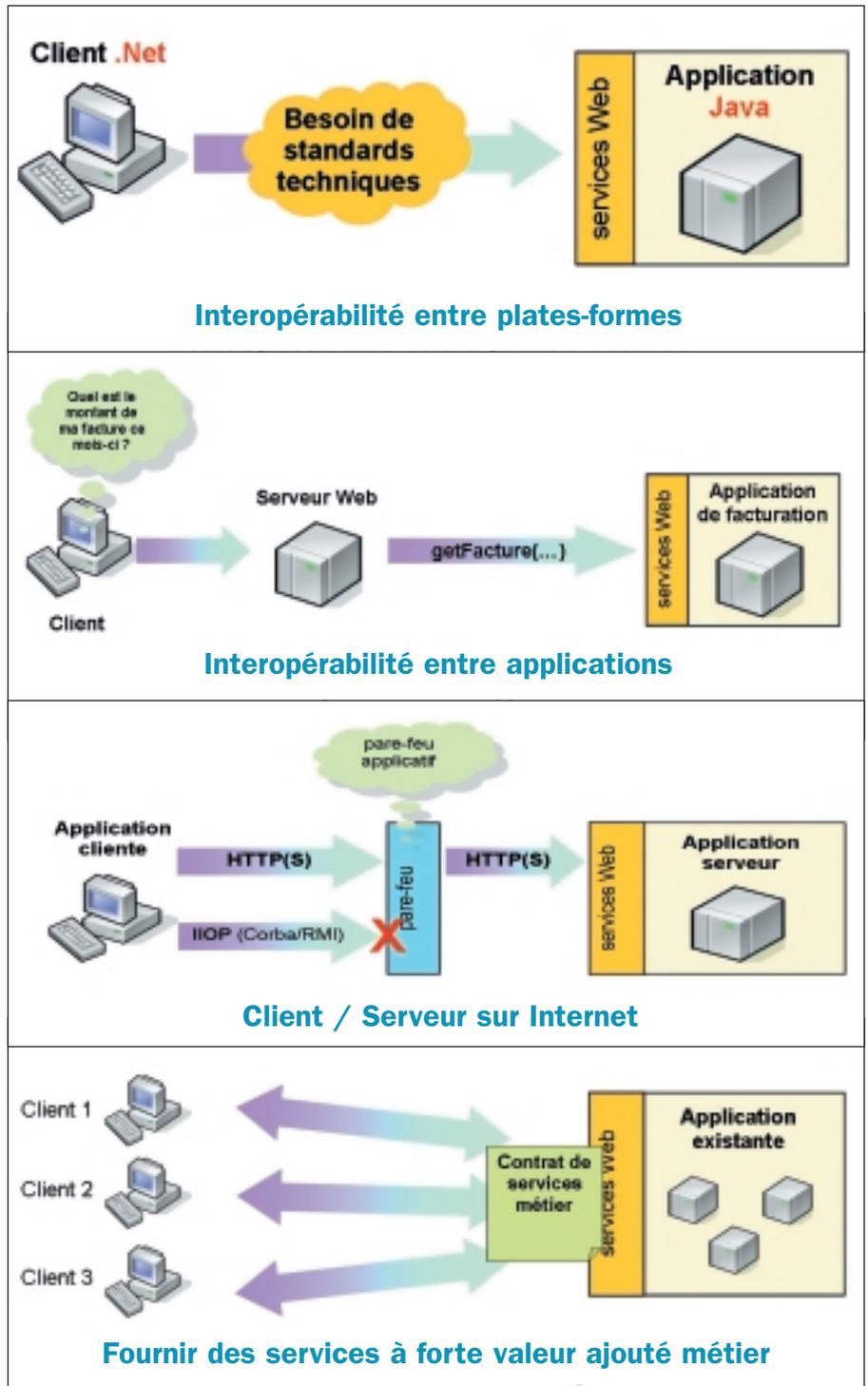
L'interopérabilité s'appuie ici sur l'utilisation des standards techniques définis par le W3C :

- SOAP (Simple Object Access Protocol ou Service Oriented Architecture Protocol) : le protocole généralement utilisé ;
- WSDL (Web Services Description Language) : langage XML utilisé pour décrire :
 1. le contrat de services : l'ensemble des opérations disponibles, ainsi que la structure des messages XML échangés (exprimée en XML Schema) ;
 2. comment, concrètement, transporter les messages XML sur 1 ou plusieurs protocoles (habituellement SOAP) ;
 3. la localisation des services.
- XML et HTTP qu'on ne présente plus.

Mais très rapidement, de nouveaux cas d'utilisation plus ambitieux sont apparus.

Intégrer des applications existantes

Les Web Services ont très vite servi à intégrer les applications existantes de l'entreprise (le-



gacy applications). Et ainsi faciliter leur réutilisation au sein du système d'information de l'entreprise et leur intégration dans des applications nouvelles (Java / .Net). L'exemple typique est la transformation d'une application existante en entrepôt de données.

L'engouement pour les Web Services, dans ce cas, est sûrement à rechercher dans :

- l'utilisation des standards du Web : HTTP et XML ;
- la démocratisation et l'abondance des serveurs Web.

Client / serveur sur Internet

Autre cas d'utilisation classique : développer une application client / serveur sur Internet. L'intérêt des Web Services réside ici dans :

- la robustesse du protocole HTTP ;
- sa capacité à passer plus facilement les pare-feu qu'un autre protocole.

Fournir des services à forte valeur ajoutée métier

Enfin, dernier cas d'utilisation : la fourniture de services à forte valeur ajoutée métier, construits en fédérant les applications existantes de l'entreprise. On sort ici du cadre des Web Services pour rentrer dans celui des architectures SOA (Service Oriented Architecture). Les Web Services fournissent alors la tuyauterie nécessaire pour interconnecter les applications. Les Web Services sont en cela la pierre angulaire technique des architectures SOA.

Quelle est la stratégie à suivre ?

Phase 1 : Définition d'un contrat de services métier

Dans cette 1re phase, on ne parle pas technique. On se focalise sur le besoin du client qu'on formalise sous la forme d'un contrat de services métier.

Un service métier est un service qui a du sens pour un expert du métier ou pour le client qui va l'utiliser.

Phase 2 : Identification des Web Services → contrat (technique) de Web Services

Cette phase consiste à :

- identifier les Web Services à partir du contrat de services métier ;
- décrire chaque service métier comme l'assemblage d'un ou plusieurs Web Services.

On commence dans cette phase à s'intéresser aux problèmes techniques, notamment aux problèmes de performance : temps de ré-

ponse, latence, bande passante, consommation mémoire, ...

Pensez : forte granularité / faible couplage

Définissez des Web Services à forte granularité. N'oubliez pas en effet, que l'invocation d'un Web Service coûte cher.

Prenons l'exemple d'une application qui désire récupérer les informations d'une personne. Vous avez la possibilité de définir :

1. Soit plusieurs services : getNom, getPrénom, getAdresse ;
2. Soit un seul : getPersonne → Personne.

La deuxième approche est la meilleure, car :

- Elle minimise le nombre de Web Services appelés ;
- Elle réduit le couplage entre l'application cliente et le serveur.

Préférez les Web Services indépendants du contexte client

Oubliez donc les sessions utilisateur (état du client conservé par le serveur entre deux appels de Web Services). Le but est de supporter plus facilement la montée en charge du nombre de clients (scalability).

Phase 3a : Identifier les Web Services asynchrones

L'objectif de cette phase est de se prémunir contre des blocages et des problèmes de performance éventuels.

Candidats possibles :

- Web Service dont le temps de réponse n'est pas borné dans le temps. Ce sont typiquement les Web Services qui s'appuient eux-mêmes sur d'autres Web Services ;
- Web Services dont le traitement est long.

Certaines plates-formes de Web Services prennent en charge la publication ou la consommation de Web Services asynchrones.

Si ce n'est pas votre cas, vous devrez probablement publier deux opérations dans le document WSDL par Web Service asynchrone :

- Une première pour envoyer la requête et obtenir en retour un identifiant (le job id) ;
- Une deuxième pour interroger régulièrement le serveur (polling) avec cet identifiant.

Prévoir aussi une opération (commune à tous les Web Services asynchrones) pour annuler un Web Service lancé ou un paramètre supplémentaire lors de l'appel, pour régler un timeout.

Phase 3b : Identifier les Web Services qui échangent de gros documents ou des données binaires

SOAP est un protocole destiné à transporter des messages XML, c'est-à-dire des messages texte. Il est donc inadapté au transport de données binaires.

De plus, certains Web Services, qui échangent de gros volumes de données, peuvent être à l'origine de problèmes de consommation mémoire, de bande passante et donc de performance.

L'objectif de cette phase est d'identifier les Web Services qui vont recourir à un mécanisme de pièces jointes (attachment). Voir ci-dessous la partie consacrée aux mécanismes de pièces jointes.

Pour le problème de bande passante, la solution consiste à compresser les pièces jointes ou à recourir à la compression HTTP.

Phase 4 : Ecriture du contrat

Il est rare d'écrire directement le document WSDL car son écriture est complexe. Généralement, le contrat est décrit dans un langage de programmation. Et, c'est la plateforme de Web Services qui se charge de générer (automatiquement) le document WSDL. En Java (JAX-WS 2.0) et en .Net, la plateforme s'appuie sur des annotations qui viennent compléter le code.

Exemple en Java en utilisant JAX-WS 2.0 :

```
@WebService(targetNamespace="http://obje
tdirect.com/facturation")
public class Facturation {

    @WebMethod
    public Facture getFacture(String nume
ro) { ... }
}
```

Les annotations @WebService et @WebMethod marquent respectivement les classes et les méthodes à publier.

Intéressons-nous maintenant à l'écriture du contrat.

Afin de maximiser l'interopérabilité du Web Service, il est important de ne pas lier sa description :

- Au langage de programmation ;
- A son implémentation.

Ne pas se lier au langage de programmation

Limitez-vous aux types suivants :

- Types simples : String, Date, int, float, ...
- Langages non objet : structure de données ;
- Langages objet : classes avec propriétés (= JavaBeans en Java) ;

Les savoirs clés de l'informatique professionnelle !



En cadeau
un Hub USB offert* !

pour l'achat de 2 ouvrages
de la collection **Info Pro** !

Rendez-vous
chez votre libraire**

du 4 au 18 novembre 2006 !



* Dans la limite des stocks disponibles

** Liste des librairies participant à l'opération disponible sur www.dunod.com

- Tableaux d'éléments ;
 - Les collections génériques de .Net ou de Java 5, c'est-à-dire les collections typées.
- Ces contraintes s'appliquent aussi au type des éléments des tableaux, des collections génériques, des propriétés, ...
- D'une manière générale, évitez :
- Les collections d'objets non typées ;
 - La surcharge de méthodes ;
 - D'utiliser la classe Object de .Net ou de Java.

Ne pas se lier à l'implémentation du Web Service

Utilisez des interfaces ou des classes abstraites pures pour séparer clairement la définition du contrat de son implémentation. Proscrire la pratique courante qui consiste à transformer n'importe quelle classe en Web Service. Pratique d'autant plus courante en Java ou en .Net où il suffit de décorer la classe par des annotations. Une classe publiable devrait être une classe qui a été pensée et conçue pour être publiée.

Quels sont les écueils à éviter avec SOAP ?

Si le S de SOAP signifie " simple ", malheureusement SOAP n'a rien de simple et est souvent à l'origine de nombreux problèmes d'interopérabilité. Nous allons en détailler deux.

Problème de version de SOAP

Il existe 2 versions de SOAP : SOAP 1.1 et SOAP 1.2 incompatibles entre elles.

.Net et la plupart des bibliothèques de Web Services Java supportent les 2 versions. Par contre, si vous êtes amenés à travailler sur d'autres plates-formes ou avec des bibliothèques plus anciennes, il se peut que vous soyez obligé de publier vos Web Services en SOAP 1.1.

Une autre solution consiste à supporter, dans le document WSDL, les 2 versions du protocole SOAP.

Problème de modèle de messages

SOAP est utilisé pour transporter les messages XML entre 2 applications. La spécification SOAP prévoit deux modèles de messages:

- Les messages de type RPC (Remote Procedure Call) ;
- Les messages de type Document.

Messages de type RPC

Messages XML destinés à représenter, indépendamment du langage de programmation, l'invocation d'un service, ainsi que son résultat éventuel. La structure générale de la requête et de la réponse est imposée par la spécification. Cette dernière aborde aussi les problèmes d'encodage des paramètres, notamment des tableaux et des graphes d'objets : RPC / encoded. Ce modèle de messages est le plus complexe des deux, mais aussi le plus contraignant.

Messages de type Document

La spécification SOAP n'impose, dans ce cas, aucune contrainte sur la structure de ces messages. Le sens des données XML véhiculées est laissé à l'appréciation des applications participant à l'échange. Ce modèle de messages offre plus de liberté, mais peut être à l'origine de problèmes d'interopérabilité.

Notez bien que dans les deux cas, la structure des messages XML échangés est complètement décrite par le document WSDL. Dans le 1er cas, le serveur est obligé de respecter certaines règles. Dans le 2e cas, il peut décrire n'importe quelle structure XML.

Le modèle de messages de type RPC est tombé en désuétude :

- Dans SOAP 1.2, seul le support du modèle Document est obligatoire ;
- On constate une évolution similaire en Java avec la dernière API : JAX-WS 2.0 ;
- Quant à .Net, il préconise depuis le début l'utilisation des messages de type Document.

Gare donc aux problèmes d'interopérabilité entre les anciennes applications qui s'appuient sur le modèle RPC et certains nouveaux outils qui ne supportent que le modèle Document.

Pour invoquer un Web Service, les plates-formes s'appuient donc aujourd'hui sur le modèle Document. Mais, comme la structure des messages XML est libre, comment par exemple repérer dans le document XML le nom du service invoqué ?

En fait, pour résoudre ce problème, les plates-formes utilisent généralement le modèle Document/literal wrapped. Ce modèle impose quelques contraintes. Notamment le fait que la balise racine du message XML transporté corresponde au nom de l'opération invoquée.

Comment échanger de gros documents ou des données binaires ?

L'échange de gros documents ou de données

binaires requiert l'utilisation d'un mécanisme de pièces jointes (attachment).

Les anciens standards

- DIME : mécanisme proposé par .Net. Requiert l'installation de WSE 2.0 (Web Services Extension) ;
- SOAP With Attachments (W3C / SOAP 1.1) : mécanisme supporté par la plupart des autres plates-formes (Java notamment).

Le nouveau standard : XOP / MTOM (W3C / SOAP 1.2)

Pourquoi un nouveau standard ?

- Rester compatible avec les autres standards (ex : WS-Security)
- Ce n'était pas le cas de DIME ou de SOAP with Attachments.

XOP/MTOM est en passe de devenir LE standard. Il est supporté à la fois par .Net (requiert l'installation de WSE 3.0) et par les nouvelles plates-formes Java.

Conclusion

Il y a encore quantité de choses à dire sur les Web Services. Nous n'avons pas parlé, par exemple, de la sécurisation d'un Web Service. Dans quels cas peut-on s'appuyer sur HTTPS, quand doit-on adopter les standards comme WS-Security ? Il faudrait aussi parler du virage pris par Java avec la nouvelle API JAX-WS 2.0. J'ai cité quelques exemples de problèmes d'interopérabilité. La vision donnée ici reste limitée. J'espère cependant vous avoir donné quelques conseils utiles.

En voici un dernier : avant de commencer un développement, évaluez le risque technique induit par l'utilisation des Web Services. Faites des tests d'interopérabilité, des maquettes, des prototypes pour valider la chaîne entière. Ces tests préalables sont indispensables, notamment si vous envisagez de ne pas vous limiter à SOAP/WSDL.

■ Philippe Guédez

Expert XML/J2EE

Objet Direct, filiale de Homsys Group

A propos de Homsys Group :

Créé en 1991, Homsys Group est spécialisé autour de la Business Intelligence (Homsys) et les technologies Objet et Internet (Objet Direct). Homsys Group est implanté à Paris, Marseille, Lyon, Toulouse, Bordeaux, Grenoble et Rennes.

www.homsysgroup.com

www.objetdirect.com

Mes premiers pas en services web

Les services Web répondent à un besoin d'interopérabilité. Ils permettent de faire appel à des fonctionnalités distantes, appartenant à une application tierce. Ainsi votre application peut valider des adresses mail, faire des recherches sur Google ou récupérer les cours de la bourse en temps réel sans que vous ayez quoi que ce soit à développer. On dit alors que l'on "consomme" un service Web.

Inversement, votre propre application peut offrir ses propres services sur le réseau pour permettre à n'importe qui de s'interfacer avec les données que vous souhaitez partager. Les services Web vont donc vous permettre de gagner en indépendance et en productivité. Attention cependant, ils ne sont pas adaptés à tous les types de développement.

Protocoles et technologies

Bien que dans la pratique utiliser des services Web soit relativement aisé, il faut encore s'y retrouver dans la jungle de solutions, protocoles et technologies existantes. Pour commencer nous allons aborder le concept du service Web.

Couplage lâche et couplage fort

Pour communiquer entre deux applications on peut utiliser deux méthodes :

- Interactions natives, on parle de couplage fort.
- Interactions via des protocoles ou format d'échanges, on parle de couplage lâche.

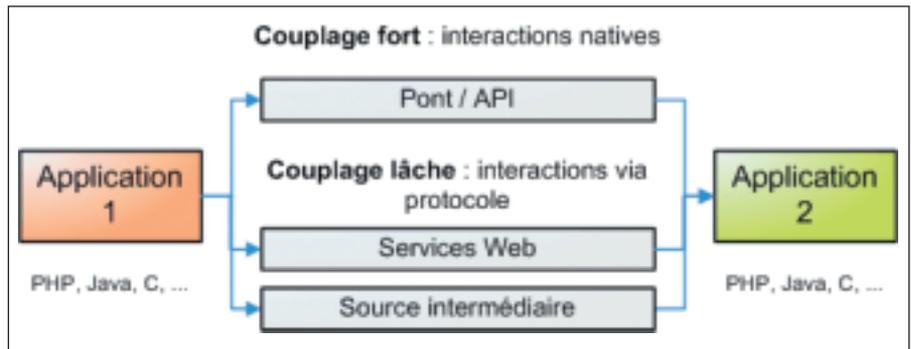
Le couplage fort permet des échanges de données plus efficaces, des structures de données plus complexes, mais engendre plus de dépendances.

Le couplage lâche consiste à utiliser un moyen (protocole) d'échange des données tiers (en passant par un fichier texte, par un flux XML, par une base de données, par des services Web).

Protocoles d'invocations de services à distance

Les services Web ont le vent en poupe, mais le concept d'invocation de services à distance n'est pas nouveau. Nous pouvons par exemple citer RMI (Remote Method Invocation), DCOM et Corba (Common Object Request Broker Architecture).

Ces solutions s'appuient sur des protocoles, des langages et des interfaces très riches et jalousement propriétaires. D'autres solutions



Couplage entre deux applications

ont émergé depuis : les services Web qui se basent sur des formats et des protocoles standardisés et massivement utilisés : XML et HTTP. De l'universalité de cette infrastructure résulte de facto une interopérabilité qui fait toujours défaut aux méthodes précitées.

Pour travailler, nous pouvons utiliser différentes approches :

REST (REpresentational State Transfer)

Cette architecture est de plus en plus utilisée pour la réalisation de services Web destinés à la communication entre machines. Le gros avantage de REST est sa simplicité. Il s'agit en effet uniquement de l'utilisation du protocole HTTP à son plein potentiel.

XML-RPC (XML - Remote Procedure Call)

XML-RPC est une spécification simple qui permet à des processus s'exécutant dans des environnements différents de faire des appels de méthodes à travers un réseau. XML-RPC est l'ancêtre de SOAP.

SOAP (Simple Object Access Protocol)

SOAP est un protocole de RPC orienté objet, bâti sur XML. Il s'agit du type de service Web le plus courant. Il bénéficie de plus d'une spécification complète et détaillée dans les normes éditées par le W3C. C'est généralement le protocole à utiliser pour des applications professionnelles.

SOAP : Les différents protocoles en jeu

Description du service

Les services distants sont décrits dans un fi-

Avantages désavantages d'une architecture SOA

- + indépendance, compatibilité
- rapidité moindre, dépendance au réseau (si sur machines différentes)

chier WSDL (Web Services Description Language).

Échanges entre applications

Simple Object Access Protocol (SOAP) permet la transmission de messages entre objets distants, ce qui veut dire qu'il autorise un objet à invoquer des méthodes d'objets physiquement situés sur une autre machine.

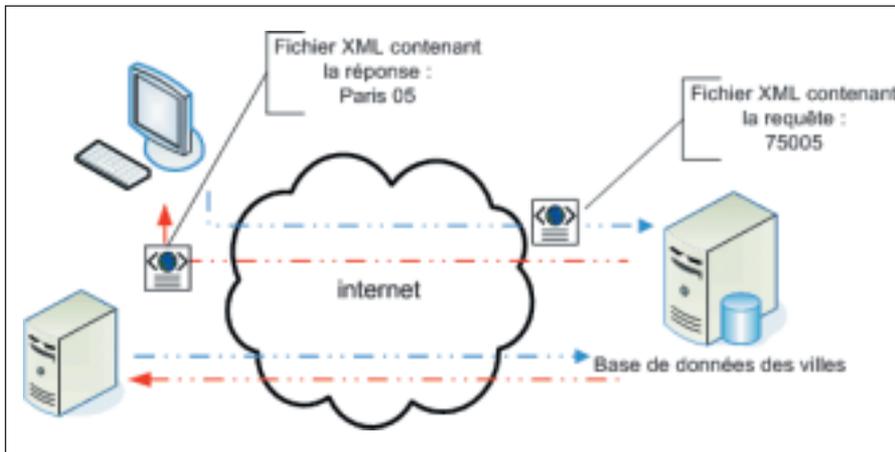
Le transfert se fait le plus souvent à l'aide du protocole HTTP, mais peut également se faire par d'autres protocoles, comme SMTP.

Les services Web en pratique

Description du service Web

Nous allons mettre en place un service Web extrêmement simple. Plusieurs machines vont entrer en jeu : un ou plusieurs clients qui vont demander de l'information et un serveur qui distribuera l'information. Au niveau langage de programmation, il n'est pas obligatoire d'utiliser le même, c'est la force du service Web. Commençons par le serveur, celui qui offre le

Web Services



Exemple de transfert d'informations

service. Ce service que nous allons développer permettra d'associer à un code postal, le nom de la ville correspondante.

Offrir un service Web

Pour offrir un service avec SOAP il nous faut :

- Un fichier décrivant le service (on parle de fichier wsdl),
- Un programme réalisant le service.

Description d'un service Web avec WSDL

Pour utiliser un service Web avec SOAP il faut connaître la structure des méthodes proposées par le service, la liste des paramètres et leur type. Ces informations sont contenues dans le fichier wsdl mais, généralement, la description des API existe sous la forme d'une documentation plus claire que le fichier wsdl. En général, on n'écrit pas son fichier WSDL à la main, on le génère avec un outil. Pour PHP on peut utiliser le Zend Studio ou Dia (avec l'extension uml2php5). En Java, la génération du wsdl, basée sur une classe d'implémentation pourra être assurée soit par des bibliothèques open-source comme Axis ou Xfire, soit plus récemment par le serveur d'application Java EE 5.

Remarque : Vous n'avez pas besoin de comprendre comment est architecturé le fichier wsdl, sauf si vous voulez faire des actions complexes. Si vous n'êtes pas concerné passez directement à l'implémentation du service Web.

Nous allons utiliser un fichier de description très sommaire.

```
<?xml version='1.0' encoding='UTF-8' ?>
<definitions
  targetNamespace='http://example.org/Ville'
  xmlns:tns='http://example.org/Ville'
  xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
  xmlns:xsd='http://www.w3.org/2001/XMLSchema'
  chema'
```

```
  xmlns:soapenc='http://schemas.xmlsoap.org/soap/encoding/'
  xmlns:wSDL='http://schemas.xmlsoap.org/wsdl/'
  xmlns='http://schemas.xmlsoap.org/wsdl/'>
```

- Les sections de <message> définissent les listes de messages qui peuvent être échangés avec le service. On y retrouve les différents types de requêtes ainsi que les différentes réponses.

```
<message name='getvilleRequest'>
  <part name='symbol' type='xsd:string' />
</message>
<message name='getvilleResponse'>
  <part name='Result' type='xsd:string' />
</message>
```

- La section <portType> liste les messages autorisés en entrée et en sortie pour chaque opération accessible sur votre service Web.

```
<portType name='VillePortType'>
  <operation name='getville'>
    <input message='tns:getvilleRequest' />
    <output message='tns:getvilleResponse' />
  </operation>
</portType>
```

- Enfin, les sections <binding> et <service> vont gérer les détails d'implémentation (protocoles ou couches transport à utiliser, adresse, port, etc.).

```
<binding name='VilleBinding' type='tns:VillePortType'>
  <soap:binding style='rpc'
    transport='http://schemas.xmlsoap.org/soap/http' />
  <operation name='getville'>
```

```
<soap:operation soapAction='getville' />
  <input>
    <soap:body use='encoded' namespace='urn:villeservice'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding' />
  </input>
  <output>
    <soap:body use='encoded' namespace='urn:villeservice'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding' />
  </output>
</operation>
</binding>
```

```
<service name='Ville_Service'>
  <documentation>WSDL associant un CP a une ville</documentation>
  <port name='VillePort' binding='VilleBinding'>
    <soap:address location='http://www.oofrance.com/soap/server.php' />
  </port>
</service>
</definitions>
```

La déclaration WSDL peut être vue comme un contrat entre le client qui consomme le service Web et celui qui l'offre. On déclare ce qui est échangé, où, et comment. C'est sur ces spécifications que les deux intervenants vont pouvoir se baser pour interpréter les échanges.

Construire le service Web avec Java

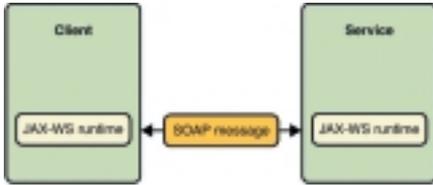
De nombreuses JSRs (Java Specification Request) et APIs visent l'implémentation des spécifications Web Service dans le langage Java. Dans le cadre de notre article nous nous intéressons à deux d'entre elles, intégrées dans Java EE 5 :

- JSR 181 (Web Services Metadata for the Java™ Platform) définissant un format d'annotation facilitant la définition de web service
- JSR 224 avec JAX-WS (API basée sur étendant l'API JAX-RPC)

Le but de ces deux spécifications est de rendre les Web services simples à développer et faciles à déployer, elles définissent :

- un modèle simplifié qui facilite et accélère le développement
- une syntaxe d'annotation et un outillage associé pour une programmation plus concise et plus facile

- un standard pour construire et déployer les Web Services



On distingue trois modèles de développement de Web Services, ces nouvelles fonctionnalités et support du langage fournissent un support pour les trois :

Partir du WSDL

Les conteneurs et IDEs (supportant nos JSRs) fournissent des outils de génération des interfaces de service ainsi que des classes associées aux définitions du schéma et aux éléments du message défini par le WSDL. Le développeur n'a alors qu'à se concentrer sur l'implémentation du service.

Faire converger Java et WSDL

Il est parfois nécessaire de mapper une implémentation existante avec une interface de contrat définie par un WSDL tiers. Le processeur JSR 181 peut alors fournir des informations quand les annotations présentes dans l'implémentation ne correspondent pas au contrat défini par le WSDL.

Partir du Java

Ce modèle de développement requis par la spécification permet au développeur de créer une classe d'implémentation et d'exposer cette classe en tant que service web par l'utilisation de simple annotation.

Mise en place avec Glassfish:

La solution que nous avons choisi de mettre en place est l'utilisation du serveur d'application Glassfish dont nous pouvons voir la console d'administration.

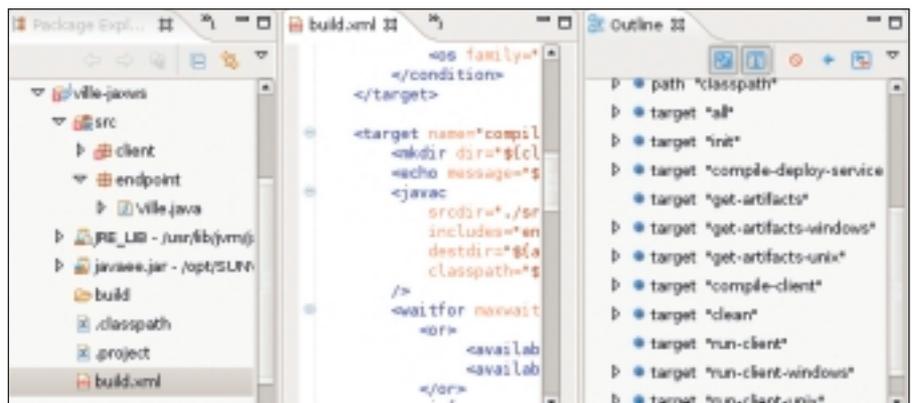
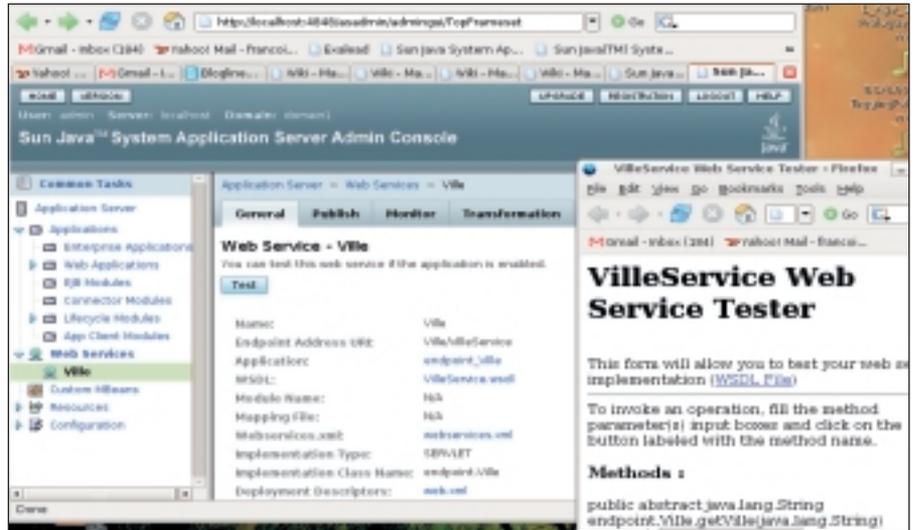
L'implémentation du web service

L'exemple couvert est un service Web très simple qui a pour fonctionnalité d'associer le nom d'une ville à son code postal. Nous allons donc faire appel à la fonction `getville()` en lui indiquant un code postal.

Le code implémentant le service web peut se résumer à cette simple classe :

```

package endpoint;
import javax.jws.WebService;
@WebService
public class Ville {
    public String getVille(String codePostal) {
  
```



```

        return VilleLocator.getVille(codePostal)
    }
}
  
```

```

    /javaws/villeWS",
    serviceName = "VilleWS"
)
  
```

Les artefacts et éléments de configuration restants, nécessaires au déploiement de ce service peuvent être tous automatiquement générés par le serveur d'application : il suffit pour cela de compiler ce fichier `Ville` dans le répertoire d'auto déploiement du serveur. Vous trouverez dans le code source fourni un script Ant facilitant la compilation, le déploiement et le test de ce service web comme de son client. On dispose alors de la description normalisée du web service :

- le fichier WSDL `VilleService.wsdl` disponible sous http://nom-du_serveur:port_du_serveur/Ville/VilleService?wsdl
- le schéma associé `VilleService_schema1.xsd`

Le WSDL peut être personnalisé à travers l'utilisation des annotations, en voici un exemple :

```

@WebService(
    name = "VilleWS",
    targetNamespace = "http://www.oofrance.com
  
```

A ce stade vous disposez d'un serveur offrant un service Web. Reste maintenant à l'exploiter !

Coder le client

Le client doit réaliser les étapes suivantes pour pouvoir invoquer les méthodes du web service :

- utiliser l'annotation : `WebServiceRef` pour déclarer une référence au service et renseigner l'élément `wsdlLocation` pour spécifier l'URI du fichier WSDL déployé.

```

@WebServiceRef(wsdlLocation="http://localhost:33094/Ville/VilleService?WSDL")
static VilleService service;
  
```

- retrouver le proxy du service, le port, en invoquant la méthode `getVillePort` sur le service :

```

    Ville port = service.getVillePort();
  
```

- invoquer la méthode du port, lui passant un code postal.

```
String ret = port.getVille("29200");
package client;
import javax.xml.ws.WebServiceRef;
import endpoint.VilleService;
import endpoint.Ville;

public class Client {
    @WebServiceRef(wsdlLocation="http://localhost:33094/Ville/VilleService?WSDL") static VilleService service;
    public static void main(String[] args) {
        Client client = new Client(); client.doVille();
    }
    public void doVille() {
        try {
            Ville port = service.getVillePort();
            String ret = port.getVille("29200");
            System.out.println("Hello result = " + ret);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Et voilà, vous disposez d'un client pouvant consommer ce service Web. Pour cela, nous avons utilisé une des dernières nouveautés de Java EE : des annotations pour les services Web.

Construire le service Web avec PHP

On peut utiliser les Web services depuis longtemps en PHP, cependant, avec PHP 4 il était nécessaire d'utiliser une bibliothèque externe (généralement nuSOAP). Depuis PHP 5, une extension native nommée " SOAP " est disponible. Pour tester, il vous suffit donc d'installer une plate-forme LAMP/WAMP. Nous vous conseillons d'utiliser Wampserver. Attention, vérifiez dans votre fichier de configuration (php.ini) que l'extension soap est bien activée :

```
extension=php_soap.dll
```

Ensuite, un bon éditeur, un peu d'huile de coude et vous pourrez mettre en place un service Web.

Faire appel à un service Web

Généralement, il y a plus de consommateurs de services Web que de personnes en mettant à disposition. Nous allons donc commencer par voir comment y faire appel. Les déclarations WSDL permettent d'automatiser l'utilisation d'un service Web. Vous ne manipulerez donc ni XML, ni SOAP directement.

```
<Encadré. Titre : Trouver des services web>
```

```
Amazon :
Google :
Annuaire Xmethods :
</Encadré>
```

Vous pouvez trouver des exemples de services Web sur l'annuaire de Xmethods à l'adresse : <http://www.xmethods.net/>.

Coder un client avec PHP

C'est PHP qui, en interne, va faire des échanges SOAP avec le serveur distant, à chaque fois que vous exécutez une méthode. La seule étape spécifique est d'initialiser votre objet en lui donnant l'adresse du fichier de déclaration WSDL. Pour nos exemples, nous allons utiliser un service Web très simple (nous le mettrons en place plus tard) qui se trouve à l'adresse <http://www.oofrance.com/soap/villes.wsdl>.

```
<?php
$wsdl = "http://www.oofrance.com/soap/villes.wsdl" ;
$client = new SoapClient( $wsdl );
```

Notre service permet d'associer le nom d'une ville à son code postal. Nous allons donc faire appel à la fonction `getville()` en lui indiquant un code postal.

On commence par créer un objet de type `SoapClient` en donnant l'adresse de la déclaration WSDL. Une fois cet objet créé, l'action "getville" sera visible comme une méthode locale de notre objet :

```
<?php
$cpcville = '75005';
$wsdl = "http://www.oofrance.com/soap/villes.wsdl" ;
$client = new SoapClient( $wsdl );
echo $client->getville($cpcville);
// Renvoie 'Paris 05'
```

Mettre en place le serveur SOAP

Une fois que vous avez accès à un fichier WSDL, PHP ou Java pourront le lire et l'interpréter. Ils connaîtront alors la liste des méthodes qu'ils peuvent/doivent exécuter/implémenter, et les paramètres à fournir.

Dans une base de données, nous avons une table " villes " qui dispose de deux champs, le code postal d'une ville et son nom.

Une fois le fichier WSDL généré, développer un serveur SOAP n'est pas beaucoup plus complexe que d'y faire appel en tant que client. Il s'agit de définir une suite de fonctions tout à fait classiques qui seront " offertes " dans le service Web. PHP s'occupe du reste. Pour

construire un serveur SOAP il nous suffit :

- de déclarer une fonction PHP classique ;
- d'instancier un serveur SOAP ;
- d'ajouter la fonction.

```
<?php
function getville($cp) {
    // Connexion à la base
    ...
    // Lecture d'enregistrements
    $sql = "SELECT nom FROM villes WHERE cp=$cp";
    $sth = $dbh->query($sql);
    $result = $sth->fetchAll(PDO::FETCH_ASSOC);
    return $result[0]['nom'];
}

ini_set("soap.wsdl_cache_enabled", "0"); //
disabling WSDL cache
$server = new SoapServer("villes.wsdl");
$server->addFunction("getville");
$server->handle();
?>
```

La dernière ligne de notre script serveur fait un appel à `handle()`. PHP prend alors la main, examine la requête SOAP en entrée, appelle votre fonction avec les bons paramètres et retourne la réponse en XML au client. Vous n'avez rien d'autre à faire.

Conclusion

Nous avons vu, au travers de ces exemples, comment mettre en place une architecture SOA simple qui permet de communiquer entre applications. Techniquement, c'est une solution qui se tient, mais attention à ne pas tomber dans le piège de la hype et de chercher à tout faire avec des services Web. Pour profiter du modèle SOA, il convient de mettre le focus sur les aspects business et non pas techniques.



■ Cyril PIERRE de GEYER

Cyril PIERRE de GEYER est Responsable d'Anaska Formation (www.anaska.com). Anaska est une SS2L spécialisé dans l'Open Source avec deux pôles :

Services autour d'OpenOffice, de PHP, de Java et de MySQL. Formations sur toutes les technologies libres.



■ François Le Droff

Ingénieur logiciel, spécialiste J2EE (<http://www.droff.com>). Membre de l'association de promotion de l'OpenSource OSSGTP (<http://www.ossctp.org>)

Web Services

choix et donc très logiquement, en Java. Voici un exemple de consommation en Java du service que nous avons créé précédemment :

```
package client;
import org.tempuri.Service;
import org.tempuri.ServiceSoap;
import javax.xml.ws.WebServiceRef;
public class NetClient {
    @WebServiceRef(
        wsdlLocation = "http://example.com/HelloYou/Service.asmx?WSDL")
    static Service service;
    public static void main(String[] args) {
        ServiceSoap port = service.getServiceSoap();
        System.out.println(port.Hello("Programmez"));
    }
}
```

Le fait que ceci soit possible est bien entendu dû au respect des différents WS-* par les deux technologies, et à leur effort créer des requêtes et réponses SOAP standard.

Dans ce cas, il s'agit évidemment d'un appel de méthode très simple via un type de base en .net (string), transformé en objet pour Java (java.lang.String). A savoir que cela fonctionne tout aussi bien lorsque vous voulez consommer un service Web renvoyant un objet instance d'une classe que vous avez créée.

Problèmes dans des cas particuliers à cause de l'interprétation des standards

Sachant que tout le monde n'interprète pas forcément les standards de la même manière, il y a un pas entre le fait qu'ils existent et leurs implémentations. C'est ce qui explique certains problèmes de compatibilité.

D'autre part, certaines limites sont liées à la structure de quelques éléments des langages. En Java par exemple, un objet java.util.Date peut être null, tandis que System.DateTime en .NET est considéré comme un type valeur, qui, jusqu'à la version 2.0, ne pouvait pas être null (problème réglé à l'heure actuelle, via la possibilité de rendre " nullable " une variable de type valeur). Malgré cela, il subsiste tout de même certains problèmes lors de la comparaison de 2 dates.

Ne soyez pas trompés par le fait que le " O " de SOAP signifie " Object ". Ce protocole n'est pas orienté objet. Ceci pose un problème au niveau de la gestion des levées d'exceptions. En effet, le protocole SOAP ne met à disposi-

tion des développeurs que l'élément SOAPFault pour indiquer au client d'un service Web qu'une erreur est survenue. La problématique qui se pose est donc de savoir comment transformer le paradigme objet des exceptions en SOAPFault, et surtout de savoir comment être certain que la technologie recevant ce type de message puisse l'interpréter correctement.

Il existe encore d'autres problèmes, comme l'impossibilité d'utiliser facilement les transactions lors de la consommation d'un service Web exposant un accès à une base de données, ou la gestion de la sécurité.

Une autre problématique lors de la transmission d'un message SOAP est l'envoi de données binaires. Si ces données sont trop volumineuses, le serveur hébergeant le service Web risque de ne pas survivre très longtemps. Une solution consisterait à compresser le message avant son envoi et à le décompresser à l'arrivée. Tout l'enjeu réside donc dans la standardisation de cette compression...

Initiative d'amélioration de l'interopérabilité

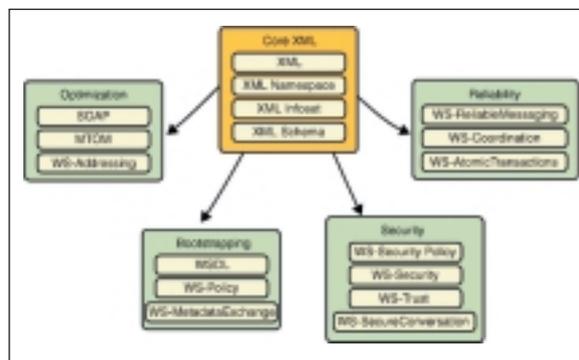
Le projet Web Services Interoperability Technology (WSIT) est un rassemblement de technologies qui doivent permettre une interopérabilité entre les Web Services Java et ceux en .Net. Les 3 points traités sont : l'optimisation des messages, la qualité de service (QoS) et la sécurité.

WSIT fournit une standardisation de la compression des objets binaires. Ceci permet d'optimiser les performances des serveurs qui peuvent maintenant compresser les données envoyées.

Au niveau de l'intégrité des données, WSIT se contente de rassembler les 3 technologies : WS-Coordination, WS-AtomicTransactions, WS-ReliableMessaging.

Le but est qu'une série de messages puisse être envoyés entre un point A et B sans aucune perte, avec réémission automatique des messages perdus. Cette base doit également permettre de créer un principe de session similaire à celui en HTTP.

Jusqu'à maintenant, les Web services dépendaient du SSL pour sécuriser la connexion. WSIT regroupe les 4 technologies : WS-Trust,



WS-Security, WS-Security Policy et WS-Secure Conversation. De plus, le WSIT apporte les fonctionnalités suivantes :

- Web Services Security Policy: Permet aux Web services d'utiliser des assertions de sécurité pour représenter de façon claire les préférences et besoins de sécurité.
- Notion de préférence de sécurité. Ceci permet de définir des règles de sécurité qui vont s'appliquer pour un ensemble de Web Services.

L'implémentation de ces fonctionnalités par le WSIT a pour but de s'assurer que les besoins des Web services en sécurité respectent le fichier WSDL. L'implémentation Java de WSIT s'appuie sur l'API JAX-WS. En .Net elle est incluse dans Windows Communication Foundation (WCF). Pour l'instant, Sun et Microsoft n'ont pas fait beaucoup de tests avec cette technologie, mais avec les nouvelles versions de WSIT, la compatibilité devrait s'améliorer.

Conclusion

La compatibilité actuelle des Web Services n'est pas parfaite dans le cas d'applications complexes. Cependant, les standards ne manquent pas et les grands acteurs de l'informatique font tous des efforts pour que l'interopérabilité soit parfaite. Les prochaines implémentations de WSIT devraient encore améliorer la chose.

■ Alexandre Decaesteker

Formateur .Net à Supinfo
alexandre.decaesteker@supinfo.com

■ Etienne Margraff

Formateur .Net à Supinfo
etienne.margraff@supinfo.com

■ David Payet

Formateur Java à Supinfo
david.payet@supinfo.com

■ Olivier Smedile

Formateur Java à Supinfo
olivier.smedile@supinfo.com

Les standards des Web Services

Le W3C (World Wide Web Consortium) définit un web service comme étant un système logiciel conçu pour permettre une pleine interaction entre machines à travers un réseau. Cette recherche d'interopérabilité a conduit à l'apparition de nombreuses spécifications qui sont aujourd'hui devenues des standards ouverts. Ceux-ci représentent la clef de voûte des Web Services et de l'interopérabilité des systèmes.

Le principe des Web Services est simple : les systèmes interagissent avec le web service en utilisant des messages dont la composition dépend de l'interface exposée. Ces messages transitent sur le réseau en utilisant des protocoles standard, par exemple TCP/IP (Transport Control Protocol/Internet Protocol) et HTTP/HTTPS (HyperText Transport Protocol) pour le transport sur le réseau, et XML (eXtensible Markup Language) dans une enveloppe SOAP (Simple Object Access Protocol) pour formater les informations transmises. L'interface exposée du web service, la surface émergée de l'iceberg, est décrite à l'aide du langage de description WSDL (Web Service Description Language).

Différents organismes

Les principaux organismes garants de l'interopérabilité des Web Services sont le W3C et l'OASIS (Organization for the Advancement of Structured Information Standards). Le WS-I (Web Services Interoperability Organization) représente quant à lui, un effort collectif des industriels permettant de promouvoir et d'améliorer l'interopérabilité des différentes implémentations de web services entre les plates-formes, les systèmes d'exploitation et les langages de programmation. Le WS-I offre aux développeurs et aux architectes une ligne directrice, des bonnes pratiques et des ressources optimisant le développement de solutions interopérables orientées service. Pour les développeurs, l'utilisation de ces standards permet l'emploi des langages, les technologies et les systèmes de leur choix. N'oublions pas que l'échange de données sur un réseau est critique, surtout si les données sont confidentielles. Pour assurer la sécurité des échanges, l'OASIS a créé le standard WS-Security, également nommé WSS, garantissant l'intégrité et la confidentialité des informations. L'OASIS est aussi à l'origine de

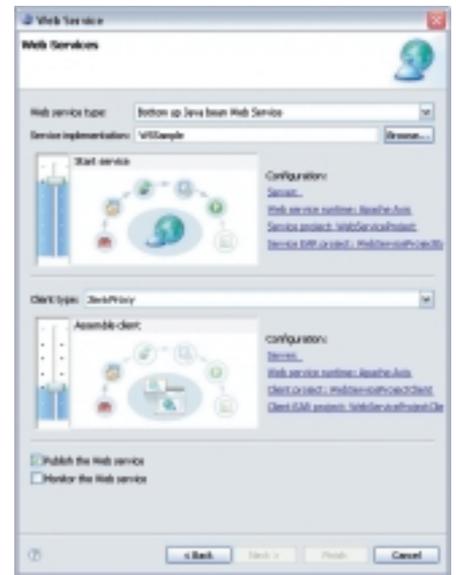
WS-Reliability permettant de garantir la qualité de service de la livraison des messages pour les applications critiques.

Le succès des Web Services n'est sûrement pas étranger à l'implication et à la promotion faites par Microsoft, qui a participé activement à l'élaboration des spécifications. Aujourd'hui, les outils s'adaptent et intègrent le support des Web Services dans leurs éditeurs et assistants. C'est notamment le cas de Netbeans et de Eclipse avec son plug-in Web Tools qui permettent de créer des services web Java simplement et rapidement.

Le succès des services web, un pari gagné grâce aux standards ?

■ Julien Chable

Administrateur CodeS Sources



Standard et interopérabilité : attention aux illusions !

La multiplication des standards complexifie l'usage des web services. Les standards de base (WSDL, UDDI, SOAP...) sont aujourd'hui matures. Cependant, le standard n'est qu'une spécification. Chaque éditeur, chaque projet peut interpréter et implémenter ce standard. " Il peut y avoir des zones floues, même si l'interopérabilité est mise en avant. Il existe WS-i et son Basic Profile qui clarifie la lecture de SOAP et WSDL. Il existe différentes implémentations d'un standard, bref une variabilité de celui-ci. Cela va sans doute perdurer. On va fédérer tout cela dans un bus ESB qui assurera l'interopérabilité. Cependant, il faut faire attention aux standards de haut niveau, il faut s'assurer de son support dans l'environnement web service (utilisé). Il faut impérativement des tests sur le terrain. Par exemple, pour notre solution d'ESB, nous avons fait des tests d'interopérabilité avec Microsoft, Axis... Aujourd'hui, il existe des bonnes pratiques et les outils ont beaucoup évolué depuis l'apparition des web services. Le développeur n'a plus à tout écrire, par exemple sous Java EE avec l'usage des annotations " analyse Alexandre Vasseur de BEA France.

Bref, le standard ne garantit pas une interopérabilité de vos web services. Assurez-vous que l'implémentation et la version des spécifications soient correctes et supportées dans votre environnement de conception et d'exécution. Les initiatives de type WS-* et les accords bilatéraux comme ceux de Microsoft et Sun améliorent la situation mais ne résolvent pas tout. Si vous êtes dans un contexte hétérogène ou que vous devez interagir avec des web services " étrangers " (non issus de votre solution de développement), les tests seront cruciaux. Soyez très vigilant sur les standards dits de haut niveau, car là, la maturité reste à atteindre. De plus, dans ces standards, il arrive que les éditeurs rajoutent leur propre fonction.

■ François Tonic

L'EPSI : de la passion de l'informatique au diplôme d'ingénieur !

L'EPSI (Ecole Privée des Sciences Informatiques) existe depuis 1961. Elle se décline sur 6 sites en France et accueille un millier d'étudiants environ, dont 250 sur Paris. Les études s'effectuent de la sortie du Bac au diplôme d'ingénieur. Avec deux constantes : un fort esprit " école " et une vraie mixité sociale.

Si certains informaticiens font leur scolarité de 5 ans à l'EPSI (de la préparation intégrée de 2 ans au diplôme d'ingénieur), les autres rejoignent l'école pour le cycle d'ingénieur qui dure 3 ans. Ce qui favorise une véritable mixité sociale. " Certaines écoles d'ingénieurs sélectionnent leurs étudiants issus de maths sup ou maths spé. A l'EPSI, ce n'est pas le cas : les passionnés d'informatique peuvent s'inscrire chez nous, dès l'obtention du bac. Nous attirons égale-



ment les étudiants de BTS ou DUT qui se posent la question de la poursuite de leurs études jusqu'au diplôme d'ingénieur ", indique **Albert Cohen**, DG de l'EPSI.

Une pédagogie d'école

D'où la particularité d'une scolarité avec tous ses attributs : des salles de classes, aucun amphî, l'appel avant le début des cours, un



travail régulièrement noté et même un danger de redoublement pour ceux qui ne seraient pas au niveau requis ! " Encadrer, contrôler, c'est tirer vers le haut. Par rapport aux écoles traditionnelles où le plus dur est de rentrer, chez nous, c'est de sortir, après avoir franchi toutes les étapes ", affirme le directeur.

C'est pour un contact très proche avec le monde du travail que les étudiants de l'EPSI

doivent faire des stages tout au long de leur scolarité, sachant que celui de dernière année (6 mois) est souvent considéré comme une pré-embauche.

"L'enjeu est important, indique le DG de l'EPSI. Ils font très attention à bien choisir l'entreprise. Leur rémunération à l'embauche est dans une fourchette de 34 à 38 K€. Mais ce qui leur importe le plus est la mission et l'évolution qui leur sont proposées."

L'ascenseur social à Paris et en province

Selon A. Cohen, dès la remise des diplômes, 95 % des étudiants ont déjà été embauchés comme ingénieurs en CDI avec le statut de cadre, les autres trouvant leur emploi dans le mois suivant la fin de leur scolarité : la cote de l'EPSI sur le marché des écoles informatiques est donc bonne. Autre indice : certaines entreprises comme Sogeti, Sopra ou Accenture viennent régulièrement dans les locaux de l'EPSI afin de présenter les missions qu'ils peuvent proposer aux futurs ingénieurs. Enfin, les anciens (plus de 2 000) donnent régulièrement des conférences aux nouveaux pour leur permettre de mieux appréhender les métiers. Une autre façon d'intégrer un "réseau" nécessaire dans toute carrière.

A l'instar d'Olivier Clos, DSI chez Orchestra, PME de l'Hérault fabriquant et vendant des vêtements pour enfants, et qui a été élu DSI de l'année par O1Informatique en 2003...



En savoir plus :

- Le **diplôme d'ingénieur** de l'EPSI est homologué par l'Etat.
- Le **coût de la scolarité** est de 4 600 € par an pour le cycle préparatoire et de 6 100 € par an pour le cycle d'ingénieur.

L'EPSI est répartie sur **6 sites** en France et compte plus de 120 professeurs et professionnels : Paris (un quart d'étudiants environ), Bordeaux, Montpellier, Arras, Nantes et Lyon.

- L'EPSI proposera des **VAE (validation des acquis de l'expérience)** à la rentrée 2007.

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

Epsilab : Microsoft Vista et Office 2007

Arras a été moteur sur le projet Epsilab, lieu de relations entre l'école et les entreprises sur des projets à haute valeur technique. Elève en 2ème année du cycle d'ingénieur à Arras, Grégory Huet, président d'Epsilab, indique : " en 2005, des conférences sur PCP/IP ont été faites. Nous avons également travaillé sur la migration de IPv4 vers IPv6 et sur TC/PIP. Cette année, nous allons réfléchir aux systèmes Microsoft Vista et Office 2007 et l'un de nos thèmes de travail sera de comparer .Net et Java. "

Le dynamisme d'Arras a séduit les autres sites d'Epsi : toutes les écoles se dotent progressivement d'un Epsilab, car l'émulation est aussi contagieuse ! www.epsilab.com



epsilab
L'Association des étudiants de l'EPSI

Information au menu

1

1 an ECO

Recevez le magazine
chaque mois
et économisez 20 €

11 Numéros
Prix au numéro : 65,45 €

45 €

(Prix France métropolitaine)



2

1 an TOUT NUMERIQUE

Lisez chaque mois le
magazine seul
en format PDF
(téléchargement)

Inscription :
www.programmez.com

11 Numéros

35 €

Tarif Monde entier



3

1 an ETUDIANT

Vous devez justifier
de votre statut d'étudiant.
Economisez 26 €

11 Numéros
Prix au numéro : 65,45 €

39 €

(offre réservée France métropolitaine)



4

2 ans

Abonnez-vous
pour 2 ans et recevez
le livre « Métier
Développeur Kit de survie »,
de J-Chr. Arnulfo
(Dunod, 272 pages).

22 Numéros
Prix au numéro : 130,90 €
+ livre 19,90 €

90 €

(offre réservée France métropolitaine
dans la limite des stocks disponibles)



OUI, je m'abonne ! ou abonnement en ligne : www.programmez.com

- ABONNEMENT 1 an ECO** au prix de 45 € TTC. Tarif France métropolitaine.
Tarifs hors France métropolitaine : CEE et Suisse : 51,83 € - Algérie, Maroc, Tunisie : 55,95 € - Canada : 64,33 € - Tom : 79,61 € - Dom : 62,84 € - Autres : nous consulter
- ABONNEMENT 1 an ETUDIANT (11 numéros)** au prix de 39 € TTC. Offre limitée à la France métropolitaine.
Photocopie de la carte d'étudiant obligatoire
- ABONNEMENT 2 ans + livre «Métier Développeur» (22 numéros)** au prix de 90 € TTC. Offre limitée à la France métropolitaine.

M. Mme Mlle Entreprise : Fonction :

Nom : Prénom :

Adresse :

Code postal : Ville :

Tél : E-mail :

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

A remplir et retourner sous enveloppe affranchie à :
Programmez ! - Service Abonnements - 22 rue René Boulanger - 75010 Paris.
abonnements.programmez@groupe-gli.com



Offre limitée,
valable jusqu'au
30 novembre 2006

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.

Un plug-in Eclipse pour Team System

Le serveur de l'offre Microsoft Team System, appelé Team Foundation Server, propose de nombreux services dont la gestion et le partage de sources ainsi que l'assignation, le partage et le suivi de tâches.

Ces services sont tous accessibles depuis les outils de développement Microsoft, notamment Team Explorer, un Add-in à Visual Studio 2005. Tous les accès au serveur se font au travers de l'utilisation d'une API cliente, elle-même basée sur des Services Web (donc HTTP). Grâce à ce mode de communication standardisé, Microsoft offre la possibilité de développer facilement des clients d'accès aux fonctionnalités de son serveur. Plusieurs projets open source ont ainsi vu le jour, afin de fournir une grande diversité d'outils d'accès. Des éditeurs externes commencent aussi à proposer des produits, comme Teamprise.

Le plug-in Eclipse de Teamprise permet ainsi d'accéder, depuis Eclipse, à des sources archivées sur le serveur Team Foundation. Celui-ci prend aussi en charge le suivi des éléments de travail et permet, par exemple, d'associer des modifications faites aux sources à des tâches vous ayant été affectées. Le plug-in fonctionne avec Eclipse 3.0 et 3.1 ainsi que Rational Application Developer 6.0.

Installation

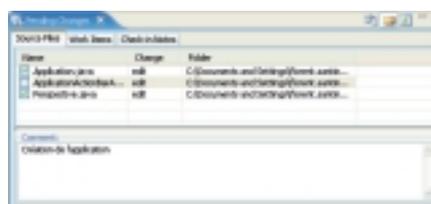
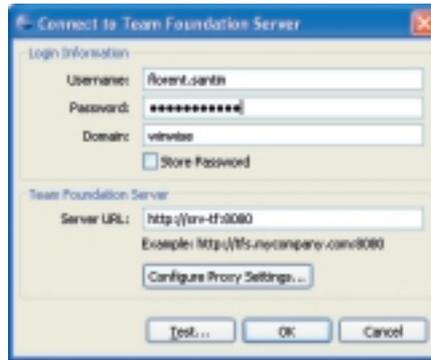
L'installation du plug-in peut se faire de deux manières

- Soit directement depuis Eclipse en utilisant " l'update manager " et en lui spécifiant l'url suivante : <http://update.teamprise.com/plugin/1.1/>
- Soit en le téléchargeant dans son intégralité afin de l'installer manuellement

Une fois installé, il est nécessaire de le configurer, afin de sélectionner le serveur Team Foundation servant de dépôt de sources ainsi que le " Team Project " ciblé. Pour ceci, il est bien sûr nécessaire de spécifier l'adresse http du serveur ainsi qu'un jeu d'identifiant/mot de passe ayant suffisamment de droits.

Utilisation

Le plug-in s'intègre dans Eclipse, comme tout contrôleur de sources, et dispose des fonctionnalités de base : Récupération de la dernière version des sources, ou d'une version spécifique (get latest version / get specific version), extraction et blocage d'un fi-



chier pour éditer celui-ci (check-out / lock), annulation des modifications locales afin de retravailler sur la dernière version valide se trouvant sur le serveur (undo pending change), mise à jour des modifications locales vers le serveur (check-in), consultation de l'historique d'un fichier (View history) et comparaison de deux versions d'un fichier (compare). Toutes ces actions sont disponibles depuis la section " Team " du menu contextuel.

Le deuxième niveau d'intégration se situe dans la fenêtre " Pending Changes " elle-même contenant plusieurs onglets:

- " Source Files " qui permet de consulter la

liste des fichiers modifiés localement en attendant d'archiver, de les sélectionner afin de créer un lot de modifications (ChangeSet) auquel un commentaire global pourra être lié et de mettre à jour le tout au niveau du serveur en cliquant sur le bouton "Check-in". Cette fenêtre permet aussi de comparer la version locale d'un fichier avec une présence sur le serveur.

- " Work Items " qui offre la possibilité de consulter la liste des éléments de travail (par exemple tâches / bug) et de les associer avec le lot de modifications courant. A noter qu'il est possible, pour chaque Work Item d'avoir accès à son contenu soit directement dans un formulaire Eclipse, soit au travers d'une page Web située sur le serveur Team Foundation.
- " Check-in Notes " qui permet d'ajouter, de saisir des commentaires spécifiques aux modifications.

Le plug-in hérite de la notion de " Work Item Tracking ", et c'est donc directement dans Eclipse, au travers de l'onglet " Work Items ", que chaque développeur va pouvoir consulter la liste des tâches qu'il doit effectuer, ou bien les bogues qu'il doit corriger. Le tout étant complètement synchronisé entre les différents intervenants et avec l'organisation globale du projet, il va être possible, hors d'Eclipse, de centraliser les informations afin de suivre l'évolution de chaque action.

Conclusion

Grâce à ce type de plug-in, il est aujourd'hui possible de faire cohabiter sans aucun souci, des sources issues d'environnements de développement différents, le tout de manière totalement intégrée : inutile d'ouvrir un outil externe pour faire de l'extraction / mise à jour de sources. De plus, grâce aux Work Items, chacune des actions effectuées sur le projet va pouvoir être tracée / analysée, offrant ainsi aux chefs de projet la possibilité de centraliser les informations d'avancement et de faciliter la gestion de projets contenant des technologies hétérogènes.

■ Florent SANTIN, Winwise

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

La gestion des exceptions

"Il y a quelque chose à dire en faveur de l'exception, pourvu qu'elle ne veuille jamais devenir la règle".

Friedrich Nietzsche

Qu'est-ce que la gestion des exceptions ?

En programmation, un système de gestion des exceptions (SGE) est un système qui permet de gérer des conditions exceptionnelles lors de l'exécution d'un logiciel. Ces conditions exceptionnelles ne sont pas des bogues, mais ne pas les traiter sera considéré comme une faute de programmation. Par exemple, le processus ne pourra ouvrir un fichier. Si une exception se manifeste, le pointeur d'exécution du processus est détourné pour exécuter des traitements liés à cette erreur. Le premier langage qui a prévu ce genre de traitement est le PL1 (Programming Language number 1) qui a été conçu fin des années 60 par IBM. De nos jours, la majorité des langages de programmation modernes disposent d'un SGE (comme C++, C#, Java, Python, etc.).

Et si les exceptions n'existaient pas ?

Nous avons vu dans la définition du SGE qu'il s'agit de gérer des conditions d'exécutions spéciales. Nous ne parlons donc pas ici d'erreurs détectables par le compilateur et/ou l'éditeur des liens (comme une erreur de syntaxe par exemple) ; mais bien d'erreurs se présentant exceptionnellement selon certaines circonstances. Par exemple, si un calcul provoque une division par zéro, ou bien si votre programme tente un accès illégal en dehors des bornes d'un tableau, etc. Si votre logiciel ne peut traiter ce genre d'erreurs immédiatement lorsqu'elles se produisent, alors votre programme plantera/bloquera, ou produira des résultats indéterminés. Si les exceptions n'existent pas dans votre langage il est nécessaire de gérer tous les cas de figure par le biais d'itératives. Suivant la nature de l'erreur, le programmeur décidera alors de poursuivre, ou bien d'arrêter le programme. Par exemple, en C il n'y a pas de SGE, et le développeur doit passer par des tests conditionnels, ou recourir à la macro `assert`. Mais des tests conditionnels comportent au moins deux désavantages :

- 1 il n'y a pas de traitement qui soit directement relié à une erreur, et rien n'oblige un programmeur de vérifier à chaque fois le code retour d'une fonction ;
- 2 dans le cas spécifique du C et d'*abort* vous ne pourrez pas fermer un *handle* en cas d'erreur (d'où "des fuites de mémoires").

Plus grave encore : le code source comportant l'ensemble de tests de conditions deviendra bien moins lisible. En outre, si une erreur se produit il est nécessaire bien souvent que l'utilisateur intervienne (production d'un message l'avertissant) puis de revenir sur le point d'entrée qui a provoqué l'erreur.

Quels sont les avantages des exceptions ?

Contrairement aux itératives de tests que le programmeur peut ne pas prévoir, une exception ne pourra être ignorée. En effet, si une exception n'est pas traitée l'application prendra fin. Dans la plupart des langages, ces exceptions sont typées et hiérarchisées et basées sur le concept "de signaux d'exceptions". Avec une SGE, le développeur peut séparer le cours normal d'exécution du cours exceptionnel. Le code source sera par conséquent plus clair et compréhensible.

Quel est le schéma d'exécution d'une exception ?

Au départ, une exception sera signalée par l'environnement d'exécution, ou encore par une méthode de signalement enclenchée par le développeur (signal), ou encore par le système d'exploitation (*), etc. Le déroulement du programme est alors dérouté vers un bloc de traitement de l'exception (en C++ l'instruction `try` définit ce bloc qui est placé entre accolades). C'est ce bloc qui décide de la suite des traitements. Soit l'erreur est fatale et il n'y aura pas de reprise possible, soit une action correctrice est envisageable (par programmation ou en sollicitant l'aide de l'utilisateur). Par exemple en C++ le principe est le suivant : un bloc de code est mis sous surveillance grâce à l'instruction `try` ; à l'intérieur de celui-ci, des fonctions peuvent envoyer un signal d'exception au programme par l'intermédiaire de l'instruction `throw` ; et enfin, dans le code source, un bloc de code de gestion d'exceptions est indiqué par l'instruction `catch`.

```
#include <iostream>
void fonction(void)
{
    throw "bonjour !";
}
int main(void)
{
```

```
try {
    fonction();
} catch (const char *valeur) {
    std::cout << valeur << std::endl;
}
return 0;
}
```

La chaîne "bonjour !" s'affiche car "fonction" a appelé un "throw" déclenchant une exception. Attention ! Nous parlons ici de "système de communication entre deux objets" : une exception lance bien un objet et non un type.

(*) : Le processeur peut signaler des événements internes comme une division par zéro, une instruction invalide, un défaut de page en déclenchant une exception par le biais d'une interruption. Sous Linux, le code lié aux exceptions se situe dans `kernel/traps.c`.

Que signifie la propagation d'une exception ?

Si jamais la clause `catch` ne peut gérer l'exception (même après une action correctrice), le compilateur peut décider que l'exception sera gérée par une fonction située plus haut dans la liste des appels de fonction. En remontant ainsi le long de la pile d'appels elle peut arriver éventuellement à "main()". Et si toutefois elle n'est toujours pas gérée, la fonction termine et fin au programme. Celle-ci appelle à son tour *abort*. Une clause `catch` peut passer l'exception à une autre clause `catch` en la propageant. L'expression de propagation prend la forme : `throw;`. Voici un exemple de propagation :

```
void FonctionManipuleChaine(void)
{
    try
    {
        Chaine sChaine("1234567890");

        // Génère une erreur d'accès en dehors des bornes
        cout << sChaine[12] << endl;
    }
    catch (Chaine::ErreurAllocation &e)
    {
        cerr << e.what() << endl;
    }
}

int main(int, char **)
```

```
{
  set_terminate(FonctionPersoTerminate);
  try
  {
    FonctionManipuleChaine();
  }
  catch (Chaine::Erreur &e)
  {
    cerr << "Propagation d'exception " << e.
    what() << endl;
    throw;
  }
  return 0;
}
```

Lorsque l'exécution arrive au niveau de l'accès au tableau sChaine qui se situe en dehors des bornes, une exception est générée. Elle n'est pas du type Chaine::ErreurAllocation et l'exception est levée. Le contrôle passe alors au niveau supérieur et donc à main(). Le bloc "try" va gérer cette erreur et relancera l'exception vers le niveau supérieur. En d'autres termes, le "throw;" va renvoyer le traitement à FonctionPersoTerminate.

Le gestionnaire de toutes les exceptions

Dans de nombreux langages comme le C++, il est possible de définir un gestionnaire d'exceptions dit "universel". Celui-ci sera chargé de récupérer toutes les exceptions quels que soient leurs types... Il suffit en C++ de spécifier trois points de suspensions entre parenthèses dans la clause catch.

```
catch (int entier)
{
  cout << "Erreur int (cause " << entier << ")\n";
}
// traitement de toutes les exceptions !
catch (...)
{
  cout << "Exception inattendue !\n";
}
```

Attention : il est évidemment impossible de vouloir récupérer l'objet dans ce cas de figure... Le catch(...) peut aussi se combiner avec une expression de propagation :

```
catch (...)
{
  cout << "Exception inattendue !\n";
  throw ;
}
```

Ce mécanisme est fort utile lorsqu'il s'agit de libérer une ressource avant que l'exception ne se propage à un niveau supérieur.

Exceptions en Java

Les exceptions en C# et Java ont de nombreux points communs (comme une hiérarchie de classes d'Exceptions qui dérivent d'une super classe : System.Exception pour C# et java.lang.Exception pour Java).

```
try {
}
catch (IOException e) {
...
}
finally {
}
```

Vous pouvez ainsi gérer des erreurs de type NullPointerException, OutOfMemoryException, ArithmeticException, ClassCastException, ArrayIndexOutOfBoundsException, negative ArraySizeException. Il est aussi facile en Java de définir votre propre exception :

```
public class MonException extends Exception {
  public MonException(String text) {
    super(text);
  }
  public String toString() {
    ...
  }
}
```

Exceptions en Python

Voici un exemple simple :

```
#!/usr/bin/python
try:
  file = open("fichier.txt", 'r')
except IOError:
  print "Impossible d'ouvrir le fichier"
except:
  print "autre erreur"
```

Vous pouvez également créer un gestionnaire général comme en C++ capable d'intercepter toutes les exceptions :

```
#!/usr/bin/python
import traceback
def MonCrochetSGE(type, value, tb):
  print "in Myexcepthook"
  lines=traceback.format_exception(type,
  value, tb)
  print "\n".join(lines)
  sys.exit(2)
sys.excepthook=MonCrochetSGE
```

Exceptions en C#

Commençons par un court exemple :

```
using System;
class Class1
```

```
{
  public static int Main(string[] args)
  {
    Console.WriteLine(Divise(100,6));
    Console.WriteLine(Divise(100,0));
    return 0;
  }
  public static double Divise(int a, int b)
  {
    return a/b;
  }
}
```

Ce code déclenche la génération d'une exception puisqu'il se produit une division par zéro. En C# pour capturer l'exception on utilise les blocs try...catch comme en Java.

```
try
{
  Console.WriteLine(Divise(100,6));
  Console.WriteLine(Divise(100,0));
}
catch
{
  Console.WriteLine("Attention : une division par
zéro est impossible !");
}
return 0;
```

En fait, il est concevable d'intercepter une exception spécifique mais il est conseillé évidemment de garder une porte de sortie pour une exception non connue.

```
try
{
  Console.WriteLine(Divise(100,6));
  Console.WriteLine(Divise(100,0));
}
catch (DivideByZeroException e)
{
  Console.WriteLine("Attention : une division par
zéro est impossible !");
}
catch (Exception e)
{
  Console.WriteLine(e.Message);
}
return 0;
```

Comme en C++ ou en Java une exception peut être lancée avec throw : throw new Exception("Exception personnelle"); Enfin le mot clé "finally" désignera un bloc qui sera toujours exécuté.

■ Xavier Leclercq - Xavier.Leclercq@programmez.com

WebLogic Workshop: l'usine à SOA

Une des grandes tendances actuelles est la mise en place des architectures SOA (service oriented architecture). BEA propose un IDE très complet, et performant, Workshop. Un outil Java EE reconnu par le marché.

Workshop se décline en deux versions: Workshop for WebLogic Platform 9.2 et Workshop Studio 3.0. Ces IDE se basent, entre autres, sur WTP (Web Tools Platform) & Nitrox (une distribution d'Eclipse ayant déjà un ensemble de plug-ins, dont des éditeurs graphiques pour les JSP, ...). Workshop for WebLogic Platform 9.2 est le successeur de Workshop 8.1 et est l'IDE dédié à la plate-forme serveur (WebLogic Platform c'est en réalité le regroupement de WebLogic Server, WebLogic Integration ainsi que WebLogic Portal).

La version 3.0 de Workshop Studio est axée sur Java EE 5.0. Même si le serveur n'est pas encore Java EE 5.0 "compliant", mais seulement J2EE 1.4, celui-ci supporte tout de même JSF, les EJB 3, les annotations, ... La migration du serveur vers la plate-forme Java EE est en cours. Workshop Studio 3.0 est un IDE ayant pour vocation des interactions avec d'autres serveurs d'application (JBoss, WebSphere Application Server, Resin, Tomcat, Jetty) et bien évidemment également WLS. Des assistants permettent de créer rapidement des EJB 3, et un éditeur WYSIWYG pour les JSF, JSP, les processus métiers ainsi que les pageflow sont intégrés.

Que ceux qui n'ont jamais utilisé Workshop ne s'inquiètent pas sur le temps de prise en main de cet IDE, c'est en réalité un ensemble de plug-ins ajoutés à Eclipse 3.1.2. La précédente version de Workshop 8.1 n'était pas basée sur Eclipse. Son utilisation est donc grandement facilitée et vous pouvez, tout comme avec Eclipse, le personnaliser à souhait. Vous pouvez même installer Workshop for WebLogic Platform 9.2 grâce à l' "Update Manager" d'un Eclipse existant (contenant déjà vos propres plug-ins).

Le principal attrait de Workshop est la présence d'un éditeur graphique nommé "Page Flow Editor" qui permet de définir votre système de navigation de votre application Java EE. Que vous utilisiez Struts, Beehive ou JSF, vous pourrez utiliser cet éditeur de la même manière. Le fichier créé portera alors l'extension JPF (Java Page Flow) et permettra de modéliser les actions, les pages JSP, ... C'est donc un niveau d'abstraction supérieur à Java EE qui permet de définir votre navigation sans avoir à être dépendant de l'implémentation utilisée.

Lorsque vous utilisez Workshop, vous pouvez utiliser toutes les vues & perspectives classiques présentes dans Eclipse, mais vous avez en plus une perspective "Workshop" qui se compose de plusieurs vues dont entre autres : "AppXplorer, DbXplorer, DbXaminer", l'éditeur ainsi que la vue "Properties". Ces vues vous permettront d'utiliser les fonctionnalités de Workshop. Pour ceux qui ont déjà des projets sous Workshop 8.1, un guide de migration vers la version Workshop for WebLogic Platform 9.2 est disponible sur le site de BEA : <http://fr.bea.com/products/workshop/>.

L'architecture de l'outil

La différence majeure entre Workshop for WebLogic Platform & Workshop Studio réside dans la plate-forme vers laquelle vous souhaitez destiner vos applications. Voici le détail de l'architecture de ces deux outils :

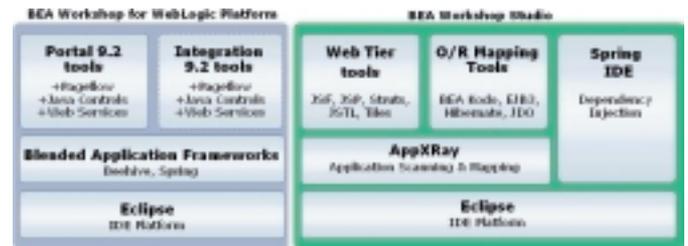


Fig : 1

Plusieurs déclinaisons de Workshop Studio existent: BEA Workshop for JSP (version gratuite), BEA Workshop for JSF, BEA Workshop for Struts et BEA Workshop Studio. Si vous avez des besoins spécifiques (soit JSF, soit Struts), vous pourrez acquérir l'une des deux offres BEA Workshop for JSF (qui intègre l'éditeur JSP ainsi que l'éditeur JSF) ou BEA Workshop for Struts (qui intègre également l'éditeur JSP ainsi que l'éditeur Struts). En revanche, si vos besoins sont plus généraux, BEA Workshop Studio est l'outil à choisir car il intègre toutes les fonctionnalités des offres précédentes.

Beehive

Beehive est un projet supporté par la fondation Apache qui est sortie en version 1.0, le 30 septembre 2005. Le but de Beehive est de faciliter le développement Java EE en proposant une couche d'abstraction supérieure à Java EE & Struts. Beehive se décompose en 3 principales branches : NetUI, Controls et Web Service Metadata (WSM). Beehive NetUI est une framework graphique, c'est une surcouche à Struts (grâce aux JPF) et comprend un ensemble de taglibs JSP (composants de base ou avancés comme les grilles et les arbres). Beehive NetUI est compatible Struts et JSF.

Beehive Controls se charge de faciliter l'accès aux données gérées par vos applications d'entreprise, c'est une implémentation du modèle Composant basée sur les annotations.

Beehive WSM simplifie la création des Web Services grâce à l'emploi des annotations (JWS), c'est grâce aux Beehive Controls que vous pourrez consommer des WebServices. Une version 2.0 de Beehive devrait voir le jour. Pour plus de détails sur Beehive, vous pouvez vous rendre à l'adresse suivante, vous y trouverez notamment de la documentation et des tutoriaux : <http://beehive.apache.org/>

BEA Workshop peut utiliser Beehive, ce n'est cependant pas obligatoire, mais recommandé, car cela simplifiera vos développements et vous permettra d'intégrer plus facilement et correctement le principe de MVC grâce à ces composants "prêt à l'emploi". De plus, Beehive est en réalité la version open-source basée sur le framework WebLogic Workshop application. En effet, BEA a contribué à ce projet qui a très vite séduit la communauté des développeurs. Vous n'êtes pas obligés d'utiliser Workshop pour intégrer Beehive, un projet de la fondation Eclipse nommé "Pollinate" vous propose une gestion de Beehive directement dans cet IDE.

Mise en œuvre

Pour comprendre le fonctionnement de Beehive et de Workshop, nous allons prendre un cas pratique classique : Un Webservice interrogeant une base de données. Nous allons utiliser MySQL 5 comme SGBD et créer une base de données nommée "my_data_base". Dans cette base, je crée une table nommée "client" contenant trois colonnes "id", "firstname" et "lastname". Il faut ensuite mettre le fichier driver MySql (fichier Jar) dans le répertoire lib de WLS. A présent, vous devez configurer la source de donnée au sein de la console d'administration de WLS : <http://localhost:7001/console/>

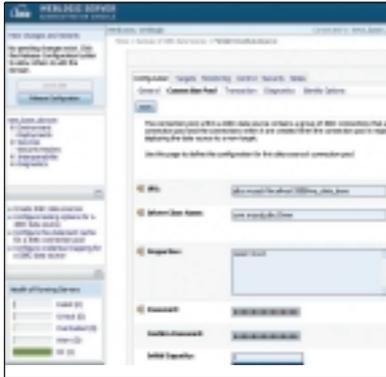


Fig : 2

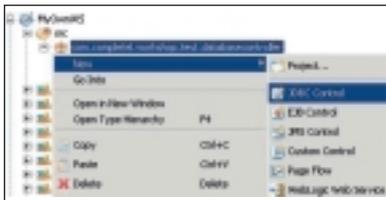


Fig : 3

Le nom à donner à la data-source ici est "MySQL5 TestDataSource" comme vous pouvez le voir dans l'image : Fig : 2

Nous allons à présent créer un nouveau projet grâce à Workshop for WebLogic Platform 9.2 de type "Web" > "Dynamic Web Project" (les utilisateurs d'Eclipse et WTP reconnaîtront ce type de projet).

Nous allons créer le Beehive Control qui nous permettra de dialoguer avec la datasource que nous venons de définir. Pour créer ce Jdbc Control, que nous nommerons

"DataSourceCtrl", utilisez le menu contextuel suivant : Fig : 3

Voici le code à insérer dans votre interface :

```
package com.completel.workshop.test.databasecontroller;

import java.sql.SQLException;

import org.apache.beehive.controls.system.jdbc.JdbcControl;
import org.apache.beehive.controls.api.bean.ControlExtension;

@ControlExtension
@JdbcControl.ConnectionDataSource(jndiName = "MySQL5TestDataSource")
public interface DataSourceCtrl extends JdbcControl {

    static public class Client {
        private int id;
        private String firstname;
        private String lastname;

        public int getId() {
            return id;
        }

        public void setId(int i) {
            id = i;
        }
    }
}
```

```
}

public String getFirstname() {
    return firstname;
}

public void setFirstname(String firstname) {
    this.firstname = firstname;
}

public String getLastname() {
    return lastname;
}

public void setLastname(String lastname) {
    this.lastname = lastname;
}

@JdbcControl.SQL(statement = "SELECT ID, FIRSTNAME, LASTNAME
FROM CLIENT WHERE ID = {id}")
Client findClient(int id) throws SQLException;
}
```

Nous pouvons maintenant créer le service web, que nous nommerons "DataSourceWS". Pour cela, utilisez le menu précédent et créez un "WebLogic Web Service". Dans l'éditeur de code, vous pouvez à présent utiliser le menu contextuel suivant afin de sélectionner votre



Fig : 4

précédent Beehive Control à utiliser (Fig : 4).

Ce menu vous permettra d'utiliser les annotations Beehive afin de faire de l'injection de dépendance.

Utilisez également ce menu pour ajouter une nouvelle "Web Method" à votre Webservice. Voici le contenu de "DataSourceWS" :

```
package com.completel.workshop.test.databasecontroller;

import java.sql.SQLException;

import javax.jws.WebMethod;
import javax.jws.WebService;

import org.apache.beehive.controls.api.bean.Control;

import com.completel.workshop.test.databasecontroller.DataSourceCtrl.Client;

@WebService
public class DataSourceWS {

    @Control
    private DataSourceCtrl dataSourceCtrl;

    @WebMethod
    public String findClientByPrimarykey(int id) {
        String res = null;
    }
}
```

```

try {
    Client client = dataSourceCtrl.findClient(id);
    res = "firstname : " + client.getFirstname() + " , lastname : " +
client.getLastname();
} catch (SQLException e) {
    e.printStackTrace();
}
return res;
}
}

```

Il ne reste plus qu'à tester votre Webservice en utilisant le menu contextuel sur le fichier et choisir "Run As" > "Run on Server". WLS intègre en effet un client de test pour Webservice. Voici la page qui vous sera proposée (ci-contre).

Saisissez alors une valeur d'identifiant, validez et vous aurez le résultat de la figure 6.



Fig : 5



Fig : 6

Vous pourrez remarquer que vous avez une historisation des différents appels que vous aurez faits ainsi que la requête et réponse SOAP qui vous ont permis d'obtenir votre résultat. Il vous suffit ensuite de créer votre propre client, que ce soit en utilisant Beehive NetUI, JSF, Struts, ou simplement avec des JSP.

Retour d'expérience

Chez COMPLETEL, nous avons choisi la plate-forme WebLogic 8.1 pour automatiser notre workflow de commandes et pour notre portail web "partenaires". Nous utilisons donc intensivement l'IDE Workshop depuis octobre 2005. Ecrites en Swing, les versions de Workshop antérieures à la 9.0 souffraient un peu de la comparaison avec Eclipse au niveau de la réactivité et surtout des fonctionnalités d'édition de code (auto-complétion, suggestions, refactoring, templates, ...). En revanche, son intégration parfaite avec les produits serveur ainsi que son mode debug très complet (sur classes Java, mais aussi JSP, processus métiers, ...), faisaient de Workshop un outil incontournable pour tous nos développements. En choisissant d'appuyer Workshop 9 sur Eclipse, BEA compense intelligemment les faiblesses de son IDE, tout en surfant sur l'engouement que cette plate-forme ouverte suscite actuellement dans la communauté Java.

Ne serait-ce que pour créer un Web Service (JWS), Workshop 8.1 se sert d'annotations insérées dans le commentaire Javadoc, en y regardant

de plus près cela rappelle XDoclet. L'utilisation des annotations au sein de Workshop est intégrée depuis longtemps, maintenant celles-ci reprennent le concept JavaSE 5.0 (JSR 175). Beehive possède également ses propres annotations à spécifier avec les Controls, ce qui vous permettra de définir vos propres composants SOA.

Lorsque vous installez l'une des versions de Workshop, vous pouvez également installer la plate-forme complète (WebLogic Server, WebLogic Portal & WebLogic Integration) afin de tester vos développements. Vous aurez également à disposition de nombreux outils, dont WebLogic Scripting Tool (WLST) et JProcessUnit. WLST facilite les tâches de déploiement et permet de "scripter" la création de ressources, la configuration du serveur... JProcessUnit sert à réaliser des tests de montées en charge et unitaires. Voici une vue d'ensemble de Workshop for WebLogic Platform 9.2 :

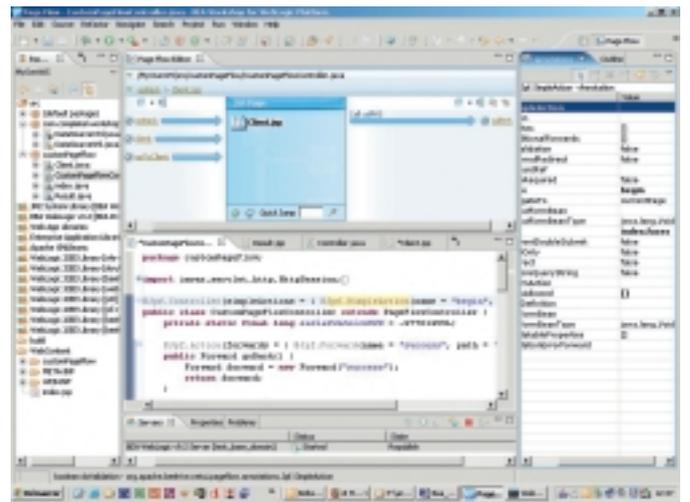


Fig : 7

Les évolutions

En ce qui concerne les évolutions coté WLS (WebLogic Server), là aussi des nouveautés sont apparues avec l'ajout d'un serveur SIP intégré (très pratique pour les entreprises étant dans le monde des télécommunications), d'outils pour faire de la répartition de charge (load-balancing), du clustering, du monitoring... Cette nouvelle version de Workshop, basée sur Eclipse, permet de profiter de tous ses avantages : système de plug-ins, auto-complétion réactive et intelligente, fonction avancée du débogueur, ... Elle ajoute également de nouvelles options, surtout grâce aux assistants comme des éditeurs visuels pour les JSP, JSF, Struts et EJB 3. Le revirement de BEA est de proposer plus d'outils pour les standards et l'OpenSource que des solutions propriétaires. Un outil d'ORM (Object Relational Mapping) pour les EJB3 est également disponible et permet de recréer très facilement des Entity Beans à partir d'une base de données. D'autres produits de BEA devraient également être suivis, comme BEA AquaLogic (ex-Liquid Data) qui est en réalité leur ESB qui intègre également une gestion des services. La fondation Eclipse a même décerné un "Community awards 2006" comme étant le " Best Commercial Eclipse-based Tool " pour



BEA Workshop. Un produit à suivre donc pour tous ceux qui font de l'intégration.

■ Maxime VIALETTE

Développeur Java EE chez COMPLETEL
Certifié SJCP & MCAD.NET - Membre du laboratoire SUPINFO des technologies Sun

Développement Linux avec Anjuta et les bindings GTKmm et SDLtnet

Anjuta est un IDE assez méconnu sous Gnome mais qui pourtant, comme nous allons le voir, dispose de plus d'une corde à son arc. Après tout, de nombreux développeurs sous Linux travaillent encore avec un simple éditeur de texte. Vim ou Emacs sont de très bons outils. Cependant, la maîtrise de ces outils prend du temps, et c'est là qu'Anjuta devient intéressant, car sa prise en main est rapide et assez intuitive.

1 Pourquoi choisir Anjuta ?

Nous avons installé sous Ubuntu Dapper la version 1.2.4a avec un simple `sudo aptitude install anjuta libcv-dev libcvaux-dev opencv-doc autoconf automake1.4 pkg-config`

```
#sudo aptitude search anjuta
i  anjuta - A GNOME development IDE, for C/C++
i  anjuta-common
```

L'éditeur convivial est le coeur d'Anjuta. Vous pouvez y éditer plusieurs codes sources en ouvrant différents onglets. Chaque onglet pourra être détaché ou non en une fenêtre d'édition individuelle (bouton droit "Attaché" (Docked)). Lors de l'encodage vous pouvez demander l'indentation du code qui est paramétrable dans les préférences. Vous disposez du style Berkeley, du style GNU, du style Kernighan et Ritchie, etc. En fait l'IDE fait appel à la commande `indent`. Remarquez que celle-ci devra être préalablement installée indépendamment de l'IDE (`sudo aptitude install indent`). Les préférences de l'éditeur comprennent également le paramétrage de l'enregistrement automatique des fichiers, ainsi que diverses règles de création de "battants". Ceux-ci permettront au développeur de masquer des blocs de codes un peu à la manière de ce que pratique l'explorateur de fichiers pour masquer l'arborescence d'un répertoire. Un signe "-" (ou un autre style) apparaît dans la marge gauche de l'éditeur indiquant un rabat. Il est ainsi envisageable de rabattre un bloc de commentaires aussi bien que des accolades. Chaque paire d'accolades est vérifiable, et les espaces inutiles peuvent aussi être supprimés de l'enregistrement. Une fonctionnalité que nous avons appréciée est la possibilité de créer rapidement un signet dans le code. Les signets étant créés, rien n'est alors plus facile que de naviguer dans le code.

Anjuta travaille par projet mais uniquement dans le cas de C/C++ (C ou C++, ou mixte C/C++). Le programmeur peut décider d'importer un répertoire de codes sources pour le transformer en projet. Mais si vous travaillez dans un autre langage (nous allons examiner comment travailler en C# conjointement avec mono), le mode projet ne sera pas applicable (par exemple en C#, java, Python ou PHP). Mais revenons-en à l'éditeur : si la syntaxe du langage est reconnue, vous disposerez d'une mise en couleur automatique du code (ce que l'on appelle la colorisation syntaxique). En outre, en mode projet, un navigateur de code (onglet 'Symboles') vous permettra de voyager dans vos classes, structures, fonctions, variables et macros. Par simple double-click, vous pourrez pointer sur un élément et vous retrouver dans le code à la position idoine. Pour faciliter l'encodage un rappel des prototypes de

fonctions est proposé lors de la saisie. Enfin, soulignons qu'Anjuta s'interfacera naturellement à CVS et à Gdb.

2 Créer une interface

Il est particulièrement intéressant de souligner qu'Anjuta est capable de s'interfacer avec Glade, un concepteur visuel d'interfaces graphiques GTK. L'interface se construit de manière graphique à la souris. Pour y parvenir, il vous suffit de démarrer un nouveau projet GTK. Entre parenthèses, Anjuta a été entièrement programmé avec GTK+. Concrètement, lors de la création d'un nouveau projet, l'IDE vous propose un assistant graphique de création d'application avec lequel vous pouvez sélectionner un projet gtkmm (bibliothèque C++ sous licence GPL de fonctions GTK+). Vous pouvez construire votre projet (menu construire/tout construire), et l'exécuter. Au départ, vous allez vous retrouver dans une fenêtre GTK vierge. Si vous voulez en modifier l'aspect ou lui ajouter des widgets vous devez vous diriger dans le menu Projet 'Editer IHM..'. Cette action aura pour effet d'exécuter Glade (qui aura été installé si nécessaire (*)).

Si nous cliquons sur la fenêtre principale pour pouvoir en éditer les propriétés, nous pourrions par exemple rebaptiser le titre de cette fenêtre. Ensuite, nous pouvons tout aussi facilement ajouter un bouton de commande auquel sera associé un événement : en cliquant sur l'onglet 'Signaux' des propriétés de celui-ci, vous affecterez un signal 'released' au traitement `gtk_main_quit`. Attention : le bouton de commande sera déposé sur une vbox (une "table" que vous trouverez avec la palette basique GTK+). Il faudra faire de même pour le signal `destroy` de la fenêtre. Enfin, vous devrez 'Enregistrer' puis 'Générer' et quitter Glade. A ce stade, si vous reconstruisez le projet et le réexécutez, vous ne constaterez aucun changement... En fait, les fichiers modifiés sous Glade n'ont pas été rechargés dans Anjuta. Glade génère les fichiers `callbacks`, `interface`, et `support` (extensions `.c` et `.h`) ; ainsi que le fichier `main.c`. Ce dernier est éditible et modifiable : il est généré mais n'est pas écrasé à chaque modification de l'interface.

Dans la fenêtre nommée `Anjutagtkmm` se trouve une `table1` puis un

(*) Nous avons rencontré quelques difficultés pour installer Glade en raison de dépendances non résolues avec `apt-get`. Nous vous conseillons d'utiliser `Aptitude` à la place d'`apt-get` car il permet de résoudre beaucoup plus de problèmes de dépendances.

```
aptitude search glade
p  eglade - Generates Eiffel code from XML produced by glade GUI builder
i  glade - GTK+ 2 User Interface Builder
id glade-2 - Dummy upgrade package for glade
i  A glade-common - Common files for GTK+ 2 User Interface Builder
...
```

button1 (vue hiérarchique). Les propriétés du bouton de commande indiqueront 33 comme "bourrage vertical", dans le but de centrer le bouton à l'exécution. Un autre fichier modifiable est *callback.c*.

Sous Anjuta, l'ajout de bibliothèques passe par le menu 'Préférences/Compilateur/Editeur de liens'.

Une fois l'application prête à la diffusion, vous pouvez employer l'entrée 'Préparer distribution' du menu 'Construire' pour obtenir une archive distribuable du projet au format tar.gz !

Voici un des fichiers source généré (Gtkmm.cc), qui est généré initialement mais non écrasé à chaque génération de l'interface :

```
// generated 2006/8/2 13:21:59 CEST by xl@xl-laptop
// using glademm V2.6.0
//
// newer (non customized) versions of this file go to gtkmm.cc_new

// This file is for your program, I won't touch it again!

#include <config.h>
#include <gtkmm/main.h>
#include <glib/gi18n.h>

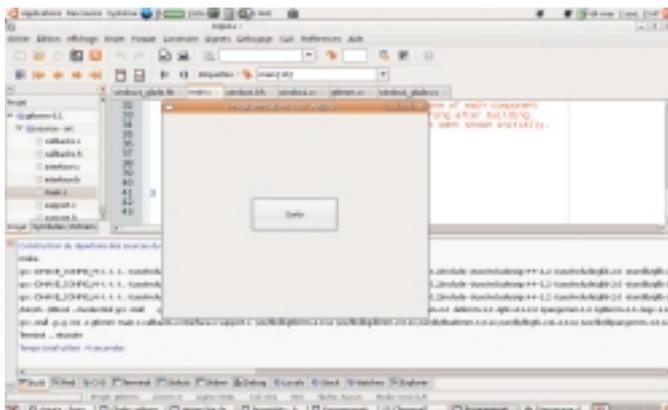
#include "window1.hh"

int main(int argc, char **argv)
{
  #if defined(ENABLE_NLS)
    bindtextdomain (GETTEXT_PACKAGE, PACKAGE_LOCALE_DIR);
    bind_textdomain_codeset(GETTEXT_PACKAGE, "UTF-8");
    textdomain (GETTEXT_PACKAGE);
  #endif //ENABLE_NLS

  Gtk::Main m(&argc, &argv);

  window1 *window1 = new class window1();
  m.run(*window1);
  delete window1;
  return 0;
}
```

Nous avons vu comment manipuler les bases d'un projet Anjuta / Glade, examinons maintenant comment utiliser Anjuta avec un autre langage comme le C#.



Exécution du projet Gtkmm.

3 Utiliser Anjuta en combinaison avec mono et SDL.Net

La bibliothèque Simple DirectMedia Layer (SDL) s'emploie dans le cadre de la programmation multimédia, par exemple dans le domaine des jeux.

Elle est sous licence GNU LGPL version 2, et est portable. Nous allons vous expliquer brièvement comment compiler et exécuter une application SDL C# sous Anjuta, mais vous pourrez très bien utiliser le même code source avec Visual Studio Express et SdlDotNet sous Windows. SDL gère non seulement l'affichage vidéo mais aussi les événements (comme celui de l'appui sur une touche), la lecture d'un CD ou les mouvements du joystick.



Sam Lantinga a créé la bibliothèque SDL en 1998, et l'a utilisée à l'époque pour porter Doom sous BeOS.

Nous allons utiliser SdlDotNet qui est un binding pour C#. Commençons par installer mono.

```
#mono -version
Mono JIT compiler version 1.1.13.6, (C) 2002-2005 Novell, Inc and
Contributors. www.mono-project.com
  TLS:      __thread
  GC:       Included Boehm (with typed GC)
  SIGSEGV  : normal
```

La version 1.1.13.6 est la dernière version stable au moment de la rédaction de ces lignes. Nous avons commencé par télécharger SdlDotNet (sdl-dotnet-4.0.4). Sous le répertoire bin/assemblies vous trouverez les DLL pour travailler avec mono. Tapons un premier programme basic SDL:

```
namespace SimpleExample
{
  public class Simple
  {
    public static void Main()
    {
      Simple app = new Simple();
      app.Run();
    }

    public Simple()
    {
      Video.SetVideoModeWindow(400,300);
      Video.WindowCaption = "Bonjour !";
    }

    public void Run()
    {
      Events.Run();
    }
  }
}
```

C'est du C# : pour le compiler vous devez utiliser en ligne de commande :

```
#mcs test.cs -r:System.Drawing.dll -r:/home/xl/SDL/sdl_dotnet-4.0.4/bin/
/assemblies/SdlDotNet.dll
```

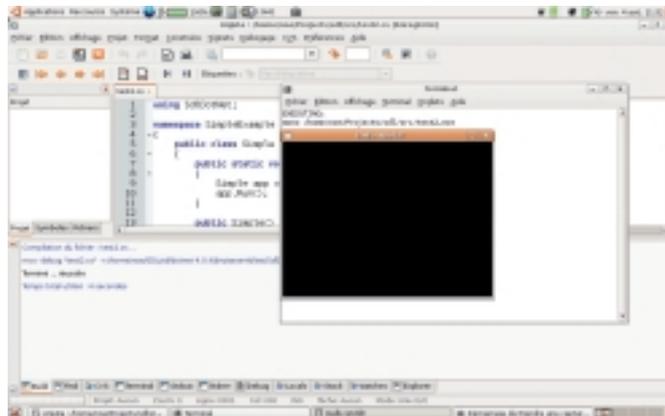
Pour lancer l'exécutable, il ne reste plus qu'à taper `mono test.exe` (attention, vous devez disposer dans votre chemin de toutes les bibliothèques nécessaires). Comment compiler et exécuter ce bout de code sous Anjuta ? Choisissez le menu Préférences/Commandes. A chaque langage vous pouvez associer des commandes spécifiques pour compiler et exécuter l'application. Indiquez comme chaîne de compilation `mcs -debug "$(current.file.name.ext)" -r:/home/xl/SDL/sdl_dotnet-4.0.4/bin/assemblies/SdlDotNet.dll`. Et dans le cadre de la commande d'exécution vous noterez `mono "$(current.full.file.name).exe`". La touche F9 compilera l'exemple et vous pourrez l'exécuter (touche F3). Cependant il faudra tuer le processus pour sortir de la fenêtre "Bonjour !". En effet, aucun événement n'est géré pour sortir de la boucle `Events.Run`. Nous allons donc en ajouter un, qui sera l'attente de l'appui d'une touche. La fonction callback `OnKeyDown` se chargera de tester la touche enfoncée. Et s'il s'avère que la touche `Escape` a été tapée, l'application quittera proprement :

```
using SdlDotNet;
namespace SimpleExample
{
    public class Simple
    {
        public static void Main()
        {
            Simple app = new Simple();
            app.Run();
        }

        public Simple()
        {
            Video.SetVideoModeWindow(400,300);
            Video.WindowCaption = "Bonjour !";
        }

        public void Run()
        {
            Events.KeyboardDown += new KeyboardEventHandler
(this.OnKeyboardDown);
            Events.Run();
        }

        private void OnKeyboardDown(object sender, Keyboard
EventArgs e)
        {
            if (e.Key == Key.Escape || e.Key == Key.Q)
            {
                Events.QuitApplication();
            }
        }
    }
}
```



L'évènement SDL "touche escape enfoncée" provoquera la fin de l'application.

4 Employer un dépôt CVS

Si vous désirez utiliser CVS conjointement à Anjuta vous devrez réaliser manuellement le premier commit. Anjuta intègre un menu CVS qui s'occupera du reste. La première opération consistera à initialiser la variable `cvsroot` (afin d'indiquer à CVS à quel endroit vous désirez créer votre dépôt) :

```
#mkdir /home/xl/cvsroot && export CVSROOT=/home/xl/cvsroot && cvs init
```

Ensuite, vous devez vous diriger vers le répertoire des sources. Par exemple dans le cas du projet `gtkmm` créé précédemment :

```
#cd ~/Projects/gtkmm/src
```

Puis effectuer le premier commit :

```
#cvs import -d loggtkmm xl initial
#cvs co loggtkmm
cvs checkout: Updating loggtkmm
U loggtkmm/Makefile
U loggtkmm/Makefile.am
U loggtkmm/Makefile.in
...
cvs checkout: Updating loggtkmm/.libs
```

Par la suite, vous pourrez mettre à jour votre dépôt ou par exemple consulter l'historique en utilisant directement le menu CVS intégré à Anjuta !

En conclusion

Anjuta 2 résulte de la fusion de `gIDE` et d'Anjuta 1.2. A l'heure actuelle, Anjuta 2 n'est pas encore assez stable pour y travailler quotidiennement, mais il s'agit clairement d'un projet à surveiller de très près. En effet, l'IDE s'est doté d'une architecture modulaire avec plug-ins et de nombreuses nouveautés ont vu le jour comme un éditeur de projet, une visionneuse de sources GTK, un plug-in `valgrind`, un gestionnaire de tâches, un plug-in de traçage d'un diagramme de l'héritage des classes, etc. Nous attendons avec impatience la version stable d'Anjuta 2.

Liste des références

Anjuta : <http://anjuta.org/>

gtkmm : <http://gtkmm.sourceforge.net/>

mono : http://www.mono-project.com/Main_Page

SdlDotNet : http://cs-sdl.sourceforge.net/index.php/Main_Page

■ **Xavier Leclercq** - Xavier.Leclercq@programmez.com

Utilisation d'un GPS en Visual Basic .NET pour Pocket PC

L'utilisation du GPS dans les applications mobiles est de nos jours plus qu'à la mode. Qu'à cela ne tienne, nous allons nous aussi intégrer des fonctions GPS dans nos applications mobiles en Visual Basic .NET pour Pocket PC.

Le GPS (Global Positioning System) est un système de localisation terrestre. A la fin des années 70, le principal souci du département de la défense des USA est de créer un système permettant à tous les éléments de l'armée américaine de se positionner de manière précise et rapide n'importe où à la surface de la terre. Le système GPS est composé d'une constellation de satellites (24 au total) positionnés de telle manière qu'au moins 3 d'entre eux soient visibles de n'importe quel point du globe. Chaque satellite émet un message codé donnant à ceux qui le reçoivent des informations sur sa position. Le récepteur GPS fait alors de savants calculs pour déterminer, grâce au décalage entre le moment et l'envoi et de la réception du signal, la position en termes de latitude, longitude et altitude (d'autres informations telles que la vitesse sont aussi calculées). Grâce à un ingénieux système de codage, les données sont très précises pour l'armée américaine, et beaucoup moins précises pour les civils. Nous devons nous contenter des données civiles, mais généralement ça suffit pour se positionner. La réception des données provenant des satellites et les calculs complexes qui doivent être réalisés pour déterminer la position sont effectués par des appareils spécialisés, de plus en plus bon marché, que sont les GPS.

Le protocole NMEA

Le GPS est un périphérique qui dialogue avec le reste du monde via une liaison série en utilisant le protocole NMEA (National Marine & Electronics Association). Ce protocole a été tout d'abord utilisé dans la marine et il est devenu au fil du temps la référence dans ce domaine. Le protocole NMEA est une suite de trames ASCII séparées par les caractères CRLF (comme dans un fichier texte). Chaque trame étant elle-même composée de champs séparés par des virgules. A la fin d'une trame, une somme de contrôle permet de vérifier que la trame reçue est valide et qu'elle peut être prise en compte. Voici un exemple de trame NMEA :

```
$GPRMC,225446,A,4916.45,N,12311.12,W,000.5,054.7,191194,020.3,E*68
```

Cette trame a pour identifiant \$GPRMC, ce qui permet de savoir quelles sont les informations que l'on va trouver ensuite et dans quel ordre. Cette trame est donc composée de 12 champs (en comptant \$GPRMC) et sa somme de contrôle est la valeur hexadécimale 68. La somme de contrôle est séparée du reste de la trame par l'étoile. Elle est calculée en effectuant la somme des caractères ASCII de la trame jusqu'à l'étoile. On extrait ensuite le modulo 255 de cette somme.

Il existe plusieurs type de trames (\$GPGGA, \$GPRMC, \$GPVTG, etc..) car chacune existe pour un type de données bien précis. La trame

\$GPVTG donne des informations sur le cap et la vitesse, alors que \$GPGGA sur les satellites disponibles et la précision des mesures.

La trame \$GPRMC

Le but est tout de même de connaître notre position. Pour cela, il existe la trame \$GPRMC qui donne la latitude, la longitude et la vitesse. Pour notre exemple ce sera parfait. La trame \$GPRMC est composée des informations suivantes :

```
00 $GPRMC
01 Heure du fix
02 Alerte (A=OK ; V=WARNIG)
03 Latitude au format ddmm.ss
04 Sens de la latitude (N=Nord=Positif, S=Sud=Négatif)
05 Longitude au format dddmm.ss
06 Sens de la longitude (E=Est=Positif, W=Ouest=Négatif)
07 Vitesse au sol en Knots (noeuds)
08 Cap vrai
09 Date du fix
10 Déclinaison magnétique
11 Sens de la déclinaison magnétique
```

Vous noterez que les champs 3 et 4 nous donnent la latitude, les champs 5 et 6 la longitude et le champ 7 la vitesse. Le champ 2 est là pour nous indiquer si les données fournies sont fiables ou non. Le GPS délivre généralement chaque seconde une série de trames dont la trame \$GPRMC, et ce, même s'il n'est pas encore dans la possibilité de donner des informations fiables (pas assez de satellites fixés par exemple). Cet indicateur permet de savoir à coup sûr si les données peuvent être utilisées ou non.

Le projet Visual Basic .NET

Afin d'illustrer toute cette théorie, nous allons créer un projet Windows Mobile 5.0 pour Pocket PC qui affichera les trames reçues du GPS connecté et traitera plus particulièrement la trame \$GPRMC. Le programme indiquera si les données reçues sont fiables et si oui, affichera la latitude, la longitude et la vitesse en temps réel.

Le contrôle SerialPort

Comme le GPS communique via liaison série, nous allons tout naturellement utiliser le contrôle System.IO.SerialPort qui est l'une des nouveautés du Compact Framework 2.0. Ce contrôle prend en charge la partie communication série. Il suffit de lui indiquer quel port utiliser, la vitesse, quelques paramètres concernant la forme du message série

(parité, nombre de bits de données, etc..) et il nous rend capable de lire et d'écrire via une liaison série. Dans notre exemple nous ne ferons que lire des données.

Et les GPS Bluetooth ?

Si vous utilisez un GPS connecté via Bluetooth à votre Pocket PC, pas de panique. Ce type de liaison Bluetooth est "vu" par les logiciels comme une liaison série. Les principes de lecture des données sont donc exactement les mêmes, à part que, dans ce cas précis, les notions telles que la vitesse, la parité, le nombre de bits de données n'ont pas d'importance, car c'est le protocole Bluetooth qui prend en charge ces détails, sinon tout est identique, sympa non ?

La classe CGPS

Vous trouverez dans le projet une classe CGPS qui est le cœur de cette application. Cette classe s'occupe de tout. Elle gère le port de communication, effectue la lecture des trames, vérifie qu'elles sont valides (la somme de contrôle) et dans le cas d'une trame \$GPRMC, effectue le découpage des données et crée un objet de type CGPS.InfoRMC qui porte toutes les informations décodées de ce type de trame. Cette classe communique avec le reste du monde par événements. Elle déclenche l'événement NewTrame à chaque nouvelle trame valide reçue, l'événement NewInfoRMC à chaque nouvelle trame \$GPRMC valide reçue, TimeoutOccurs en cas d'arrêt de la communication avec le GPS et enfin, un événement ExceptionOccurs en cas d'erreur interne rencontrée (pour le débogage principalement).

Elle possède une méthode Start pour lancer le moteur d'acquisition des trames et tout naturellement Stop pour l'arrêter. Il est possible de paramétrer la vitesse du port série, ainsi que le délai avant le déclenchement de l'événement TimeoutOccurs. Les paramètres de type parité, nombre de bits de données et nombre de bits de stop sont réglés en dur à l'initialisation de l'objet, car ce sont des valeurs fixées par le protocole NMEA. En cas de besoin il est très simple de rendre ces propriétés modifiables.

Le code de découpage des trames NMEA

La classe CGPS utilise un Timer interne pour la lecture des données en provenance de la liaison série. A chaque déclenchement du Timer, la classe lit l'intégralité du buffer d'entrée du contrôle SerialPort, puis s'occupe de découper chaque trame. Chaque trame est ensuite testée au niveau de sa somme de contrôle. Si la trame reçue est bonne, l'événement NewTrame est déclenché avec la trame en paramètre. Si la trame est de type \$GPRMC, elle est vérifiée, découpée, certains calculs sont effectués afin d'obtenir des données numériques utilisables pour la latitude et la longitude, et la vitesse est convertie en Km/h, l'événement NewInfoRMC est ensuite déclenché avec les données fraîchement récupérées en paramètre.

```
Private Sub pTimer_Tick(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles pTimer.Tick

    If pSerialPort.IsOpen Then

        If pLastReception = 0 Then
```

```
        pLastReception = System.Environment.TickCount

    Else

        Dim wInterval As Long = _
            System.Environment.TickCount - pLastReception

        If wInterval > pTimeout Then
            pLastReception = System.Environment.TickCount
            RaiseEvent TimeoutOccurs()
        End If

    End If

    Dim wTrame As String
    Dim k As Integer

    Try
        pBuffer &= pSerialPort.ReadExisting()
    Catch ex As Exception
        pTimer.Enabled = False
        pSerialPort.Close()
        RaiseEvent ExceptionOccurs(ex)
    Exit Sub
    End Try

    If pBuffer <> "" Then

        Do

            k = pBuffer.IndexOf(vbCrLf)
            If k = -1 Then Exit Do

            wTrame = pBuffer.Substring(0, k)
            pBuffer = pBuffer.Substring(k + 2)

            If CheckTrame(wTrame) Then

                pLastReception = _
                    System.Environment.TickCount

                RaiseEvent NewTrame(wTrame)

                Select Case wTrame.Substring(0, 6)

                    Case "GPRMC,"
                        Dim wInfoRMC As InfoRMC
                        Try
                            wInfoRMC = New InfoRMC(wTrame)
                        Catch
                        End Try
                        If wInfoRMC IsNot Nothing Then
                            RaiseEvent NewInfoRMC(wInfoRMC)
                        End If

                End Select

            End Select
```

```

End If
Loop
End If
End If
End Sub

```

C'est dans le code du constructeur de la classe CGPS.InfoRMC que la trame de type \$GPRMC est fournie. Si cette trame est correctement formée, les données sont placées dans les différentes propriétés de cette classe, sinon une exception est déclenchée et la création de l'objet échoue. C'est brutal comme méthode, mais ça marche.

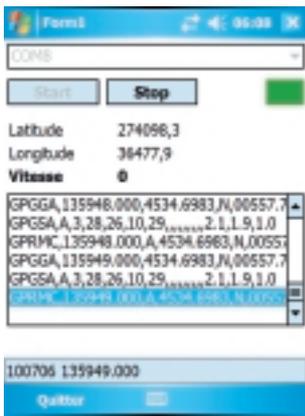


Figure 1 : Ecran de l'application

L'application

Nous avons donc un " moteur " permettant de récupérer les données d'un GPS. Dans cette version, ce moteur est seulement capable de réagir aux trames de type \$GPRMC, mais il est extrêmement simple d'ajouter les fonctions lui permettant de réagir aux autres trames, grâce à son architecture ouverte.

Quelques liens sur les GPS

Pour faire évoluer la classe CGPS vous allez avoir besoin d'informations sur les GPS, et le protocole NMEA. Je vous conseille le très bon site GpsPassion <http://www.gpspassion.com> qui propose beaucoup d'articles autour du GPS et du NMEA, ainsi qu'un article très complet sur le GPS en général que vous trouverez à cette adresse : <http://ditwww.epfl.ch/SIC/SA/publications/FI98/fi-5-98/5-98-page1.html>

En conclusion

Il est extrêmement simple d'utiliser un GPS, et Visual Studio 2005 nous donne tous les outils pour construire ce type d'application, que ce soit pour PC ou pour les machines mobiles. A titre d'information, le code de la classe CGPS fonctionne directement et sans modification sur un Smartphone Windows Mobile 5.0 et aussi sur PC.

Vous pouvez télécharger les sources de cet article à cette adresse : http://www.codeppc.com/dotnetcf2/gps1/sources_gps.zip



■ Stéphane Sibué

Responsable développement .NET et mobilité chez Antéis informatique (www.anteis.fr)

Microsoft MVP Windows Mobile Devices depuis 2003. Webmaster et Fondateur de CodePPC

(www.codeppc.com), la référence francophone du développement Windows Mobile.

PHP 5 avancé

Cyril Pierre de Geyer
Eric Daspet

Nouveautés 3ème édition :
- Les services Web
- PDO



recommandé par :



Le livre référence, déjà en librairie !

Développer un plug-in pour Netbeans 5.0

Après avoir pris contact le mois dernier avec les rudiments de l'écriture d'un plug-in pour Netbeans, nous abordons aujourd'hui des notions plus avancées afin d'aboutir à un plug-in totalement fonctionnel.

Le mois dernier, nous nous sommes surtout attachés à comprendre l'organisation interne de Netbeans, ce qui n'est pas une mince affaire, comme avec tous les environnements d'une telle envergure. Ainsi en va-t-il de même avec Eclipse. Aujourd'hui, nous allons réaliser un plug-in complet, configurable depuis les options de Netbeans et qui permettra l'écriture et l'exécution d'un script Jython au sein d'un projet Java, ce qui peut trouver son utilité par exemple dans l'écriture de petits scripts d'essais avant finalisation en Java. Pour que le lecteur profite au maximum du présent article, nous lui suggérons vivement de prendre connaissance de celui du mois dernier, car nous considérons ici que la terminologie propre à

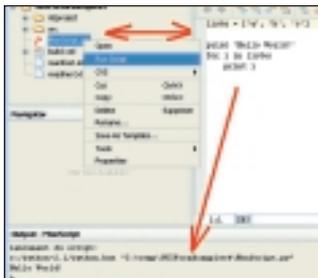


Fig : 1 - Notre plug-in de démonstration en action dans Netbeans.

Netbeans est acquise. Le plug-in que nous allons écrire n'est qu'une extension du mode de développement en Java et non un plug-in totalement dédié à Jython. (Fig : 1) Jython est un interpréteur Python Open Source écrit en Java, qui gagne à être connu. Vous pouvez vous le procurer à <http://www.jython.org>.

Le lecteur désireux d'aller plus

loin pourra étudier d'autres didacticiels à <http://platform.netbeans.org>. Il pourra aussi étudier avec grand profit des plug-ins complets tels que le plug-in Jython/Groovy du projet Coyote (<https://coyote.devjava.net>). Comme de coutume, les sources complètes de notre plug-in sont disponibles sur le Cd-Rom accompagnant le magazine ou sur notre site www.programmez.com

1 Associer un éditeur avec une extension de fichier

Commençons par créer un projet pour notre plug-in. C'est un simple projet de module Netbeans que nous baptisons PluginJython. La première chose à obtenir est que Netbeans identifie un script Jython en cours d'édition, ce qui permettra d'ajouter une action au menu contextuel du script pour lancer celui-ci. Netbeans identifie le contenu d'un document édité à partir de l'extension du nom de fichier physique, extension à laquelle est associée un type MIME unique que Netbeans utilise en interne. Dans l'explorateur, cliquez avec le bouton droit de la souris pour faire apparaître le menu contextuel. Sélectionnez 'File Type', ou, si cette entrée est absente, 'File/Folder', puis 'File Type'

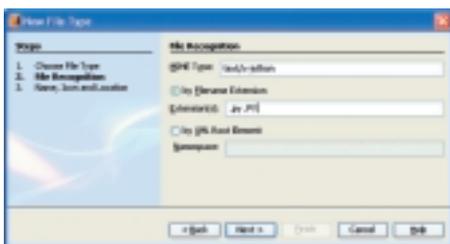


Fig : 2 - Création d'une association Type MIME/Extension.

sous 'Netbeans Module Development' dans la boîte de dialogue qui va apparaître. Dans la boîte de dialogue suivante donnez text/x-jython comme type MIME et .py .PY comme extension et

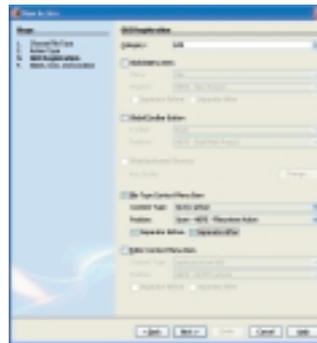


Fig : 3 - Création d'une action associée aux scripts Jython.

cochez le bouton radio 'By Filename Extension', comme illustré ci-contre. (Fig : 2) Dans la dernière boîte de dialogue, donnez Jython comme préfixe et donnez la localisation d'une icône 16x16. L'assistant génère alors un squelette du projet. Pour le lecteur qui souhaite approfondir le fonctionnement de Netbeans, il est très instructif de consulter le fichier layer.xml à ce stade. Dans notre projet nous

trouvons un fichier template.py qui est un modèle de script Jython. En effet, lorsque vous essayerez le plug-in, vous verrez que celui ci est intégré automatiquement à l'assistant 'New File' de Netbeans. Le modèle de script Jython est rangé sous l'arborescence 'Others'. Remplissez pour l'instant template.py avec le modèle de code Jython qui vous semble pertinent. Notre plug-in est opérationnel. Vous pouvez l'essayer tout de suite en pressant la touche F6. Dans la nouvelle instance de Netbeans qui va apparaître, créez d'abord un projet Java, et constatez que vous pouvez alors ajouter un script Jython à ce projet.

2 Associer une action aux scripts

Cette action aura pour but de lancer le script. Depuis le menu contextuel du projet, ajoutez cette fois une action. Dans la première boîte de dialogue, sélectionnez 'Conditionally Enabled' et renseignez le champ 'Cookies Classes' avec 'Editor Cookie'. Le dialogue suivant définit la catégorie de l'action. Donnez 'Edit'. Puis définissez comment doit apparaître l'action dans l'IDE. Choisissez 'File Type Context Menu Item' puis choisissez à votre convenance l'emplacement de l'entrée de menu de l'action. N'omettez pas de renseigner le champ 'Content-Type' avec le bon type MIME, à savoir text/x-jython. Votre dialogue doit ressembler à celui illustré ci-dessus (Fig : 3). Validez. Dans la classe Java de votre action, ajoutez ce code :

```
protected void performAction(Node[] activatedNodes) {
    JythonDataObject d = (JythonDataObject)
        activatedNodes[0].getCookie(JythonDataObject.class);
    FileObject f = d.getPrimaryFile();
    String filename = FileUtil.getFileDisplayName(f);
    NotifyDescriptor nd =
        new NotifyDescriptor.Message(
            "Vous êtes sur le point de lancer: \n"+ filename);
    DialogDisplayable.getDefault().notify(nd);

    Runtime rt = Runtime.getRuntime();
    try {
        String cmd = "jython.bat" + " " + "\"" + filename + "\"";
```

```

System.out.println(cmd);
rt.exec(cmd);
} catch (IOException ex) {
    ex.printStackTrace();
}
}
}

```

Ce code montre comment remonter au script à partir du noeud sélectionné par l'utilisateur dans l'explorateur. La méthode reçoit les noeuds actifs en argument. Le noeud sélectionné est toujours celui à l'indice zéro du tableau (l'API le garantit). A partir de là, on obtient le DataObject que le noeud ne fait que représenter, puis à partir du DataObject qui est un fichier virtuel, nous accédons au fichier physique. Le reste du code montre comment ouvrir une boîte de dialogue (vous devrez intégrer l'API Dialog à votre projet) et lancer le script comme processus externe à la JVM. Ce procédé ne vaut que pour cette première démonstration. Ainsi, les éventuelles sorties produites par le script ne sont pas redirigées dans Netbeans. C'est pourquoi, dans cette première démonstration (pluginjython sur le Cd-Rom ou à défaut sur notre site), le script Jython se contente d'ouvrir une fenêtre Swing. En outre, à ce stade nous supposons que Jython lui même est dans le PATH du système hôte. Nous allons maintenant affiner tous ces points.

3 Rendre le plug-in configurable

Par souci de clarté, nous travaillons à partir d'un nouveau projet pluginjython1 sur le Cd-Rom. Au départ, ce plug-in est le même que le précédent, hormis le corps de la méthode donnée ci-dessus, qui est vide. Nous souhaitons maintenant obtenir un plug-in plus évolué. Ainsi, nous voulons d'abord pouvoir faire pointer notre plug-in sur l'exécutable Jython, que celui-ci soit ou non dans le PATH du système. Notre but est donc d'intégrer une option de configuration aux options de configurations avancées de Netbeans comme illustré ci-contre (Fig : 4). Nous attirons l'attention du lecteur sur le fait que la méthode employée est valable au moment où nous écrivons ces lignes mais que l'équipe de Netbeans envisage de refondre cette partie de la plate-forme, dans un souci de performance notamment. Cette réserve faite, comment nous y prendre ? Nous ne disposons pas d'assistant pour ce travail. En revanche, nous savons que tout sous Netbeans correspond à une arborescence de fichiers virtuels. Nous devons donc commencer par nous brancher sur cette arborescence, ce qui n'est pas simple. Je n'ai pas trouvé de document décrivant complètement l'arborescence de Netbeans et en donnant une vision claire (Si vous en connaissez un, signalez le sur notre forum, je suis preneur ;). En tâtonnant et en étudiant

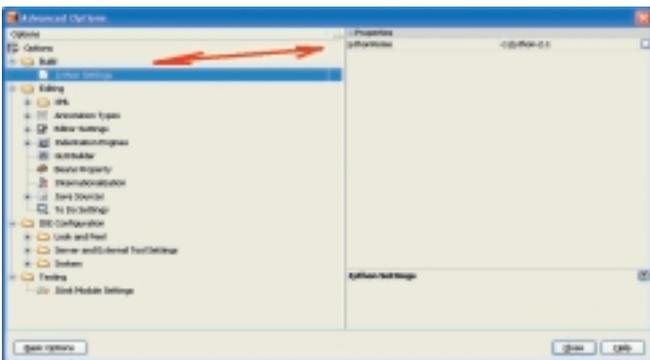


Fig : 4 - Notre plug-in est désormais intégré aux options de configurations de Netbeans.

d'autres plug-ins, on comprend qu'une branche UI du système concerne l'interface utilisateur et que sous cette branche la branche Services concerne les options, pour ce qui est de la représentation de la branche Services proprement dite. Nous ajoutons donc manuellement deux branchements dans le fichier layer.xml, un sur le service, un sur sa représentation comme ceci :

```

<folder name="Services">
  <file name="JythonModuleOption.settings"
    url="JythonModuleOption.settings">
  </file>
</folder>
<folder name="UI">
  <folder name="Services">
    <folder name="Build">
      <file name="JythonModuleOption.settings.shadow">
        <attr name="originalFile"
          stringvalue="Services/JythonModuleOption.settings" />
      </file>
    </folder>
  </folder>
</folder>

```

Notre service pointe sur un fichier à l'extension .settings alors que l'on aurait pu s'attendre à mettre une classe Java. En fait, le fichier .setting pointe à son tour sur une classe Java. Ce fichier intermédiaire a entre autres pour rôle d'éviter à Netbeans le chargement des classes tant que ce n'est pas absolument nécessaire. La branche UI/Services du code XML ci-dessus est en quelque sorte une réplique de la branche Services et est basée sur le .settings de celle-ci, comme l'indique l'extension settings.shadow (Vous avez le droit de prendre une aspirine avec votre café :). Voici maintenant le contenu du fichier .settings qui pointe enfin sur une classe Java :

```

<?xml version="1.0"?>
<!DOCTYPE settings PUBLIC "-
//NetBeans//DTD Session settings 1.0//EN" "http://www.netbeans.org
/dtds/sessionsettings-1_0.dtd">
<settings version="1.0">
  <module name=
    "com.programmez.fred.pluginjython1" spec="1.0"/>

  <instanceof class="org.openide.util.SharedClassObject"/>
  <instanceof class="org.openide.options.SystemOption"/>
  <instanceof
    class="com.programmez.fred.pluginjython1.JythonModule
Option"/>

  <instance class=
    "com.programmez.fred.pluginjython1.JythonModuleOption"
method="getDefault" />
</settings>

```

Ce fichier décrit la hiérarchie de notre classe JythonModuleOption qui dérive de SystemOption qui dérive de SharedClassObject. En plus de tout cela, nous fournissons un point d'entrée: la méthode GetDefault. Notre classe doit être écrite selon certaines règles. La première est qu'elle doit être un JavaBeans avec des accesseurs écrits selon les règles des

JavaBeans. Le constructeur par défaut n'est pas requis lorsqu'un point d'entrée est fourni comme nous l'avons fait. Contexte JavaBeans oblige, si l'on souhaite que notre classe affiche sa propre icône, on devra créer une classe BeanInfo à cet effet. Voici le code de notre classe :

```
package com.programmez.fred.pluginjython1;

import org.openide.options.SystemOption;
import org.openide.util.HelpCtx;

public class JythonModuleOption extends SystemOption {

    private static JythonModuleOption instance = null;

    public static final String
        PROPERTY_JYTHONHOME = "Jython Home";

    public static JythonModuleOption getDefault() {
        return (JythonModuleOption)
            findObject(JythonModuleOption.class, true);
    }

    protected void initialize() {
        setJythonHome("c:/jython-2.1");
        super.initialize();
    }

    public String displayName() {
        return "Jython Settings";
    }

    public HelpCtx getHelpCtx() {
        return HelpCtx.DEFAULT_HELP;
    }

    public String getJythonHome() {
        return (String) getProperty(PROPERTY_JYTHONHOME);
    }

    public void setJythonHome(String newhome) {
        putProperty(PROPERTY_JYTHONHOME, newhome, true);
    }
}
```

Nous voyons que la méthode `getDefault` a pour rôle de retourner une instance de la classe. Le procédé est particulier à Netbeans et se justifie notamment pour éviter des instances inutiles. Si `findObject` trouve une instance, celle-ci est utilisée, sinon l'instance est créée. Nous en avons terminé, notre option de configuration est maintenant opérationnelle. Essayez là et faites pointer Jython Home sur le répertoire de Jython dans votre système.

4 Rediriger les sorties et lancer le script

Il s'agit maintenant de rediriger les sorties produites par le script vers le panneau de sortie de Netbeans. Le mois dernier, nous avons expliqué comment obtenir une référence sur ce panneau. Sous

Netbeans une tâche n'est normalement pas lancée directement par le Runtime de la JVM comme nous l'avons fait dans la première partie de cet article, mais par une classe `ExecutionEngine`, qui s'attend à recevoir un objet de type `Runnable`. Et c'est seulement dans cet objet que l'on doit invoquer le Runtime de la JVM. Pour quelle raison est-ce ainsi ? Difficile à dire. Sans doute s'agit-il, parmi d'autres explications possibles, de donner un moyen à Netbeans de conserver des informations sur les processus lancés depuis l'IDE. `ExecutionEngine` ou pas, les règles de redirection des I/O d'un processus lancés par la JVM restent valables. Le lecteur voudra bien se reporter à l'article "Lancer des Processus depuis la Machine Virtuelle Java" de Programmez! 82, qui traite complètement de cette question. Rappelons seulement qu'afin d'éviter des débordements de tampons ou des deadlocks, nous devons créer deux threads, un qui lit les sorties du processus, l'autre qui écrit ces sorties dans la cible de la redirection. Voici le nouveau code de l'action, qui rend notre plug-in totalement opérationnel. Un dernier mot: pour pouvoir compiler les projets, vous devrez y intégrer les libraires Netbeans 'Settings Options API', 'I/O API' et 'Execution API'.

```
package com.programmez.fred.pluginjython1;

import java.io.BufferedReader;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import org.openide.cookies.EditorCookie;
import org.openide.execution.ExecutionEngine;
import org.openide.filesystems.FileObject;
import org.openide.filesystems.FileUtil;
import org.openide.nodes.Node;
import org.openide.util.HelpCtx;
import org.openide.util.NbBundle;
import org.openide.util.actions.CookieAction;
import org.openide.windows.IOPProvider;
import org.openide.windows.InputOutput;
import org.openide.windows.OutputWriter;

class StdHandler extends Thread {
    InputStream is;
    OutputWriter ow;

    StdHandler(InputStream is, OutputWriter ow) {
        this.is = is;
        this.ow = ow;
    }

    public void run() {
        try {
            InputStreamReader isr =
                new InputStreamReader(is);
            BufferedReader br =
                new BufferedReader(isr);
            String ligne=null;
```

```

        while ( (ligne = br.readLine()) != null)
            ow.println(ligne);
    } catch (IOException ioe) {
        ioe.printStackTrace();
    }
}

class ScriptRunnable implements Runnable {

private InputOutput io;
private String cmd;

    ScriptRunnable(String cmd, InputOutput io) {
        this.cmd = cmd;
        this.io = io;
    }

    public void run() {
        try {
            if (io != null) {
                OutputWriter owo = io.getOut();
                OutputWriter owe = io.getErr();
                owo.println("lancement du script:");
                owo.println(cmd);

                Runtime rt = Runtime.getRuntime();
                Process proc = rt.exec(cmd);

                StdHandler stdout =
                    new StdHandler(proc.getInputStream(), owo);
                StdHandler stderr = new
                    StdHandler(proc.getErrorStream(), owe);

                stdout.start();
                stderr.start();
                int exitVal = proc.waitFor();
            }
        } catch (Throwable t) {
            t.printStackTrace();
        }
    }
}

public final class RunScriptJython extends CookieAction {

protected void performAction(Node[] activatedNodes) {
    // Obtenir la localisation de Jython
    // dans les options de Netbeans
    String jython_home =
        JythonModuleOption.getDefault().getJythonHome();
    // Un travail soigné poserait un séparateur
    // en fonction de l'os hôte.
    String jython_cmd = jython_home + "/" + "jython.bat";
    File f = new File(jython_cmd);

```

```

// Jython absent physiquement ?
if(f==null || !f.exists())
    return;

// récupérer le nom de fichier du script
// à lancer
JythonDataObject d = (JythonDataObject)
    activatedNodes[0].getCookie(JythonDataObject.class);
FileObject fo = d.getPrimaryFile();
String filename = FileUtil.getFileDisplayName(fo);
// commande totale
String cmd = jython_cmd + " " + "\"" + filename + "\"";

// Accéder au panneau de sortie dans le "Mode" Output de Netbeans
InputOutput io =
    IOProvider.getDefault().getIO(d.getName(), false);
try {
    io.getOut().reset();
} catch (IOException exc) {
}

    ScriptRunnable sr = new ScriptRunnable(cmd, io);
    // C'est parti mon kiki! :-
    ExecutionEngine.getDefault().execute(d.getName(), sr, io);
}

protected int mode() {
    return CookieAction.MODE_EXACTLY_ONE;
}

public String getName() {
    return NbBundle.getMessage(RunScriptJython.class,
        "CTL_runScriptJython");
}

protected Class[] cookieClasses() {
    return new Class[] {
        EditorCookie.class
    };
}

protected void initialize() {
    super.initialize();
    putValue("noIconInMenu", Boolean.TRUE);
}

public HelpCtx getHelpCtx() {
    return HelpCtx.DEFAULT_HELP;
}

protected boolean asynchronous() {
    return false;
}
}

```

■ **Frédéric Mazué** - fmazue@programmez.com

A la découverte de Groovy

Groovy est un langage de script écrit en Java, pour l'univers Java. Regorgeant de trouvailles brillantes et de sucres syntaxiques savoureux, il mérite vraiment d'être découvert.

Quand un morceau de musique balance bien, qu'il donne de l'enthousiasme et envie de bouger, on dit qu'il "groove". Quand un langage est bien balancé et qu'il donne envie de programmer, il s'appelle Groovy. Groovy est un langage de script encore méconnu, car tout récent, puisqu'il a fait son apparition en l'an de grâce 2004. Il est vrai que les langages de script ne sont pas très populaires dans la sphère Java. Ainsi Jython (<http://www.jython.org> et Programmez! 66), bien que lui aussi tout rempli de qualités et plus ancien, ne perce pas réellement. Votre serveur pense que cela est dommage, car un langage de script apporte dans bien des situations rapidité d'écriture, concision et souplesse. Groovy est un langage de script interprété, dynamiquement typé et qui s'inspire de Python, de Ruby et évidemment de Java. Il emprunte aussi largement à la programmation fonctionnelle. Bien que tout jeune, il jouit déjà d'une belle maturité, dispose de nombreux modules d'extension et bénéficie de la sécurité apportée par la JVM ainsi que de l'immense panel de bibliothèques du monde Java (Fig. 1). Le découvrir totalement demanderait sans doute un ouvrage complet. Nous allons essayer de donner au lecteur un bon aperçu de ce qu'est Groovy, ainsi que l'envie d'aller plus loin avec ce langage.

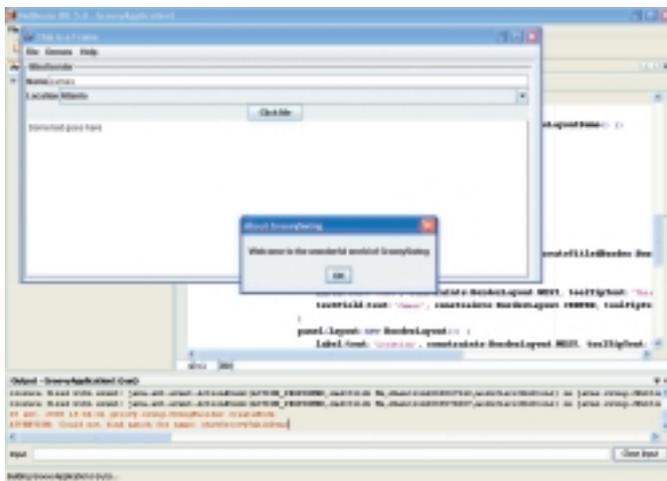


Fig : 1 - Démonstration swing/Groovy, prise sur le site officiel et lancée depuis Netbeans 5.0.

1 Téléchargement et installation

Groovy est Open Source et peut être téléchargé à <http://groovy.codehaus.org/>. Entièrement écrit en Java il vient sous la forme d'une archive qu'il suffit de décompresser dans le répertoire de votre choix sur votre système Windows, Linux ou Unix, pourvu que celui-ci dispose d'un JDK 1.4 ou supérieur. Après avoir désarchivé, vous devez créer ou mettre à jour trois variables d'environnement pour terminer l'installation :

- GROOVY_HOME doit pointer sur le répertoire de désarchivage de Groovy.
- Le PATH de votre système doit intégrer GROOVY_HOME/bin. (%GROOVY_HOME%\bin) sous Windows.
- JAVA_HOME doit pointer à la racine de votre JDK.

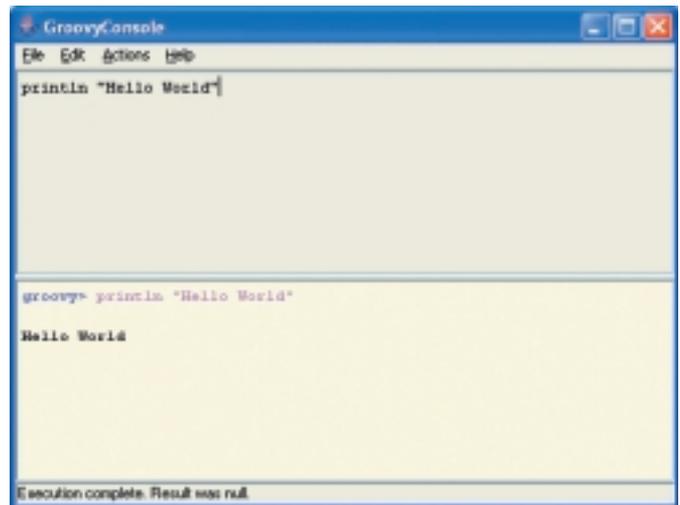


Fig 2 : La console de Groovy est un interpréteur interactif.

Le répertoire bin de Groovy contient des scripts .bat ou shell. Pour tester votre installation vous pouvez lancer un de ces scripts :

groovyConsole

Cette Groovy Console est une sorte d'interpréteur interactif. Saisissez une ligne de code Groovy dans la partie supérieure, validez par Ctrl-Enter. Le résultat de l'exécution apparaît dans la partie inférieure (Fig. 2).

2 Hello World en tous genres

Conformément à la tradition nous allons écrire un Hello World, et même plusieurs, car il existe différentes façons de procéder avec Groovy. Un script Groovy est un simple fichier texte portant l'extension .groovy. Par exemple, mon_script.groovy est une possibilité, tandis que mon-script.groovy serait illégal. Écrivons une ligne de code dans ce script :

```
println "Hello World"
```

Et lançons l'exécution par

```
groovy mon_script.groovy
```

ou simplement

```
groovy mon_script
```

Si n'importe quel éditeur de texte convient pour écrire du code Groovy, autant bénéficier des meilleurs outils. Il existe des plug-ins, encore en développement mais déjà utilisables, pour Eclipse 3.1 et Netbeans 5.0. Vous les trouverez à <http://groovy.codehaus.org/cvs-usage.html> et <https://coyote.dev.java.net> respectivement. Revenons maintenant à nos Hello Worlds. La syntaxe de Groovy est très inspirée de celle de Java et le langage permet la programmation objet. Les classes peuvent descendre de classes Java. Voici donc un Hello World de classe :-)

```
class HelloWorld {
    static void main(args) {
```

```

System.out.println("Hello World!")
}
}

```

Ce code est du Groovy, mais on jurerait du Java. Votre oeil d'expert a bien sûr remarqué tout de suite les différences. Les arguments reçus par main ne sont pas typés et il n'y a pas de point virgule à la ligne de code d'impression. Ce point virgule est optionnel. Les arguments sont automatiquement rangés dans une liste. Il est amusant de constater que si on écrit notre Hello World exactement comme en Java :

```

class HelloWorld {
    static void main(String[] args) {
        System.out.println("Hello World!");
    }
}

```

Groovy l'exécute quand même, ce qui ouvre des horizons pour ce qui est de prototyper rapidement du code Java. Mais se limiter exclusivement à la syntaxe de Java serait du gaspillage tant Groovy est riche. Finissons en d'abord avec nos Hello Worlds. En voici deux autres suffisamment parlant d'eux même pour se passer de commentaires :

```

class HelloWorld implements Runnable {
    void run() {
        System.out.println("Hello World!")
    }
}

```

mais oui, ce script va fonctionner :-)

Autre possibilité :

```

class HelloWorld {
    void go() {
        System.out.println("Hello World!")
    }
}

hw = new HelloWorld()
hw.go()

```

3 Des types de haut niveau

Les types de haut niveau, à savoir les listes et les dictionnaires, sont un des atouts maîtres de Python ou Jython. Groovy les reprend, le dictionnaire s'appelant ici Map sans doute pour plus de cohérence avec l'univers Java. Grâce au typage dynamique, une liste peut contenir n'importe quoi comme en Python. Par exemple :

```

liste = [1, "1", new String("Un")]
println liste
println liste[0]

```

La Map se manipule comme ceci :

```

ma_map = ["nom": "Fred", "age": 45]
println ma_map
println ma_map.age

```

Les clés sont les chaînes nom et âge. Ici aussi le typage dynamique permet de mettre n'importe quoi dans une Map. Voyons maintenant comment itérer dans les listes et les Maps :

```

for(item in liste)
{
    println item
}

for(key in ma_map.keySet())
{
    println ma_map[key]
}

```

ou encore

```

for(value in ma_map.values())
{
    println value
}

```

Aussi immédiates soient elles, les itérations ci-dessus peuvent encore gagner en élégance grâce aux closures que nous verrons plus loin. Pour l'instant, découvrons le range qui est une liste séquentielle dont la syntaxe s'apparente à ce que l'on trouve dans des langages fonctionnels comme Haskell.

```
mon_range = 1..10
```

construit une liste séquentielle de nombre entier de 1 à 10. Cette liste est dite inclusive car elle contient le dernier élément c'est-à-dire la valeur 10. La déclaration d'une liste exclusive est

```
mon_range_ex = 1..<10
```

Vous pouvez itérer dans le range pour le vérifier. Par exemple :

```
for(i in 1..<10)
    println i
```

La valeur d'incrémement de la séquence n'est pas obligatoirement 1

```
mon_range = (1..10).step(2)
```

Une liste séquentielle ne contient pas nécessairement des entiers. Ainsi

```
mon_range = 'a'..'z'
```

est correct. Il est commode de penser à un range comme à une liste séquentielle mais en fait, ce sont deux types bien distincts. Faire l'amalgame peut engendrer une erreur. Ainsi

```
mon_range = [1..10] // n'est pas un range
```

ne déclare pas le range escompté, mais une liste à un élément, cet élément étant le range. Nous laissons là la découverte des types de haut niveau de Groovy, mais nous encourageons très vivement le lecteur intéressé à consulter la partie relative au slicing dans le guide de l'utilisateur.

4 Les closures

C'est une des armes secrètes de Groovy. Le terme closure ou fermeture en français vient de la programmation fonctionnelle. Pour faire simple, une closure est un ensemble constitué d'une fonction et de son environnement lexical (portée). En programmation fonctionnelle, ceci fait que les fonctions sont des valeurs de première classe, ce qui signifie qu'une fonction peut être passée en argument à une autre fonction, ou être la valeur de retour d'une fonction. Dans le monde de

Groovy, on peut voir une closure comme quelque chose d'intermédiaire entre un simple bloc de code et une fonction anonyme comme on peut en trouver en Python. Définissons une closure

```
carre = {it*it}
```

par convention, il désigne l'unique paramètre reçu par défaut. Il est possible de passer plusieurs paramètres à une closure en employant des paramètres nommés. Nous avons donc pour l'instant un bloc de code qui élève une valeur au carré et qui équivaut à cette valeur ainsi que nous l'avons expliqué. Posons que l'on ait une liste d'entiers, on veut la liste des carrés de cette liste. Pour cela, on applique la closure à chaque élément de la liste, au moyen de la méthode collect de l'objet liste. Cela donne :

```
carres = [1, 2, 3].collect(carre)
println carres
```

ou encore, en plus concis

```
carres = [1, 2, 3].collect{it*it}
```

Nous avons vu plus haut comment itérer dans une liste au moyen d'une boucle for. Un grooviste averti aura le réflexe d'itérer avec une closure, comme ceci :

```
[1, 2, 3].each{println it}
```

Et on itérera dans une Map avec des paramètres nommés et avec une syntaxe légèrement différente:

```
ma_map.each{key, value -> println key + " : " + value}
```

qui imprime

```
nom : Fred
age : 45
```

Pour l'instant, Groovy ne supporte pas l'équivalent des InnerClasses de Java. On ne peut donc pas déclarer de classe telle qu'un ListenerAdapter. Au lieu de cela, on utilise des closures pour implémenter les méthodes. Voici un exemple de code pris sur le site officiel de Groovy; Si on a ce code en Java

```
Processor deviceProc = ...
deviceProc.addControllerListener(new ControllerListener() {
    public void controllerUpdate(ControllerEvent ce) {
        ...
    }
})
```

Il n'est pas possible d'avoir en Groovy l'équivalent de la classe ControllerListener. Pour implémenter la méthode controllerUpdate, on écrit alors cela :

```
deviceProc.controllerUpdate =
{ ce -> println
  "j'ai été appelé en recevant l'événement $ce" }
```

Le signe \$ est un sucre syntaxique qui permet d'intégrer un paramètre dans une chaîne de caractères.

5 GroovyBeans et Expandos

Un GroovyBean est un JavaBean bénéficiant d'une syntaxe simplifiée.

```
import java.util.Date

class Personne {
    // propriétés
    @Property Integer age
```

```
@Property String nom
@property Date date

static void main(args) {
    def personne = new Personne(age:46, nom:"Fred", date:new Date())
    println("Hello ${personne.nom}")
    personne.age = 45
    println("Age: ${personne.age}")
    println("Date: ${personne.date}")
}
```

Ce Bean définit trois propriétés qui sont publiques par défaut. Lorsque ce script est lu et compilé en byte-code par Groovy, des accesseurs sont automatiquement générés pour chaque propriété publique. On remarquera la syntaxe qui permet de définir les propriétés dans le constructeur, et on appréciera le gain en concision par rapport au code équivalent en Java. L'Expando est une sorte de JavaBeans dynamique. Il est possible de lui ajouter des propriétés à l'exécution. Un Expando n'est pas sans rappeler les objets JavaScript. Exemple avec une petite closure au passage.

```
personne = new Expando()
personne.nom = "Fred"
personne.age = 46
personne.hello =
{println "Hello, mon nom est $nom, mon age est $age" }

personne.hello()
personne.age = 45
personne.hello()
```

6 Le Groovy JDK

Il s'agit d'un ensemble de méthodes ajoutées à celle de la JDK pour la rendre plus "groovy" comme disent les auteurs du langage. Et on peut dire que ce n'est pas exagéré. Vous qui êtes de fins connaisseurs en Java vous savez quelle quantité de code il faut écrire pour imprimer toutes les lignes contenues dans un fichier texte. Avec Groovy, sa JDK, son itération dans les types de haut niveau et ses closures, l'affaire se règle avec une seule ligne de code!

```
new File("texte.txt").eachLine{println it}
```

7 En guise de conclusion

Nous espérons avoir donné au lecteur l'envie de découvrir plus avant Groovy. Nous avons vu qu'il est original, puissant, élégant, mais son intérêt va très au-delà du contenu de cet article qui ne présente que la face visible de l'iceberg. Groovy est réellement rempli de trouvailles, il propose aussi un compilateur pour générer du byte-code Java, il est possible d'embarquer l'interpréteur dans une application, il existe un module de programmation COM et quantités d'autres bonnes choses.

■ **Frédéric Mazué** - fmazue@programmez.com

Lexique

closure : Terme de programmation fonctionnelle. Une closure ou fermeture est l'ensemble constitué d'une fonction et de son environnement lexical.



Maintenant, acheter c'est aussi donner

Novembre en Enfance est une opération nationale de sensibilisation et de récolte de fonds au profit de 10 associations* qui oeuvrent quotidiennement pour le respect des Droits de l'Enfant. Durant tout le mois de novembre, vous pouvez agir en vous connectant sur www.novembre-en-enfance.org.

Pour en savoir plus, rendez-vous sur www.novembre-en-enfance.org

* Action Contre la Faim, Aide et Action, Association Contre la Prostitution des Enfants, Asmae, Bureau International Catholique de l'Enfance, Les Blouses Roses, Enfants du Monde - Droits de l'Homme, Handicap International, J'ai un Rêve et Sol En Si.



Une production



Sous le parrainage de



Avec la participation des entreprises



En partenariat avec les media



amBX : ambiance garantie

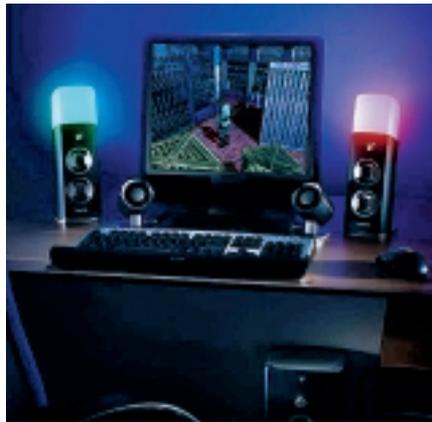


L'amBX, c'est le projet très ambitieux de Philips de proposer aux joueurs un environnement sensoriel total. Immersion garantie des pieds à la tête !

On y est déjà habitué pour l'environnement sonore (un bon surround-3D-7.1 transforme des jeux comme F.E.A.R. en une véritable expérience existentielle !) mais l'amBX élargit le champ sensitif aux variations de lumière, de luminosité, de température, de courants d'air, et même de vibrations. Le dispositif s'appuie évidemment sur de multiples périphériques dédiés, comme des enceintes avec LED multicolores, des lampes à variateur, de petits ventilateurs thermiques, ou encore un bracelet vibreur qui transmet par exemple aux chevilles les aspérités du sol sur lequel évolue le personnage. Pour le développeur, l'amBX repose sur un moteur dédié et un langage de scripts qui permettent de gérer tous les éléments d'ambiance. Ces "boîtiers actifs" peuvent en outre communiquer entre eux par WiFi ou Bluetooth.

L'amBX sur les rails

Concrètement, l'amBX a déjà été intégré au jeu d'aventures Broken Sword de THQ. Mais



sans apparemment éveiller beaucoup d'intérêt chez les joueurs... Cependant, Philips vient de signer un accord bien plus important avec le studio anglais Kuju Entertainment, déjà développeur de Train Simulator pour Microsoft. L'accord de développement amBX porte sur Rail Simulator, suite du précédent, dont la sortie est prévue en septembre 2007 : on y simulera des trains à vapeur, à moteur diesel

ou électrique, sur les principales voies ferrées allemandes et anglaises. L'amBX visera alors à reproduire totalement l'environnement ferroviaire, soit les vibrations du train, son contexte sonore, les variations de lumière selon les lieux traversés et l'heure du jour, et enfin... le déplacement d'air !

Faudra-t-il tout attacher dans la pièce, si on se fait un petit tour de TGV ?

L'amBX semble donc prendre dès maintenant réalité pour le développement (Philips fournit déjà le kit et les outils nécessaires aux studios volontaires), d'autant qu'il s'agit d'une plate-forme très accessible par son langage de scripts. Mais quid au niveau du joueur ? Le coût des différents périphériques de l'environnement amBX ne peut s'adresser qu'à des passionnés. Sur ce point, le choix de Rail Simulator devrait être judicieux pour lancer le concept, car on sait les amateurs de train prêts à toutes les folies pour étancher leur soif de "réalisme".

Site : www.ambx.com

Metromania : auto-copinage & Java



Auto-copinage, car l'auteur de cette rubrique est aussi l'éditeur d'un jeu de société, Metromania, disponible dans toutes les bonnes boutiques spécialisées et sur www.spiel-ou-face.com ! 2 à 4 joueurs s'y affrontent pour construire le meilleur réseau de métro dans des villes qui changent à chaque partie : c'est simple et fun, mais aussi stratégique, et plutôt sauvage... Le jeu connaît un succès certain, mais n'existe pour l'instant qu'en version jeu de plateau. Une version jouable en ligne serait ardemment souhaitée. Alors, si un programmeur Java a le courage de s'y lancer, il est le bienvenu... et bon courage avec les cases triangulaires ! Contact sur www.metromania.org.

Animation et modélisation faciales

LipSync MX est un utilitaire de synchronisation des lèvres, destiné avant tout aux animateurs Flash. Le logiciel est remarquable de simplicité, et synchronise animation et parole aussi bien en Français qu'en Anglais, Japonais, et Allemand. Sa dernière version (MX v2) élargit les fonctions d'importation et d'exportation, en particulier celles des fichiers SWF et MP3. On notera aussi chez cet éditeur très dynamique de middlewares une version désormais autonome de Facial Studio, auparavant simple



plug-in de 3ds Max. Un pur bijou pour modéliser rapidement des visages et des têtes en 3D à partir d'une librairie d'éléments graphiques prête à l'emploi. Site : www.di-o-matic.com

Vers un standard du "mobile gaming"

Les plus grandes compagnies de téléphonie et d'informatique mobile, de Nokia à Texas Instruments en passant par Samsung ou Symbian, viennent de se rallier à l'OMA, ou Open Mobile Alliance. Objectif : définir enfin un standard commun pour les jeux sur mobiles. Selon l'OMA, et en toute logique, la fragmentation des plates-formes freine le développement des jeux mobiles, leurs adaptations multiples faisant enfler les coûts de développement. L'Alliance planche donc déjà sur la définition de normes techniques minimales et communes, et multiplie les meetings (en Grèce en octobre, au Canada en décembre) pour rallier tous les acteurs du marché, développeurs comme fabricants.

Site : www.openmobilealliance.org

EN BREF

Le CAC 40 des jeux vidéo ?



BilGame : le nom du site est amusant, son principe aussi. Les utilisateurs vendent ou échangent leurs jeux vidéo, sur toutes machines, entre eux. Chacun fixe son prix, dans une monnaie virtuelle, les "Levels". Une cote de chaque jeu s'établit ainsi selon l'offre et la demande. Et les Levels gagnés par la vente peuvent être dépensés pour l'achat. C'est sympathique, utile, et ludique en même temps. On notera quand même qu'il faut payer au site un abonnement, à la durée ou au nombre d'échanges. Mais ce coût reste franchement modeste, en tout cas beaucoup plus avantageux que les taux usuraires pratiqués par certaines boutiques spécialisées dans les échanges...

Site : <http://www.bilgame.com>

Le français est joueur !

Une étude réalisée conjointement depuis les données du Panel Médiamétrie, du OUI (Observatoire des Usages Internet) et de la REM (Référence des Equipements Multimédia) donne des chiffres très intéressants, et souvent impressionnants, sur l'équipement et les habitudes multimédia des français.

L'ordinateur est présent dans 50,6% des foyers, un sur deux. Le pourcentage tombe à 28,1% pour les consoles de jeux, et à 14% pour les consoles portables. Les foyers équipés des 3 plates-formes sont évalués à 9,5%. Pour la pratique des jeux vidéo, près de 4 millions de français jouent à un jeu vidéo chaque jour ! C'est un public plutôt jeune, puisque 44% a entre 13 et 24 ans. Deux tranches d'âge se détachent cependant des statistiques par leur pourcentage

supérieur de "pratiquants" : les 13-19 ans et, surprise, les 35-49 ans !

Le jeu vidéo est pratiqué par 89% des joueurs à domicile, normal... Mais 5,6 % avouent s'y consacrer sur leur lieu de travail ! Sans doute des journalistes spécialisés dans les tests de jeux vidéo ?..

Site : <http://www.oui.net>

Cherchez l'erreur...

... et la fortune est à vous ! Plus sûr qu'un studio de développement, fondez une entreprise de tests de productions vidéo-ludiques : c'est la nouvelle voie à la mode dans l'industrie des jeux vidéo, que vient de suivre une vedette reconnue du genre, Andy Robson. Auparavant "Lead Test Manager" de Lionhead, le studio du mythique Peter



Molyneux (Populous, Black & White...), Robson vole désormais de ses propres ailes pour tous les développeurs/éditeurs avec Testology. La nouvelle société propose des batteries de tests en cours et/ou en fin de développement selon des forfaits et des procédures variées. Bref, histoire de chercher l'erreur... Tenté ? Ils recherchent du monde...

Site : <http://www.testology.co.uk>

Développement de carrière

En corrélation avec le très fréquenté Gamasutra, voici un nouveau site, GameCareerGuide, dont le titre est explicite... Il s'agit d'offrir aux développeurs débu-



tants comme chevrons un lieu unique regroupant toutes les news, outils, tutoriaux, offres d'emploi et possibilités de carrière du marché. Le site est en anglais et hélas très axé sur le monde anglo-saxon.

Cependant, ses guides pour démarcher les studios, déterminer son profil personnel, bien débiter dans une première équipe, réorienter sa carrière, sont clairs et pratiques. Pour le reste, le contenu semble encore incomplet, mais le site n'est même pas vieux de deux mois.

Il va d'ailleurs supporter la Game Career Fair, qui se tient à Londres au Café Royal, les 3 et 4 octobre prochain, durant l'habituel Longon Game Festival. Avec du beau monde parmi les gros éditeurs, d'Electronic Arts à THQ en passant par Eidos ou Blitz Games...

Site : <http://www.gamecareerfair.com>



L'interface dans la lucarne

Un ballon en guise de manette pour jouer à Pro Evolution Soccer ou

Fifa sur PS1 ou PS2 ? Non, vous ne rêvez pas ! Le dispositif ShootPad se compose d'un socle vert (simulant la pelouse) sur lequel est fixé le ballon-manette, qui peut rouler partiellement et être frappé sur place (il bouge un peu quand même, mais aucune crainte de casser un carreau).

Ce gadget indispensable sera disponible dès novembre.

Prévoyez aussi d'acheter un short.

Windows Vista révélé



- ▶ **Auteur :** collectif
- ▶ **Editeur :** Micro Application
- ▶ **Prix :** 17,30 €

Si les livres sur Vista demeurent encore rares, ils ne sauraient tarder à se multiplier. Celui-ci privilégie les utilisateurs et un peu les administrateurs. L'ensemble des nouveautés systèmes est abordé. Notons que .Net 3.0 est survolé en quelques lignes et WinFS est toujours présenté comme le futur système de fichier de Windows... La mise en page est claire et très visuelle.

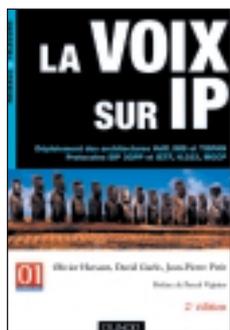
Internet danger : les arcanes de la toile

- ▶ **Auteur :** Thierry Gensollen
- ▶ **Editeur :** éditions Thierry Gensollen
- ▶ **Prix :** 23 €

La sécurité Internet constitue un sujet toujours d'actualité. Ce livre permet de réviser ses bases. On démarre par un petit historique de l'informatique et les fondamentaux de la sécurité informatique.

L'auteur redéfinit les cookies, les spams, les virus et l'ensemble des attaques et arnaques possibles. Sans révolutionner le genre, ce livre a l'avantage de redonner une idée générale de l'insécurité du web. À vous ensuite d'en tirer les conséquences.

La voix sur IP 2e édition



- ▶ **Auteur :** collectif
- ▶ **Editeur :** Dunod
- ▶ **Prix :** 69 €

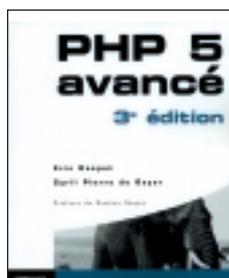
La VoIP se développe rapidement en entreprise et chez les particuliers. Le sujet demeure complexe et les techniques employées nécessitent une bonne

pratique. Cet ouvrage vous propose une plongée dans les principaux protocoles de la VoIP avec leur présentation, leur fonctionnement, l'encodage nécessaire.

Il manque juste IAX pour avoir une vue technique d'ensemble. Les auteurs n'ont pas oublié les aspects de sécurité, un point faible de la VoIP, même si un chapitre à part aurait été le bienvenu avec un panorama élargi. On lira avec intérêt les parties consacrées à la qua-

lité de la VoIP, un souci récurrent dans ces solutions. La partie sur les problématiques avancées apporte des réponses précises sur la redirection des appels et la traduction d'adresse. Un ouvrage de référence qui conviendra aux administrateurs et aux développeurs souhaitant maîtriser le sujet.

PHP 5 avancé 3e édition



- ▶ **Auteurs :** E. Daspet, C. P. de Geyer
- ▶ **Editeur :** Eyrolles
- ▶ **Prix :** 42 €

PHP 5 avancé revient dans une 3e édition mise à jour et augmentée. Tous

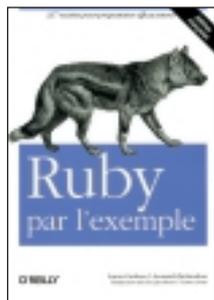
les aspects de PHP 5 sont abordés : architecture, configuration, structure de base, les données, les tableaux, les formulaires, etc. Les auteurs décrivent clairement chaque élément (fonction, commande, ou pratique) avec le code approprié. La sécurité n'est pas oubliée, ni l'interopérabilité avec les autres plates-formes. Un must pour le développeur PHP débutant ou expert !

Wi-Fi solutions de sécurisation

- Auteur :** Philippe Atelin
- Editeur :** Eni
- Prix :** 10 €

La sécurité des réseaux Wi-fi demeure un sujet sensible, que les utilisateurs oublient encore trop souvent. Ce petit livre vous propose une vue d'ensemble des options de sécurité Wi-fi. On débute par les mécanismes proposés dans le 802.11, les VPN, IPSec, le WEP, WPA ou encore EAP ainsi que le nouveau 802.11i. Très utile si on souhaite sécuriser rapidement son wi-fi !

Ruby par l'exemple



- ▶ **Auteur :** collectif
- ▶ **Editeur :** O'Reilly
- ▶ **Prix :** 48 €

Ruby constitue un des langages en pointe du Web 2. En pleine ascension, notamment grâce au framework Ruby on Rails, Ruby a la réputation d'être

d'un apprentissage " facile ". Mais pour bien l'exploiter, il faut connaître les bonnes pratiques, les astuces. O'Reilly vous propose de

découvrir tous les aspects ou presque de Ruby grâce à 337 recettes pour les développeurs. Comme toujours dans cette collection, les recettes sont segmentées en différentes thématiques (chaînes, tableaux, hachages, persistance, Ruby on Rails, interface...). Chaque recette commence par l'énoncée du problème, puis sa solution (avec le code adéquat). On termine souvent par une discussion pour aller plus loin ou préciser certains aspects de la solution.

Un petit regret, les aspects de sécurité trop peu présents. Une bible de référence pour tout développeur Ruby !

Les briques du web 3e édition



- ▶ **Auteur :** Jennifer Niederst Robbins
- ▶ **Editeur :** O'Reilly
- ▶ **Prix :** 52 €

Avec Web 2, l'évolution des langages, sites et applications web, il est nécessaire de se replonger

dans les fondations du web pour utiliser les bonnes techniques et technologies. Cet ouvrage a l'ambition de dresser un tableau, le plus complet possible, des standards, langages et technologies web.

Très orienté développeur, ce livre est un véritable pense-bête, même si le web 2 ne tient pas de place spécifique. L'auteur se concentre sur les standards et langages purement web, par contre, les langages Ruby, PHP, ASP, Ajax, etc. sont peu présents. La partie serveur aurait mérité une place plus développée. Il n'en demeure pas moins un excellent ouvrage pour tout développeur web.

Coffret ressources : Visual basic 2005 et SQL Server 2005

- ▶ **Auteurs :** T. Groussard, J. Gabillaud
- ▶ **Editeur :** Eni
- ▶ **Prix :** 49 €

Une idée originale ! L'éditeur Eni rassemble dans un coffret unique deux ouvrages .Net pour maîtriser l'accès aux données à partir de VB.Net et connaître le développement avec SQL Server 2005 et .Net. Cela permet d'avoir une vue d'ensemble des deux environnements et de la manière de connecter une base à une application .Net.

DÉVELOPPEZ 10 FOIS PLUS VITE

PLATEFORME PROFESSIONNELLE
DE DÉVELOPPEMENT

Windows, .Net, WebServices, Java, RAD

WINDEV est l'Atelier de Génie Logiciel préféré des développeurs professionnels en France. WINDEV était déjà le plus complet des environnements en standard.

Maintenant, en version 11 il sait même faire le café, grâce à ses nouvelles fonctions de programmation «Domotique» de pilotage des équipements électriques par courant porteur !

Les versions 11 proposent **11** nouveaux concepts, **111** nouveautés, **111** nouvelles fonctions du langage, **111** nouvelles fonctions en Java, **111** nouvelles fonctions en PHP...

Découvrez ces nouveautés, demandez le dossier gratuit accompagné d'un DVD et de 100 témoignages.

Livré en standard

- Gestion du cycle de vie
- Support de **Vista**
- Support de **Lotus Notes**
- Générateur automatique d'IHM avec héritage
- Base de Données gratuite HYPER FILE, Linux et Windows
- Bases tierces: Oracle, AS/400...
- 3-tiers
- Générateur d'états (PDF, Codes barres ...) librement distribuable
- Multilingue (20 langues)
- Hot Line gratuite
- RAD
- Langage L5G
- MultiThreads
- Profiler
- Refactoring
- Composants métier
- Gestion des règles métier
- Export automatique vers Word, Excel et Open Office
- Déploiement-1-clic
- XML •HTTP
- Liaison série et USB
- RFID •UML
- Multimedia
- Planning & Gestion de l'équipe
- Etc (demandez le dossier gratuit)



UN CODE UNIQUE:

Windows, .Net, Java, PHP, WebServices, J2EE, XML, Internet, Ajax, Pocket PC, SmartPhone, Client riche ...



...et maintenant
il fait même le café



Fournisseur Officiel de la
Préparation Olympique



**ENEZ DÉCOUVRIR
WINDEV 11 & WEBDEV 11
PRÈS DE CHEZ VOUS
NOVEMBRE 2006 : 12 VILLES**
inscrivez-vous sur www.pcsoft.fr (gratuit)

Version Express
en téléchargement

Demandez le dossier technique gratuit (100 pages en couleurs, en français) accompagné de 100 témoignages prestigieux et d'un DVD

WINDEV[®]
XI



_LE JOURNAL DE NOTRE INFRASTRUCTURE

_33^e JOUR : L'information est cloisonnée, ingérable. Les gens ne disposent pas de toutes les données nécessaires pour prendre leurs décisions. Gilles s'est mis en tête de permettre à tous d'y accéder, en même temps.

_Nous arrivons à 18 écrans par personne.

_36^e JOUR : C'est clair. Il nous faut le Middleware* IBM Information Management. L'information sera désenclavée et disponible à la demande, quel qu'en soit le format. Elle sera fiable et pertinente. Nous allons prendre de meilleures décisions et renforcer notre compétitivité.

_L'accès, c'est le succès !



Information Management

IBM.COM/TAKEBACKCONTROL/INFO/FR

*Logiciels d'infrastructure. IBM, le logo IBM et Take Back Control sont des marques d'International Business Machines Corporation aux États-Unis, dans d'autres pays ou les deux. © 2006 IBM Corporation. Tous droits réservés.

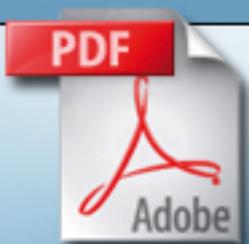
Lisez **Pro**grammez!

Le magazine du développeur

Chaque mois chez votre marchand de journaux

- les actus
- les dossiers
- le code

Achetez le numéro en vente actuellement
Trouvez le point de vente diffusant Programmez
A côté de chez vous



Achetez le numéro
en vente actuellement