Programmez !
www.programmez.com
Mensuel - Juillet/Août 2007 - N°99
LE MAGAZINE DU DÉVELOPPEMENT

Les bases de données de 2008

Microsoft® SQL Server™ 2008

MySQL 6.0
Le moteur Falcon

Créez et déployez un site

WEB 2.0

Ajax, JavaFX, Silverlight, Flash / Flex / AIR, Ruby on Rails

Tout pour choisir le bon langage, la bonne technologie.
Comment créer un site web 2.0 ?
Choisir son framework Ajax. Quelle sécurité ?
De la vidéo live dans son blog !

TÉLÉPHONE
Développez pour les Nokia s40

ROBOT
Programmez votre Tux Droid !

CODE

Mozilla
Programmation avancée avec XUL

Outil
Coder avec Emacs
PowerShell : le shell surpuissant !

Web
Struts ou JSF : comment choisir ?

Métier : programmeur de jeux

Printed in France - Imprimé en France -
BELGIQUE 6,45 € - SUISSE 12 FS -
LUXEMBOURG 6,45 € - Canada 8,95 $ CAN
DOM Surf 6,90 € - TOM 940 XPF - MAROC 50 DH
M 04319 - 99 - F: 5,95 € - RD

WINDEV WEBDEV WINDEV Mobile
PLATEFORME PROFESSIONNELLE
DE DÉVELOPPEMENT (AGL)
Windows, .Net, WebServices, Java, RAD
WINDEV
J'EN
RÊVE
WINDEV XI
UN AGL
COMPLET
EN STANDARD
Inclut en standard :
• IDE
• ALM
• Toutes les bases de données
• Merise
• UML
• RAD
• RAD Java
• langage de 5° génération (L5G)
• Windows
• .Net
• API
• DLL
• Générateur d'IHM
• Tests de non régression
• Gestionnaire de sources
• Installateur automatique
• WebServices
• XML
• SOA
• 3tiers
• Composants
• PDF
• Codesbarres
• Règles métier
• Lotus Notes natif
• Déploiement gratuit
• Générateur d'états
• Auto-formation livrée
• etc...
• En français
• Support Technique gratuit*
Fournisseur Officiel de la Préparation Olympique
Développez 10 fois plus vite
01 INFORMATIQUE
100 RÉUSSITES RÉVÉLÉES
WINDEV
WEBDEV
L'AVENIR
112 pages de témoignages 2007
Demandez le dossier technique gratuit (en couleurs, en français), accompagné de 112 pages de témoignages et d'un DVD. Version Express Gratuite.
Tél Province 04.67.032.032 Tél Paris 01.48.01.48.88 info@pcsoft.fr
WINDEV XI
www.pcsoft.fr

SOMMAIRE N°99

> Actus

L'actualité en bref 8
Agenda 8

> Evénements

Rsdc 2007 10

> SGBD

SQL Server 2008 : les nouveautés 14
MySQL 6.0, le moteur Falcon pour MySQL 18

> Projets

Les tests de recette avec Fitnesse 19

> Dossier Web 2.0 : choisir, créer, déployer !

Mashup : le concept à tout faire 25
Le téléphone 2.0 : une timide réalité ! 26
2007 : l'année du RIA 27
Les limites du Web 2.0 30
Evoluer vers le Web 2.0 : tout ou partie ? 32
La sécurité et le Web 2.0 34
Comment choisir son framework Ajax 36
Une webcam " live " pour mon blog 2.0 39
Silverlight 1.1 : créez vos applications riches et multimedia sur Internet 41
Créer et déployer son site ASP.Net Ajax 45
Création et déploiement d'une application Web 2 avec Rails 48
Des RIA en Java avec JavaFX Script 51

> Carrière

Epita ou Epitech ? 54

> Technique

Mettez du carburant dans votre mobile 58

> Développement Web

XulRunner & XPCom (2e partie) 63
Struts, JSF : lequel choisir ? 67

> Code

Programmez votre Tux Droid 71
(Re) découvrir Emacs 75
Manipulations avancées avec PowerShell 78

> Temps libre

Ludique 81

Donnez votre avis sur ce numéro
www.programmez.com/magazine_satisfaction.php

CD-Rom 99 PROGRAMMEZ !

Adobe Coldfusion 8 bêta
Dernière version bêta de l'outil de création d'applications Internet

.Net Framework 3.5 bêta 1
Dernière version bêta.

Silverlight 1.1 alpha
Permet de visionner des animations de type vectorielles ainsi que des contenus multimédias intégrant de l'audio et de la vidéo, à l'instar du Flash d'Adobe.

Expression Blend 1.0 *(60 jours)*
Concevez des interfaces interactives riches pour l'environnement Windows (anciennement " Interactive Designer ").

NetBeans 5.5.1
Eenvironnement de développement écrit en Java, et permettant d'écrire, compiler, déboguer et déployer des programmes.

QT Jambi
Versions : Windows & Linux - 32 bits. Binding java pour le framework graphique Qt. Il permet de développer des applications Java avec Qt en utilisant toutes les fonctionnalités de Qt.

Aptana
IDE Open source pour la création de sites web, orienté Javascript.

Tux Droid
Projet d'objet communiquant programmable, sous Linux. Avec synthèse de voix, en français

Emacs 22.1
Emacs est une famille d'éditeurs de texte disposant d'un ensemble extensible de fonctionnalités et qui est très populaire parmi les programmeurs.

Source des articles.

ERRATUM n°98 : SQL Server

Lionel Billon, Chef de produit SQL Server chez Microsoft France nous signale une erreur sur les fonctionnalités haute disponibilité de SQL Server, dans le dossier SGBD du mois dernier : *" Il est stipulé que nous supportons le mode actif/passif seulement alors que nous supportons également le mode actif/actif... Qui plus est, la haute disponibilité ne se résume pas au clustering, et sur le sujet nous avons dans notre version 2005 des fonctionnalités très intéressantes (opération de maintenance en ligne, mirroring, sur la thématique BI ReadOnly shared DB). "*

Pourquoi peindre avec les doigts ?

Visualisation Java pour clients riches et Ajax

ILOG JViews 8.0, la dernière version de la suite d'outils graphiques Java d'ILOG, couvre l'ensemble des fonctionnalités de visualisation avancée.

ILOG JViews 8.0 offre :

- Des composants graphiques puissants : diagrammes, courbes, tableaux de bord, cartographie, diagrammes de Gantt
- Des services évolués : agencement automatique de graphes, affichage performant pour des jeux de données volumineux
- Plusieurs techniques de déploiement : clients riches, applications Web interactives Ajax, Eclipse/RCP et portails
- Une expertise prouvée dans les industries les plus exigeantes : Informatique, télécoms, transport, énergie et défense.

Testez un de nos produits Java dès aujourd'hui http://jviews.ilog.com

© 2007 – ILOG S.A. – ILOG, CPLEX et leurs logos respectifs sont des marques déposées par ILOG.

ILOG®
Changing the rules of business™

NOUVEAU : l'emploi

Programmez!
LE MAGAZINE DU DÉVELOPPEMENT

Rédaction : redaction@programmez.com
Directeur de la Rédaction : Jean Kaminsky
Rédacteur en Chef : François Tonic
Ont collaboré : J.M. Maman, F. Mazué, R. Guy, P. Rouget, D. Seguy, C.Robert, M. Chaize, J-P. Riehl, A. Crepet, M. Isvy, C. Padych, G. Vidal, D. Magny, G. André, J. Chable, S.Haim-Lubczanski, G.Delamarre, D. Frontigny, D. Tizon, N. Blanco, J-P. Planas, Kysoh, N. Baron, M. Oubechou.
Dessin : Jissey
Maquette : AJE Conseils
Images : tous droits réservés
Publicité : Pour la publicité uniquement :
Tél. : 01 41 77 16 03
coordination@programmez.com
Editeur : Go-02 sarl, 6 rue Bezout - 75014 Paris Coordination@programmez.com - Dépôt légal : à parution - Commission paritaire : 0707K78366 - ISSN : 1627-0908 - Imprimeur : ETC - 76198 Yvetot
Directeur de la publication : Jean-Claude Vaudecrane
Le numéro comporte un CD Rom.

Abonnement : Programmez 22, rue René Boulanger, 75472 Paris Cedex 10 - abonnements.programmez@groupe-gli.com Tél. : 01 55 56 70 55 - Fax : 01 55 56 70 20 - du lundi au jeudi de 9h30 à 12h30 et de 13h30 à 17h00, le vendredi de 9h00 à 12h00 et de 14h00 à 16h30.
Tarifs abonnement (magazine seul) : 1 an - 11 numéros France métropolitaine : 45 € - Etudiant : 39 € - CEE et Suisse : 51,83 € Algérie, Maroc, Tunisie : 55,95 € Canada : 64,33 € Tom : 79,61 € - Dom : 62,84 € Autres pays : nous consulter.
PDF : 35 € (Monde Entier) souscription en ligne.

Édito

Au-delà de... quoi ?

Je vous propose pour cet été, une digression, en deux parties.

Premier point

Le " i " est comme une collection de haute couture. Un " aventurier " l'accole à ses produits et les autres suivent le chemin pour rester tendance, y'a pas à dire, l'originalité nous surprendra toujours. Il y a quelques années, c'était le " e " popularisé avec le " eCommerce " ou encore la " eAdministration ". Apple faisait sensation il y a dix ans avec le iMac et plus récemment avec son iPod. Désormais Google prend le train en marche avec son iGoogle et mine de rien, nous sommes passés du " eCommerce " au " iCommerce ". Et tout le monde n'y a vu que du feu. Mais que fait la police ? Verra-t-on un jour du iWindows, du iLinux ou… un iCerveau. Comme si, par ce " i " magique, on allait au-delà de quelque chose.
Personnellement, je lance la dInformatique, la dProgrammation, le dLogiciel, le dPortable, le dCommerce, avec bien entendu © partout… Comme quoi, le développement durable passera aussi par l'informatique. Mais à savoir si celui-ci sera équitable…

Deuxième point

Les éditeurs se mettent à nous coller du " beyond " à toutes les sauces*. Au-delà du réel, voilà un bien beau concept comme on n'aimerait pas en voir trop souvent. Dans les bases de données, on nous gratifie désormais d'un " au-delà du relationnel ". Sceptique ? Non, pourquoi donc. On nous promettait il y a plus de dix ans la base de données objet ! Résultat : le relationnel a gagné. On nous promettait il y a 2-3 ans, du XML partout écrasant tout. Résultat ? Le relationnel reste en place. Alors haranguer les gentils développeurs, DBA et DSI en promettant d'aller au-delà du relationnel, j'ai comme un léger doute. Et si pour le futur de la base de données, il s'agissait du SGBD sémantique ?
On parle de Web 2.0 partout, tout le temps. Programmez ! le fait régulièrement et notamment dans ce numéro d'été avec un grand dossier sur le sujet. Mais dès aujourd'hui, il y a concrètement un " au-delà " du web 2.0. Quelle que soit la numérotation qu'on lui donne, les plates-formes comme AIR, JavaFX, Silverlight 1.1 / WPF représentent une nouvelle génération, même si on habille une application desktop à la sauce web 2… Par contre, là où cela devient intéressant c'est la gestion plus fine des applications connectées, déconnectées. Car au-delà de l'interface, la question de la gestion des données, et de son fonctionnement se pose. D'ailleurs Google a décidé de prendre le problème à bras le corps avec Google Gears.
Et si on fusionnait les applications desktop et web 2.0 pour n'en faire qu'une, fonctionnant dans un navigateur en mode connecté / déconnecté ?

Et le fun ?

Avec l'été, les plages de claviers fins n'attendant que vous, nous vous proposons quelques sujets plus légers, plus funs. Programmer un Tux droid ou partir dans le merveilleux monde du développement mobile avec les terminaux Nokia. Et pour finir, si vous avez toujours rêvé de devenir développeur de jeux, nous vous donnons toutes les pistes indispensables…

François Tonic - *Rédacteur en chef*

*Après Intersystems: VMware, Adobe, Microsoft…

PROCHAIN NUMERO

N°100 - septembre - Parution : 31 août 2007

N°100

IN-DIS-PEN-SA-BLE !!!

Un numéro historique. Toutes les références du développement :
• Les 100 meilleurs logiciels • Les 100 termes à comprendre
• Les 25 gourous de l'histoire du soft
• Quel avenir pour le développeur ?

Dossier Web 2.0

2e partie

MARKETING SOFT

- FAUT-IL AMÉLIORER UN LOGICIEL NUL ?

- MAIS NON : IL SUFFIT SIMPLEMENT DE TROUVER DES UTILISATEURS PLUS STUPIDES !

Souriez avec Jissey
www.programmez.com
chaque semaine,
un gif animé

© 2007 Microsoft Corporation. Tous droits réservés. Microsoft, Visual Studio, le logo Visual Studio, et « Votre potentiel, notre passion. » sont des marques de Microsoft déposées et/ou utilisées aux États-Unis et/ou dans d'autres pays.

Microsoft®
Visual Studio®

Votre défi : Créer des applis riches et dynamiques pour PC ou mobiles.

Vos armes : Utilisez Visual Studio et Windows Vista pour créer de la valeur et pas seulement des données.

Plus d'informations sur **releveztouslesdefis.com**

TERMINAUX

La virtualisation s'attaque aux mobiles !

Une petite société française, Trango Virtual Processor, propose un hyperviseur adapté aux terminaux mobiles et à l'embarqué. Jusqu'à présent, il était fréquent sur un téléphone mobile, qu'il faille deux processeurs pour exécuter les piles et les applications. Avec Trango, il est possible de faire la même chose mais avec un seul processeur, grâce à l'utilisation d'un hyperviseur. Surtout, la petitesse de l'environnement permet de prendre uniquement de 20 à 30 Ko dans la mémoire, autorisant, selon les ressources, d'exécuter de 10 à 100 images virtuelles ! Les marchés visés sont : la téléphonie, les terminaux de paiement, l'embarqué, les set top box. Pour l'utilisateur, cela est transparent car Trango peut être totalement enfoui et l'utilisateur ne verra même pas que les applications fonctionnent sur des environnements virtualisés. L'éditeur fournit de la paravirtualisation, obligeant à une adaptation des systèmes virtualisés. Trango permet aussi de consolider la place de Linux sur l'embarqué, un marché encore difficile. Et surtout cela rassure les constructeurs et éditeurs. Car il est possible avec l'hyperviseur de faire tourner un Linux d'un côté et de garder les piles propriétaires / sensibles, sur une autre image virtuelle. Actuellement, Windows CE (mais pas Windows Mobile) et certains Linux ont été portés. Symbian ne l'a pas encore été, mais cela semble sur les rails. Tout dépend de la demande des clients. Sur le calendrier, Trango se colle aux annonces des fondeurs de processeurs. On attend à court terme les multi cœurs et le multi thread… Trango s'y prépare activement. Pour le développeur, l'éditeur propose des SDK et outils (dont un plug-in Eclipse) pour pouvoir adapter systèmes et applications à l'hyperviseur. Les développeurs peuvent alors utiliser le langage de leur choix.

OUTILS

Microsoft lance son IDE web 2 en ligne !

Voici une petite révolution que Microsoft vient de dévoiler en préversion : Popfly. Il s'agit d'un environnement en ligne pour développer des sites web 2.0. Il se divise en trois parties : un créateur de pages web entièrement personnalisable, un outil de création de mashup et enfin un module pour créer des communautés. L'outil mashup est sans doute le plus prometteur par son aspect tout graphique, doté d'une ergonomie plutôt agréable. On peut tout concevoir sans la moindre ligne de code, même si on accède immédiatement au code. De nombreux objets mashup sont disponibles par défaut. Un outil très puissant, même si une petite période d'adaptation s'avère indispensable pour maîtriser les liens et les paramètres des objets. Popfly se base sur la technologie silverlight. Si à sa sortie, l'environnement sera disponible sur Firefox, IE et Safari, seul le support IE est réellement opérationnel.

OUTILS

Web Performance Suite 3.4 cet été

Web Performance Suite est une suite logicielle dédiée aux tests de performance Web. L'outil simplifie le test de charge des sites les plus complexes, de JavaScript et AJAX aux frameworks dynamiques. Avec la version 3.4, la suite logicielle se voit enrichie d'un nouveau module : Advanced Server Analysis. La capacité du serveur Web étant connue, l'étape suivante consiste alors à accroître les performances. Advanced Server Analysis détermine les serveurs source de problèmes (à savoir le serveur Web, le serveur d'application, ou la base de données). Les indicateurs de performance clé sur le système sont analysés, et les problèmes automatiquement mis en évidence dans des rapports professionnels. Pas moins de vingt et une métriques du système d'exploitation sont surveillées durant un test de performance : utilisation CPU et mémoire, échec de connexion TCP/IP, etc. Le module Analysis Server Analysis est un module optionnel qui sera proposé à un prix de l'ordre de 2 500 Euros H.T. La version 3.4 de Web Performance sera disponible dès le début de l'été.

SOCIÉTÉ

Linagora & AliaSource : un nouveau leader est né

Les deux sociétés de services du monde libre / open source ont annoncé leur rapprochement pour mieux répondre aux demandes et devenir un " gros " du secteur. Le nouvel ensemble comptera ainsi une centaine de personnes et sera présenté à Paris, Lyon et Toulouse. Linagora souhaite créer un acteur majeur capable d'accompagner les grands projets et le déploiement de solutions libres en France mais aussi à l'étranger, avec l'ambition d'être présent en Belgique, Suisse, Angleterre et Allemagne. Pierre Baudracco, Directeur Exécutif d'AliaSource " Avec ce rapprochement, nous réunissons deux sociétés de premier plan pour former le leader incontesté du logiciel libre. Nos solutions accompagnées des offres d'industrialisation du Groupe Linagora franchissent une nouvelle étape dans leur développement. Ce rapprochement permet à nos solutions de passer un nouveau cap et offre une opportunité de participer à un projet ambitieux de création d'un véritable partenariat de référence sur lequel nos clients peuvent s'appuyer pour mener à bien leurs projets Libres. "

Agenda

JUILLET

Du 10 au 14 juillet 2007 à Amiens. **Les 8e Rencontres Mondiales du Logiciel Libre (RMLL).** http://www.rmll.info/

Le 11 juillet à l'IBM Forum de Paris - **Sécurité et Qualité : Venez tester IBM RATIONAL !** Séminaire Technique Gratuit. http://www-05.ibm.com/fr/events/qualite_logicielle.html

SEPTEMBRE

19 Septembre 2007 - Paris - **Salon emploi informaticiens et ingénieurs IT.** | http://www.kavesta.fr/informatique/index.htm

Du 25 au 29 septembre, Paris, Paris Expo, Hall 5 - Porte de Versailles - **APPLE EXPO 2007** - 24e édition du salon dédié au monde du Macintosh. - http://www.apple-expo.com/

Le 27 septembre 2007, Palais des Congrès Porte Maillot - **Salon Les Jeudis** – Emploi Informatique & Ingénierie - www.lesjeudis.com

Recevez la Newsletter www.programmez.com/inscription.php

ENVIRONNEMENT

Adobe lance Flex 3.0 et AIR

Fini Apollo ! Adobe a dévoilé le nom définitif de sa nouvelle plate-forme : AIR (Adobe Integrated Runtime) qui est disponible en version bêta. Fondamentalement, rien ne change réellement. Les applications AIR pourront être codées avec Flex, Ajax Flash, HTML, JavaScript puis déployées sur le desktop. Côté développeur, on dispose du AIR SDK, contenant les éléments de base pour démarrer ses projets AIR. Nouveauté importante : Flex 3.0 supporte AIR par défaut, et Flex 2.x n'est plus supporté ! Les développeurs sont donc obligés de passer à Flex 3.0 ! Pour les outils, une extension Dreamweaver est disponible. L'environnement Aptana est compatible et on peut aisément créer et gérer ses projets AIR. Il inclut un module de déploiement et intègre le SDK. De plus, côté Ajax, on peut utiliser divers frameworks (Ext JS, Prototype et script.aculo.us). Le runtime AIR (environ 9 Mo) s'installe à la première utilisation. Pour la partie HTML, Adobe se repose sur le moteur WebKit (issu d'Apple). Si MacOS X et Windows sont supportés, la version Linux du runtime arrivera un peu plus tard. La version 1.0 est toujours attendue pour le second semestre 2007. Pour le module de mise à jour, l'application AIR interrogera le serveur et téléchargera la mise à jour éventuelle.

Alors que Flex 2 a été lancé il y a à peine un an, Adobe propose déjà la bêta de Flex 3. Comme on l'a vu ci-dessus, ce lancement se couple à AIR et son modèle de développement. À cela se rajoute le SDK Flex en open source. Il sera totalement ouvert d'ici fin 2007. Adobe adresse aujourd'hui les applications riches de bureau avec AIR, le navigateur avec Flash / Flex et le mobile avec Flash Lite, avec un code commun sur le navigateur et le bureau. Aujourd'hui, Adobe met en avant LiveCycle Enterprise Suite. Il s'agit de proposer aux entreprises une plate-forme complète riche intégrant une gestion des processus métiers, une génération de PDF, la gestion des DRM, la signature des applications, des formulaires. Côté interface, on cible PDF ou Flex. On peut le comparer à InfoPath de Microsoft. Il s'agit de mettre en place des formulaires interactifs entre l'entreprise et le client.
Les prochains mois d'Adobe seront donc chargés pour répondre à Microsoft sur Silverlight / Expression et même à Sun avec JavaFX et surtout, mieux adresser les terminaux mobiles.

Site : http://labs.adobe.com/technologies/flex/

Perforce

Gestion *rapide* de configuration logicielle (GCL)

La fonctionalité Folder Diff,

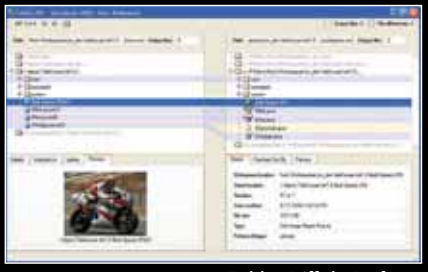
un atout de productivité du système de GCL Perforce.

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

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

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

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



Folder Diff de Perforce

PERFORCE
SOFTWARE

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

RSDC 2007
Rational mise sur le développement collaboratif

Début juin, à Orlando, près de 3000 développeurs et analystes se sont réunis pour suivre la 10e conférence des utilisateurs Rational. De nombreuses annonces ont ponctué l'événement de l'éditeur.

Comme l'a rappelé Scott Hebner (directeur stratégie et communication de Rational), le temps passe. Ce show à l'américaine doit être replacé dans son contexte : depuis maintenant près de 10 ans, IBM migre ses profits du matériel vers le logiciel et le service. Avant 3 ans, la moitié des profits doit provenir de ce secteur. Rational constitue de facto la pierre angulaire de l'édifice. IBM emploie pour le développement logiciel, 185 000 personnes dans 57 sites, avec des équipes dont la taille varie de 5 à 2 500 personnes utilisant de multiples méthodologies. Rien que pour cet usage, le développement de la division Rational se justifierait. Mais ces produits sont également utilisés par près de 11 000 développeurs à travers le monde, ce qui en fait l'une des communautés les plus dynamiques. Rational considère que le développement et l'animation de cette communauté font partie du cœur de sa stratégie pour les mois et années à venir.

Jamie Thomas présentant les nouveautés de Rational.

Une question d'image ?

La première surprise est venue du Dr Danny Sabbah (directeur général IBM Rational) en présentant à l'auditoire les points négatifs de la société ! Pour lui, " Vision sans exécution n'est qu'hallucination ". Rational est vu comme trop cher, trop compliqué et ne répondant pas correctement notamment aux demandes d'amélioration. En fait toutes les améliorations des produits existants doivent avoir pour objectif la correction de ces problèmes. Cela passe notamment par l'arrivée d'un client web très léger (client Ajax), l'amélioration des transactions, du nombre d'utilisateur par serveur, de la fiabilité, de l'utilisation mémoire ainsi que des performances et de la productivité des équipes. L'accent est mis sur le déploiement plus rapide à travers IBM Education Assistant et IBM Software Assistant.

Les premières conséquences de cette stratégie ont porté leurs fruits avec près de 12% de décroissance annuelle des problèmes remontés par les clients. Pour la réactivité, un investissement important a été fait dans la résolution des problèmes. Pour la qualité et l'amélioration de la qualité des lancements de nouveaux produits et nouvelles versions, l'organisation utilise des partenaires actifs pour la qualification des versions.

Le Dr Danny Sabbah.

Miser sur le collaboratif

Pour saisir tout l'intérêt de ces annonces, il faut comprendre l'importance de la notion " métier " dans les outils de développement. Depuis que le mot " logiciel " existe, l'industrie court après la génération automatique d'application. On spécifie, pousse (ou clique) un bouton, ou presque ! Ce rêve ne s'est jamais réalisé, mais régulièrement une nouvelle technologie, forcément révolutionnaire, est venue le ranimer. Ce fut en son temps la modélisation.

La complexité et l'hétérogénéité des systèmes est au-delà de la puissance pure des techniques de modélisation généralistes. A un certain moment, on est obligé d'incorporer des méthodes liées aux différentes industries. Dans le domaine des outils de développement, la pénétration d'un nouveau marché se traduit par l'utilisation d'outils adaptés à ce marché.

Première annonce forte, l'intention de rachat par IBM de l'éditeur suédois TELELOGIC. L'éditeur se spécialise dans la gestion du cycle de vie des applications, de l'architecture d'entreprise (notamment avec la SOA), la gestion des processus métiers, le développement embarqué, la gouvernance. Seconde annonce tout aussi importante, l'acquisition de Watchfire, un éditeur spécialisé dans la sécurité, la détection des failles, l'analyse préventive. Ce rachat doit renforcer l'offre Rational dans ces domaines.

Team Concert et Jazz

La troisième annonce a été détaillée par Jamie Thomas, VP développement : il s'agit d'un nouveau produit, Team Concert en version bêta, basé sur la technologie collaborative Jazz, pour la gestion des cycles de vie applicatifs destiné aux petites ou moyennes équipes utilisant les méthodologies "agiles". Cet outil sera intégré avec ClearCase et ClearQuest, par synchronisation bidirectionnelle. D'après les explications fournies par les représentants de Rational, cet outil permettra de tirer parti du meilleur de la stratégie de développement Open Source, c'est-à-dire des petites équipes, réparties dans le monde entier et collaborant sur un outil ou une architecture. Le dernier point était l'importance gran-

LEONARDI est une marque de LYRIA S.A. Les autres marques citées dans ce document sont des marques déposées par leurs propriétaires respectifs

L'ihM

EN TOUTE SIMPLICITÉ !

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

« il vous faut combien de temps pour la réaliser ? »

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

LEONARDI

ATELIER FORMATION 5/6 JUILLET 2007
300 €
REMBOURSABLES

LYRIA

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

dissante donnée par Rational à sa communauté de développeurs. Les clients sont désormais impliqués dès les débuts des développements, avec une prise en compte améliorée et accélérée des RFE (demandes d'amélioration). Puis dès que possible, des utilisateurs sélectionnés prennent en main des versions en cours de développement. Rational prend également en compte la diversité géographique de ses clients, et compte organiser près de 600 " roadshows " et formations.

On verra également le passage d'un portail éditeur/ industrie, à un portail orienté thématique (SPACES) pour réconcilier toutes les informations sur un sujet donné, comme les jeux par exemple. Bien sûr, les outils traditionnels (forums, blogs, wikis, newsletters) ne seront pas oubliés.

Que nous réserve le futur ?

Et l'avenir dans tout ça ? Nous avons été impressionnés par l'importance donnée à " Second Life " durant toute cette conférence. En effet, Rational considère les " communautés sociales " comme primordiales dans l'évolution ultérieure des systèmes. Rod Smith, VP technologies émergentes, dont le rôle est de tester les applications concrètes des nouvelles technologies auprès de clients sélectionnés, nous a éclairé sur l'utilisation pratique des technologies avancées de simulation virtuelle comme celles utilisées dans les jeux MMORPG (World of Warcraft, DAOC, LOTRO etc.…) pour les applications "sérieuses".

Des essais ont été faits, mais il manque encore 3 éléments : la standardisation des outils de modélisation, la précision des moteurs physiques pour les interactions et surtout un modèle plus tangible de retour sur investissement. Plus concrètement, nous devrions voir le support du parallélisme qui est en développement depuis longtemps, au niveau des langages et compilateurs. Ce devrait être une évolution visible dans les prochaines années. L'avenir verra également l'émergence ou la variation de nouveaux rôles dans le cycle de vie applicatif, ainsi que le support de technologies hétérogènes.

Gilbert Vidal
gilbert@gilbertvidal.com

Les outils Rational mis à jour

- ClearCase et ClearQuest passent en version 7.0.1 avec une amélioration de la sécurité par implémentation de Listes Contrôles d'Accès utilisateurs au niveau des dossiers.
- Un nouveau produit Rational Asset Manager version 7.0.1 pour la gestion des metadatas des ASSETS ; le dictionnaire associé utilisera DB2, et l'utilisation permettra la gouvernance des services dans un environnement SOA. Une intégration à ClearQuest est possible.
- R.P.M. disposera d'une interface Web ainsi que d'un générateur de rapports amélioré. Une interface sera possible avec les ERP pour les informations concernant les coûts ou les ressources humaines.
- Method Composer disposera d'une intégration avec RPM au niveau des templates et d'une bibliothèque de processus améliorés avec plus de contenus sur SOA.
- BuildForge voit arriver un générateur de rapport amélioré ainsi qu'une généralisation de l'internationalisation (notamment en Français).

Ces versions arriveront progressivement à partir du 29 juin.

Licence vs. Philosophie Open Source

Il a beaucoup été question pendant la conférence RSDC 2007 de la philosophie Open Source en matière de développement. Mais qu'en est-il exactement concernant les nouveaux produits ? Soyons clair, les produits basés sur Jazz, ainsi que les nouveaux outils de développement collaboratifs sont et seront des produits commerciaux, même s'ils incluent des produits Open Source tels Apache ou Eclipse. Il faudra donc les payer en monnaies sonnantes et trébuchantes. Mais IBM Rational souhaite que ses produits soient adaptés aux besoins et avec une qualité maximum dès la release. Ils encouragent donc les développeurs à télécharger les version bêta pré-commerciales, avec leur code source, et à participer à la mise au point. Les modifications proposées, si adoptées, seront donc récupérées par IBM. John Kellerman responsable Rational Jazz et Eclipse nous a indiqué qu'il était de l'intérêt de tous d'obtenir un produit final le plus fiable et le plus en adéquation possible avec les besoins des utilisateurs.

SGBD

Informix évolue de nouveau

Informix, une des bases de donnée historiques, vient d'être mise à jour par IBM avec le projet Cheetah (ou version 11) d'Informix Dynamic Server. Cette version est taillée par les centres de données en cluster avec reprise d'activité après une panne et haute disponibilité. Cette mouture doit être capable de gérer les risques d'énormes volumes sur mainframe. Cheetah reprend le contrôle d'accès avancé de DB2 v9. Pour le développeur, on dispose d'une nouvelle API de monitoring permettant d'intégrer un moniteur de contrôle dans son application. À cela, cette version d' Informatix rajoute une nouvelle interface, un meilleur support du XML. Elle sera disponible début juillet.

Recevez la Newsletter www.programmez.com/inscription.php

APPLICATIONS

Google lance Google Gears

Google vient de lancer Google Gears. Il s'agit d'une extension Firefox et Internet Explorer pour pouvoir créer des applications capables de fonctionner en mode déconnecté. Pour ce faire, Google Gears propose trois fonctions majeures : un serveur local pour assurer le cache et la disponibilité des ressources de l'application sans devoir se connecter au serveur en ligne, une base de données pour sauvegarder et accéder aux données et enfin un thread permettant aux applications de fonctionner en arrière plan avec de bonnes performances et une bonne réactivité. Il s'agit pour le moment d'une pré-version. Pour pouvoir créer et adapter ces applications Web, Google rend disponibles des API accessibles en javascript. On a accès aux fonctions décrites ci-dessus : LocalServer, Database et WorkerPool. Gears utilise la licence BSD. Une version Safari est en cours de développement. À n'en pas douter, Google Gears va rapidement devenir un " standard " pour les applications web connectées/ déconnectées.

FORMAT

OpenXML disponible en Java !

Initié par la société Wygwam, experte en technologie Microsoft et en particulier .Net, le projet OpenXML4J (licence BSD / Apache 2) permet de manipuler OpenXML en Java. Le développeur Java dispose donc d'une implémentation native, permettant d'embarquer dans son code (desktop ou serveur) les éléments pour générer, manipuler des fichiers OpenXML. Actuellement, OpenXML4J fonctionne en Java 1.5, une compatibilité 1.4 est envisagée mais non réalisée à ce jour. Eclipse est l'outil de développement conseillé, même si NetBeans est utilisable en important l'API dans son projet. Un script Ant facilite les builds. Le projet cherche des contributeurs pouvant aider à implémenter de nouvelles fonctions ainsi que des testeurs. Site : http://www.openxml4j.org

RUBY

CodeGear adapte Rails

Petite révolution : CodeGear, la filiale de Borland s'occupant des outils de développement, va sortir un environnement intégré de développement entièrement dédié à Ruby on Rails ! Il y a quelques mois, l'éditeur avait investi le monde PHP. Ruby IDE doit améliorer le développement avec Rails et Ruby en général.

Rails étant devenu un framework de plus en plus en pointe dans le monde Web 2 et plutôt bien accueilli par les développeurs, il manque encore une panoplie d'outils performants et surtout dotés d'une interface de développement pour construire rapidement les applications.

Cet outil supportera totalement Ruby on Rails, avec la complétion de code, des fonctions de navigation dans le code, des assistants, un gestionnaire de projet, des modules de déploiement, un éditeur de dépendance.

On ne devrait donc plus uniquement coder à la main… L'outil devrait être disponible en version finale plus tard dans l'année.

Soyez prêts pour les nouvelles architectures SOA!

Quint&cens • 4096

SOA est devenu en peu de temps le mot clé des développements logiciels.

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

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

SOFTEAM Formation 2007 : NOS BEST SELLERS

	UML pour la maîtrise d'ouvrage	2 j
	Analyse et conception avec UML	4 j
SOA	Méthodologie pour SOA	2 j
SOA	Architecture SOA	2 j
SOA	Développement de Web Services en Java	3 j
SOA	Développement de Web Services en C#	3 j
	Architecture et intégration : la synthèse	3 j
	Programmation orientée objet avec Java	4 j
	Développement d'applications JEE 5	5 j
	Développement d'applications JEE 5 Front End	4 j
	Développement d'applications JEE 5 Back End avec EJB 3	3 j
	Maîtrise du framework (Struts / JSF / Spring)	3 j
	Développement d'applications .NET / C#	4 j

Calendrier complet et inscriptions en ligne
www.softeam.fr

Rennes · PARIS · Nantes · Sophia

Tél. : 01 53 96 84 00 - Fax : 01 53 96 84 01
Paris : 21 avenue Victor Hugo - 75016
Rennes - Nantes - Sophia Antipolis

SOFTEAM Think Object

Microsoft® SQL Server™ 2008 les nouveautés

Microsoft vient de lever le voile sur le futur de son serveur de base de données (nom de code Katmai). Son nom est désormais officiel, il s'agira de SQL Server 2008. Microsoft a ainsi décidé de rompre avec les cycles de développement classique et de séparer la sortie de versions, du développement des fonctionnalités du moteur. On devrait dès lors voir apparaître une version tous les 2 à 3 ans, incluant les fonctionnalités qui ont été développées et testées à date.

Cette nouvelle coïncide avec l'arrivée de la CTP (Community Technology Preview) de SQL Server 2008. La disponibilité de celle-ci au public montre une volonté forte de Microsoft de mettre à disposition les fonctionnalités du moteur au fur et à mesure de leurs disponibilités pour avoir du feed-back. La CTP actuelle est celle de juin (aussi connue sous le nom de CTP 3), accessible via Connect (http://connect.microsoft.com/sqlserver), les versions devraient se suivre à des rythmes de l'ordre de 2 mois. Passons en revue les fonctionnalités clés du point de vue du développeur, sachant que toutes ne sont pas encore dans cette CTP de juin.

Transact-SQL simplifié

Le langage de SQL Server (T-SQL / Transact-SQL) est procédural, il est lui-même une extension du langage standardisé qu'est le SQL.
Rien à voir donc avec les langages objets actuels, cependant quelques améliorations tirées de ces derniers sont ponctuellement injectées dans le T-SQL.
Certaines lourdeurs au niveau du Transact-SQL ont été revues, au niveau des déclarations et des incréments de valeurs vous pourrez désormais écrire :

```
-- SQL Server 2005 et avant
DECLARE @var int
SET @var = 1

SET @var = @var + 1

-- SQL Server 2008
DECLARE @var int = 1

SET @var += 1
```

Autre reproche fréquent au niveau de l'insertion, l'impossibilité de fournir plus d'une ligne avec la clause VALUES, ce qui a pour effet de faire grossir les scripts générés par des outils de génération de scripts tels que l'assistant d'export de base de données (fourni dans le SQL Server Hosting Toolkit qui est gratuit), ou Visual Studio Team Edition for Database Professionals.
La nouvelle syntaxe permet de fournir autant de lignes que vous le souhaitez dans l'INSERT / VALUES.

```
INSERT INTO MaTable (ChampA, ChampB, ChampC)
VALUES (1, 2, 3), (4, 5, 6), (7, 8, 9)
```

Dernier apport au niveau de la syntaxe, celui du MERGE défini par le standard SQL. Il permet de combiner les 3 opérations de modification de données que sont le INSERT / UPDATE / DELETE en une instruction qui en fonction d'une condition appliquera l'une des opérations à la destination des données par rapport à la source. Exemple : vous avez importé un fichier d'inventaire dans votre base de données tous les jours, vous souhaitez supprimer les enregistrements correspondant aux articles passés à 0, mettre à jour ceux dont le nombre a changé et non null, et ajouter ceux qui n'existaient pas encore. Dorénavant cela se fait en une commande :

```
MERGE MaTable AS dest
USING (SELECT ProduitID, Quantite FROM Inventaire) AS src
ON (src.ProduitID = dest.ProduitID)
WHEN MATCHED AND (src.Quantite != 0) THEN
    UPDATE SET dest.Quantite = src.Quantite
WHEN MATCHED AND (src.Quantite = 0) THEN
    DELETE
WHEN NOT MATCHED THEN
    INSERT (ProduitID, Quantite)
    VALUES (src.ProduitID, src.Quantite )
```

L'intérêt est énorme au niveau de mises à jours incrémentielles / différentielles faites par exemple dans un DataWarehouse ou tout autre type d'application faisant ce genre de traitement.

Donnez votre avis sur ce numéro www.programmez.com/magazine_satisfaction.php

Nouveaux Intel® C++ et Fortran compiler 10.0 Professional Editions

Offrent le meilleur support pour créer des applications multi-thread pour Windows*, Linux* ou Mac OS X. Seule la version professionnelle des compilateurs Intel offre l'étendue des fonctionnalités d'optimisation avancée et de multi-threading qui incluent la vectorisation, l'auto-parallélisme, l'OpenMP*, le data prefetching, le loop unrolling et threading building block, des librairies multimédia et de traitement mathématique.

Intel Compiler 10.0 Professional édition est livré avec des librairies qui vous permettent de réaliser des threads comme un expert dès votre premier jour d'utilisation. Les APIs sont constamment mises à jour pour utiliser automatiquement les fonctionnalités des nouveaux processeurs

"'Dans une dizaine d'années, un programmeur qui ne pensera pas d'abord "parallèle" ne sera pas un programmeur'"

James Reinders
Chief Software Evangelist
des produits Intel Software

Les compilateurs et les librairies sont pré-validés pour fonctionner ensemble :

- **Intel® C++ and Fortran Compilers** parallélisent automatiquement votre code et l'optimisent pour la performance afin de tirer profit des processeurs multi-core avec un minimum d'efforts.
- **Intel® Math Kernel Library 9.1** vous fournissent les fonctions mathématiques multi-thread qui surpassent le code compilé individuel et les autres librairies.
- **Intel® Integrated Performance Primitives 5.2** (seulement C++) sont des fonctions parallèles hautement optimisées qui accélèrent le développement pour les médias, la cartographie et le traitement de signal.
- **Intel® Threading Building Blocks 1.1** (seulement C++) consiste en de multiples routines testées et optimisées pour simplifier le développement d'applications thread robuste et "scalable".

SOS DEVELOPERS

Intel C++ Compiler 10 Standard Windows ou Linux : 330 €HT
Intel C++ Compiler 10 Professional Windows ou Linux : 440 €HT
Intel Visual Fortran Compiler 10 Standard Windows ou Linux : 440 €HT
Intel Visual Fortran Compiler 10 Professional Windows ou Linux : 515 €HT
Intel Math Kernel Library for Windows ou Linux : 295 €HT
Intel Integrated Performance Primitives for Windows ou Linux : 147 €HT
Intel Threading Building Blocks for Windows ou Linux : 220 €HT

Plus d'informations sur : **www.sosdevelopers.com/intel.htm**
ou au 0825 07 06 07

Copyright* 2007 Intel Corporation. Intel, le logo Intel, Intel. le logo Leap ahead sont des marques déposées d'Intel Corporation ou de ses filiales aux Etats-Unis et autres pays. Tous droits réservés.

Avec .Net… SQL Server 2008 et Visual Studio 2008

Comme le T-SQL n'est pas objet et ne sait pas bien gérer les traitements très complexes, c'est tout naturellement que nous nous retournerons vers .Net pour nos développements de ce type. Intégré depuis SQL Server 2005 avec le Framework en version 2.0, .Net bénéficie de quelques plus dans cette version. Ainsi voyons nous le Framework version 3.0 arriver avec ses principes de Workflow (WF) et de communication unifié (WCF) entre autres. L'intérêt est évident, puisqu'il devient possible de gérer des traitements nécessitant des Workflows de manière centrale dans le serveur de base de données. Pour ceux connaissant le principe, une maigre tentative avait été faite sous SQL Server 2000 dans son Ressource Kit.

Est-ce que Visual Studio 2008 exploitera tout de ce qui est dans Visual Studio ? La réponse est non, un élément de cette réponse vient du fait que le Framework supporté par SQL Server 2008 est le 3.0 et non pas le 3.5. Cependant, SQL Server 2008 devrait être le moteur de prédilection de Visual Studio 2008. En effet un grand nombre de nouveautés de ce dernier devraient être optimisées dans le premier.

Pour ne citer que lui, LINQ bénéficiera d'une intégration bien plus native avec SQL Server 2008 qui offrira des écritures de requêtes beaucoup plus légères au niveau serveur que ce qui est le cas actuellement avec SQL Server 2005. La couche LINQ to Entities bénéficiera, elle aussi de tout un lot d'améliorations tirées de SQL Server 2008.

La partie d'Ado.Net intitulée " Occasionally Connected Systems " bénéficie d'avancées dans SQL Server 2008. Les commandes MERGE entre autres, des fonctions supplémentaires pour le suivi des champs timestamp, ainsi que le suivi de modifications de données réalisé sans trigger (Change Data Capture) vont dans ce sens pour simplifier la vie des applications déconnectées.

Nouveaux types

Chaque version, voit arriver son lot de nouveaux types, dans cette version ils sont nombreux. Cela va des nouveaux types date au traitement d'un fichier comme source de données directe du moteur.

Les types datetime, actuellement au nombre de 2, se voient accompagnés de quelques autres. Les types date et time permettant de ne traiter que la partie qui nous intéresse, d'où gain de place important. Le type datetime2 permet d'exploiter toute la précision du type DateTime .Net et d'apporter une précision supérieure. Il est aussi possible de stocker le fuseau horaire avec le type DateTimeOffset.

Le type personnalisé de Table voit sont apparition, il permettra le passage d'arguments complexes au niveau des procédures stockées. Les types Utilisateurs écrits en .Net, quant à eux, voient leur limite passer de 8ko à 2 Go de données. Les types tables sont aussi exploitables par .Net (et au niveau client aussi).

```
CREATE TYPE MonTypeTable
AS table (a int, b varchar(50))
```

```
CREATE PROC MaPoc (@liste MonTypeTable READONLY) AS
INSERT MaTable SELECT * FROM @liste
```

Arrivée aussi des index spatiaux, avec la création de points comme type de données dans le moteur. Exploités avec .Net ceux-ci permettront aussi bien le stockage de coordonnées et indirectement de formes complexes, mais aussi les requêtes complexes de type : " Quels sont tous les points proches de celui-ci ? ". Tout ce qui gravite autour du monde de la géo localisation devrait en bénéficier. Si vous traitez des coordonnées géographiques cette version est pour vous.

Dernier type, défini par le standard SQL, la possibilité d'utiliser un fichier comme source de données. Appelé FileStream il permet d'exploiter tout fichier comme partie intégrante de la base de données avec une gestion des transactions dans cette dernière.

Les autres avancées de cette version

Parmi les autres modifications marquantes, on pourra noter l'intégration native du moteur d'indexation de contenu (Full Text Search). Celui était jusqu'à lors un composant externe, il était livré non seulement avec SQL Server, mais aussi Windows et d'autres. L'intégration avait déjà débuté sous SQL Server 2005 avec le stockage des index dans des fichiers rattachés à la base de données. L'idée ici est de fournir des scénarios d'optimisation plus poussés sur des requêtes mixtes (sur données et index classiques et sur index intégral).

Un nouveau mécanisme fait son apparition, le CDC (Change Data Capture). Ce mécanisme fait appel au journal de transaction de la base de données pour effectuer le suivi des modifications faites sur une ou plusieurs tables dans une base. Ce mécanisme, contrairement aux triggers qui permettent réellement l'interception des commandes de mise à jour et d'y mettre fin, agit après l'opération de manière asynchrone, scanne le journal et envoie le tout dans une table de suivi des modifications, et cela sans les problèmes de blocage couramment rencontrés avec les triggers. Il devient alors virtuellement possible d'avoir accès à une table à n'importe quel point dans le temps.

De la CTP à la RTM

Les fonctionnalités seront mises à disposition au fur et à mesure de l'avancée des développements. Certaines des fonctionnalités décrites ici n'y sont pas encore, vous serez aussi peut être étonnés de ne pas en trouver d'autres déjà dans le Service Pack 2 de SQL Server 2005. Seule certitude, les fonctionnalités des CTP seront dans la version finale, une promesse de Microsoft. Et qui sait, peut-être y aura-t-il de bonnes surprises dans les CTP suivantes.

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

L'ACTU AU QUOTIDIEN
www.programmez.com

L'ESSENTIEL DU DÉVELOPPEMENT

@Sienerinformatique

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

DEVELOPPEMENT.NET

Visual Studio 2005 Professional

Des outils conçus pour relever les défis de tous vos développements

Version Boite **795** €HT sans abonnement MSDN

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

Avec abonnement MSDN Pro

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

à partir de, par an, **585** €HT

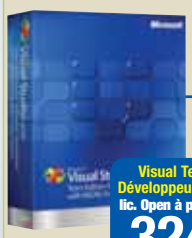
LICENCE OPEN 2 ans

Avec abonnement MSDN Premium

LICENCE OPEN 2 ans

à partir de, par an, **1325** €HT

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



Visual Studio 2005 Team System

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

Visual Team Tester, Développeur, ou Architect lic. Open à partir de, par an **3247** €HT

Visual Team Suite lic. Open à partir de, par an **6499** €HT

Nous consulter pour devis sur mesure

Microsoft GOLD CERTIFIED Partner

Borland

Turbo Delphi Win 32 **Turbo Delphi .NET**

Turbo C++ 2006 **Turbo C#**

Des solutions extensibles et personnalisables comportant plus de 200 composants glisser-déposer pré-intégrés fournissant tous les éléments nécessaires au démarrage d'un vrai développement.

395 €HT

COMPOSANTS NET ASP JAVA

Infragistics

NetAdvantage for .NET 2006 Volume 3

à partir de **725** €HT

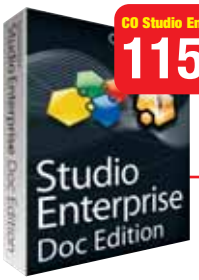
Support .net2 et Visual Studio, Contrôles Windows Presentation Foundation, interfaces stylées Office 2007…

NetAdvantage for JSF 2006 Volume 2

à partir de **575** €HT

Composants serveurs Java type AJAX pour applications J2EE (grille, menu, arborescence, tableau, calendrier ect.)

OUTIL DE CREATION D'AIDE



CO Studio Ent Doc Ed* **1150** €HT

ComponentOne Doc-To-Help 2007

La solution la plus connu du marché pour créer l'aide en ligne et la documentation des applications que vous avez développées

Nouvelle version 2007

**Le CO Studio Entreprise Doc Edition est un pack comprenant Doc- To-Help et la suite de composants Studio Entreprise*

ComponentOne Doc-To-Help Entreprise 2007 **820** €HT

Doc-To-Help for Word 2007 **615** €HT

DÉVELOPPEMENT XML

XMetaL 5.0 Author — XMetaL JUSTSYSTEMS

à partir de **658** €HT

La puissance pour les auteurs de contenus

XMetaL permet aux auteurs de se consacrer exclusivement à la qualité et la cohérence de contenus de haute qualité, en faisant abstraction de l'ensemble des détails de formatage, contrôle de validité, et maintien des hyperliens Avec un an de maintenance

ALTOVA xmlspy 2007

Version Pro **395** €HT

Version Entreprise **799** €HT

Un outil de développement XML pour modéliser, éditer, déboguer et transformer toutes les technologies XML. Permet de générer du code en plusieurs langages. Idéal pour les développeurs J2EE, .NET, Eclipse, ou bases de données.

ÉDITEUR DE TEXTE

Nouvelle version 13

44,95 €HT

UltraEdit 32 v13

L'éditeur de texte et programmation n°1.

Plus d'un million d'utilisateurs.
10 ans d'existence et de développement

INSTALLATION

399 €HT

Language Pack **359** €HT

Mise à jour concurrentielle **209** €HT

altiris

Wise Installation Express

Développez simplement et rapidement vos packages MSI les plus basiques

Si vous cherchez un outil de développement de packages MSI simple pour vos besoins de développement les plus courants, alors n'hésitez pas et choisissez Wise Installation Express.

DESIGN

Microsoft Expression Web

Nouveau produit

Enfin disponible

Créez un site Internet à votre image

Un outil professionnel de conception dédié à la création de sites modernes fondés sur des standards et proposant une qualité supérieure sur le Web.

à partir de **295** €HT

Visitez notre espace

L'ESSENTIEL DU DÉVELOPPEMENT

pour trouver tous les outils dont vous avez besoin

CONTACTEZ NOUS AU

01 56 20 24 20

www.sienerinformatique.com

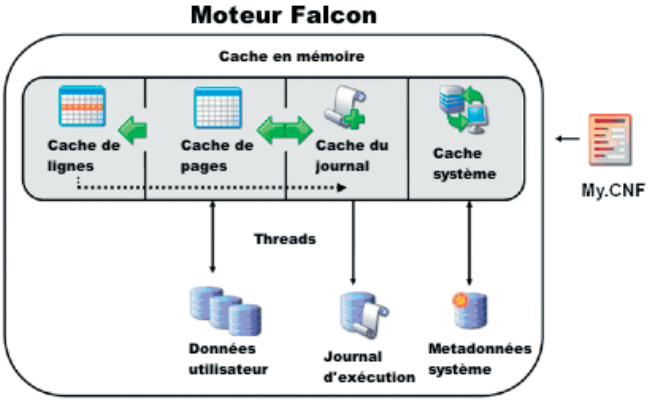
MySQL 6.0 : le moteur Falcon

MySQL AB travaille d'arrache-pied sur un moteur de table transactionnel maison. Depuis avril 2007, Falcon est même devenu le fer de lance de la version 6.0 de MySQL. Rapide, simple à utiliser, adapté pour le Web, et, bien sûr, transactionnel, Falcon est un moteur de stockage pour MySQL : il s'insère dans le serveur de base de données comme MyISAM ou InnoDB. Falcon est un projet distinct de FireBird, mais a été bâti sur l'expérience de son auteur pour tirer les meilleures performances.

Falcon mode d'emploi

La philosophie du moteur est d'éviter autant que possible d'utiliser le disque. Au fond, son but n'est pas d'accélérer les opérations avec le disque, mais de les supprimer purement et simplement, pour utiliser à la place des systèmes de cache habiles, en mémoire. Jim Starkey note que les processeurs et la mémoire ont fait des progrès importants depuis les années 70, mais que les disques n'ont pas suivi.

Les données sont mises en cache en mémoire. Jim Starkey appelle cela le cache de ligne, à ne pas confondre avec le cache de requête : ce dernier enregistre les résultats des requêtes fréquentes pour les servir plus vite. Ici, le cache de ligne, qui est distinct du cache de requête, mais travaillera finalement avec lui, enregistre des lignes en mémoire. Ces lignes sont alors réutilisées dès qu'une requête les sollicite. Falcon est assez habile pour rechercher les lignes dont il a besoin en mémoire, puis sur le disque. Une requête peut donc tirer une partie des lignes déjà chargées par une requête précédente, avant de commencer à solliciter le disque. Ce système fonctionne pour les sélections comme pour les modifications : les lignes qui sont conservées en mémoire sont capables d'entretenir des versions différentes. Cela permet aux transactions de travailler de manière indépendante, aux UPDATE de s'exécuter simultanément, sans poser de verrou de table, ni même de ligne. Jim Starkey appelle cette technique 'Multi-Version Concurency Control'. Cette approche permet d'obtenir un très haut niveau de simultanéité dans l'exécution des commandes SQL, aussi bien en lecture qu'en écriture. Pour gérer ce cache, Falcon dispose de deux paramètres, sur les trois disponibles pour le configurer : falcon_min_memory et falcon_max_memory. Le moteur va utiliser tout ce qu'il peut de mémoire pour stocker les lignes et leurs versions. Lorsque la limite supérieure est atteinte, un thread parallèle entre en jeu, et fait le tri entre les lignes qui sont souvent utilisées, et les autres. Ce nettoyage de la mémoire fonctionne sur le même critère que pour le cache de requêtes, mais il est beaucoup moins fréquent : au lieu de faire de la place pour de nouvelles données quand elles se présentent, Falcon fait un grand nettoyage de temps en temps. Dès qu'il s'approche de falcon_max_memory, il tente de réduire sa consommation à falcon_min_memory. Cette technique de gestion du cache porte le nom de *scavenging,* ou vidange en français. D'un autre côté, Falcon conserve une seule version des données sur le disque. Cela renforce considérablement leur sécurité et leur intégrité. Pour cela, les données sont écrites sur le disque avec une technique d'écriture prudente, qui laisse l'espace de table toujours cohérent. Même en cas de crash, durant une écriture, le fichier reste très cohérent. De la même façon, les transactions sont enregistrées séquentiellement dans un journal d'opération, avec les données et une



date. Parallèlement, un thread se charge de transférer ces transformations et données sur le disque. Seuls les blob (grands objets binaires) sont écrits directement dans l'espace de données, pour ne pas polluer le journal. Comme InnoDB, Falcon fonctionne avec un système d'espace de table. Toutes les tables sont rangées dans un seul grand fichier, que Falcon se charge de faire croître au fur et à mesure des besoins. Contrairement à InnoDB, il n'y a aucun paramètre de configuration à manipuler, tout est automatique. Falcon crée un fichier .fts pour enregistrer les tables et les informations sur la base; il crée ensuite deux fichiers, .fl1 et .fl2 pour enregistrer les journaux d'exécution sur le disque. Enfin, il utilise le traditionnel fichier .frm pour enregistrer les définitions des tables. Les données sont compressées avant d'être écrites sur le disque. Avec cette approche, on fait porter le poids du travail sur le processeur, qui doit compresser, et non plus sur le disque, qui doit écrire ou lire moins de données d'un seul coup. Les données sont utilisées dans la compression lorsqu'elles sont en cache mémoire.

De leur côté, les index sont des B-tree, avec une compression de préfixe. Ce sont des index à taille variable, contrairement aux autres moteurs de MySQL. Cela permet de gagner en espace disque. Les index sont totalement indépendants des données : ils sont manipulés en interne sous forme d'entiers. Ils sont aussi capables d'avoir plusieurs versions, comme les données en mémoire, puisque ces dernières peuvent avoir plusieurs versions. Lors de la sélection de données, l'index est analysé pour produire une liste continue de lignes à lire. Ces lignes sont alors lues sur le disque, dans l'ordre le plus efficace. Ce travail préparatoire évite au disque de lire les données aléatoirement, avec un effet de girouette inefficace. Cette approche est semblable aux tables qui organisent les données sur le disque en fonction d'une clé primaire. Falcon a étendu le concept à tous les index de la table.

Architecture

Les architectures 64 bits seront idéales pour lui, mais il fonctionne aussi sur les systèmes 32 bits. Cela sera un avantage indéniable sur le moteur actuel MySQL, qui ne tire pas avantage de toutes les capacités des nouveaux serveurs. Il est disponible au téléchargement sur le site de MySQL, dans la rubrique MySQL 6.0 : http://dev.mysql.com/downloads/mysql/6.0.html. Les versions Linux, Source et Windows sont disponibles. Une version bêta de Falcon devrait être publiée vers la fin de l'année 2007.

■ **Damien Seguy** - http://www.nexen.net/

Les tests de recette avec Fitnesse

Combien de développeurs se sont entendus dire : « Hum… hum, vous êtes sûr d'avoir bien compris la spécification détaillée ? » . Ou encore : « Avant, elle marchait bien cette fonction : pourquoi ça ne marche plus maintenant ?! ». Voire : « Mais on ne l'avait pas déjà fait corriger ce bug là ? Je ne comprends pas : plus vous codez… et moins l'appli fonctionne ?!? ».

Laissez faire le temps et vous pouvez être certains que spécifier, vérifier et assurer la non régression d'une application de gestion deviendra un douloureux défi. Heureusement, il existe une solution encore méconnue des développeurs et des maîtres d'ouvrage : les tests de recette automatisés. Ils permettent de relever ce défi sans sarcasme, ni douleur.

« Un autre test est possible »

Examinons d'abord les pratiques actuelles autour du test fonctionnel. La maîtrise d'ouvrage, ou « MOA », a pour habitude de pratiquer des tests manuels ou des tests d'interface utilisateur, appelés aussi « tests GUI » . Au premier abord, les tests GUI semblent pertinents et efficaces : quoi tester de mieux que ce que l'utilisateur a en face des yeux ? Seulement les tests GUI connaissent quelques limitations importantes (Fig. 1).

D'abord, ils agissent sur l'interface utilisateur et sont sensibles à ses fréquentes modifications : chacune de ses évolutions nous contraint à adapter les tests GUI.

Ensuite, les tests GUI manipulent l'application par… son interface graphique (d'où leur nom !). Or ils ne permettent pas de rendre une spécification plus complète, suffisamment précise et univoque. Pourquoi ? D'une part parce que l'objet du test GUI n'est pas le scénario fonctionnel, ou ses règles de gestion. En agissant au niveau de l'interface utilisateur, les règles de gestion sont testées mais en surface. Par exemple, un logiciel de gestion de crédit peut définir la notion d'OLB, c'est-à-dire le capital restant dû, dans son fonctionnement interne. Si la GUI intègre certainement la notion d'OLB, il paraît plus improbable qu'elle en explique le fonctionnement. Par exemple, l'OLB s'applique-t-il avant le paiement d'une échéance, ou après… ?

D'autre part, la spécification ne devient pas plus complète, plus précise et univoque tout simplement parce que la personne qui spécifie et teste l'application discute rarement avec la personne qui la conçoit et la développe. Or,

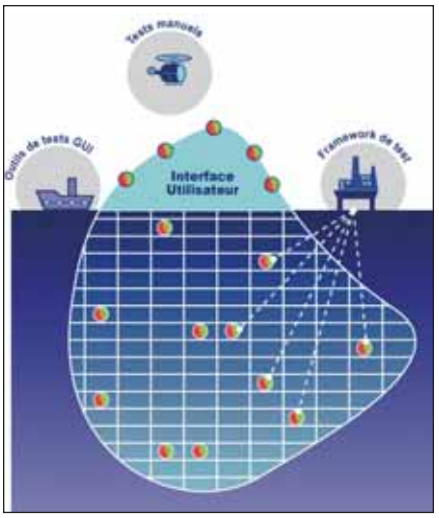


Fig. 1 - Les différentes pratiques de test fonctionnel

avec le temps, l'application va naturellement se compliquer... et la spécification risque de devenir de plus en plus incomplète, imprécise et ambiguë.

Les tests manuels, quant à eux, souffrent d'un manque de productivité. Ce point est important car le risque de régression augmentera exponentiellement avec le nombre de fonctionnalités. Mais justement, n'est-ce pas lorsque l'application se complique qu'il devient vital de pouvoir rapidement les vérifier tous, aussi nombreux soient-ils ?

Enfin, les tests manuels sont généralement consignés dans un document Word, tel qu'un cahier de tests. Il s'agit d'un document électronique condamné à être sans lien direct avec l'application telle que construite. Avec le temps, il n'est pas rare de constater qu'un cahier de tests ne correspond plus à son application en production.

Des tests fonctionnels efficaces devraient agir comme un contrat univoque entre la MOA et les développeurs. Ce contrat serait rigoureusement insensible aux évolutions de l'interface utilisateur.

Ils devraient pouvoir être exécutés automatiquement pour assurer la non régression de l'application. Leur exécution automatique devrait être en lien direct avec l'application développée, en l'invoquant directement par exemple. Dès lors, on pourrait les qualifier de « spécifications exécutables ».

Les tests de recette « racontent des histoires »

Pour illustrer notre propos, rien ne vaut un exemple concret de scénario fonctionnel, mieux : une histoire. L'histoire que nous allons utiliser provient d'Octopus, une suite logicielle intégrée de micro crédit, développée par OCTO Technology pour le compte d'une ONG : ACTED (1).

Azam est un petit artisan qui a besoin de fonds pour lancer son entreprise de confection artisanale. Azam se rend donc à l'agence ACTED de Kaboul, le 1er janvier 2006. L'agence lui propose un contrat de crédit de 1000 TJS (acronyme du somoni, la monnaie tadjik) aux conditions suivantes : le crédit comprend 2 échéances mensuelles, avec un taux d'intérêt mensuel fixe négocié de 3%. A ce stade, on souhaite tester que l'échéancier du contrat créé est « valide » : en tenant compte des intérêts, cela signifie simplement qu'Azam s'engage à rembourser mensuellement 500 TJS de capital et 30 TJS d'intérêts pendant 2 mois, soit 530 TJS par mois. Azam se présente le 1er février et rembourse 530 TJS.

A ce stade, on souhaite tester que l'échéancier courant d'Azam a été mis à jour avec ce paiement.

Azam se présente le 1 mars et rembourse 530 TJS. Il a soldé son prêt.

A ce stade, on souhaite tester que l'échéancier courant d'Azam a été mis à jour avec ce paiement, et que le crédit a été soldé.

Fin de l'histoire.

Pour tester ce scénario fonctionnel d'Octopus, très simplifié pour nos besoins, nous avons donc besoin d'un outil :

- utilisable par un non-développeur pour exprimer ce scénario en un test de recette ;
- invoquant l'application lors de son exécution ;
- et permettant de le vérifier de manière automatisée.

(1) Vous voulez en savoir plus ? Rendez-vous sur : http://www.octopusnetwork.org/

Le premier point interdit l'usage des frameworks de tests unitaires : on ne peut décemment pas demander à un expert en micro finance d'utiliser Visual Studio ou Eclipse pour exprimer ce test de recette. Il maîtrise en revanche Excel sur le bout des doigts. C'est là que Fitnesse arrive à la rescousse.

Fitnesse à la rescousse

Fitnesse est un outil open source pour les tests de recette (Fig. 2).

Fitnesse est composé de plusieurs parties :

- un site web comprenant des pages wiki qui sont les tests de recette : le site « Fitnesse » ;
- un moteur d'exécution de ces tests de recette : le moteur « FIT » ;
- du code de liaison avec l'application à tester. Ce code peut être écrit en C#, en JAVA ou en C++ ;
- et enfin l'application à tester (écrite suivant ces mêmes langages). En l'occurrence, Octopus est écrit en C#.

Dans son principe, Fitnesse envoie des données tabulaires à l'application à tester, reçoit le résultat de l'exécution et vérifie sa conformité par rapport au résultat attendu. Si la vérification est conforme, alors elle est surlignée en **vert.** Sinon, elle est surlignée en **rouge**.

Fitnesse est conçu pour être utilisable par un maître d'ouvrage pour exprimer ses tests de recette : il utilise un simple site wiki pour les écrire et les exécuter.

Fitnesse présente une fonction très utile : les tableaux saisis dans Excel peuvent être collés dans le site wiki, en respectant la structure tabulaire des données. L'utilisation d'Excel et l'édition wiki facilitent son appropriation et rendent l'outil très simple d'utilisation. En quelques minutes de Fitnesse, l'histoire d'Azam peut être exprimée par le test de recette suivant : (Fig : 3)

Ce test vérifie l'opération de création d'un échéancier de contrat, et son contenu après remboursements, comme l'indiquent les titres « Création du contrat » puis « Vérification de l'échéancier ». Il présente des noms de colonne compréhensibles et en français, des échéanciers au format tabulaire parfaitement intelligibles. Le site Fitnesse reste un site web : il peut donc être enrichi de commentaires, d'images, de définitions ou d'explications précises sur les règles de gestion et sur le test de recette. Par exemple, chaque tableau est précédé d'une phrase explicative

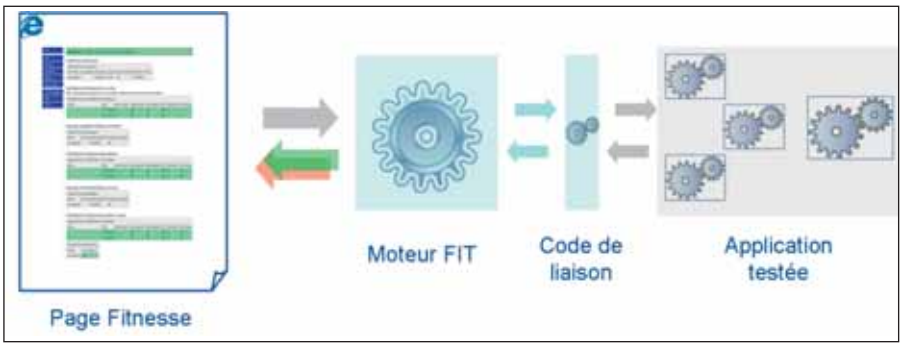


Fig 2 - Fonctionnement de Fitnesse

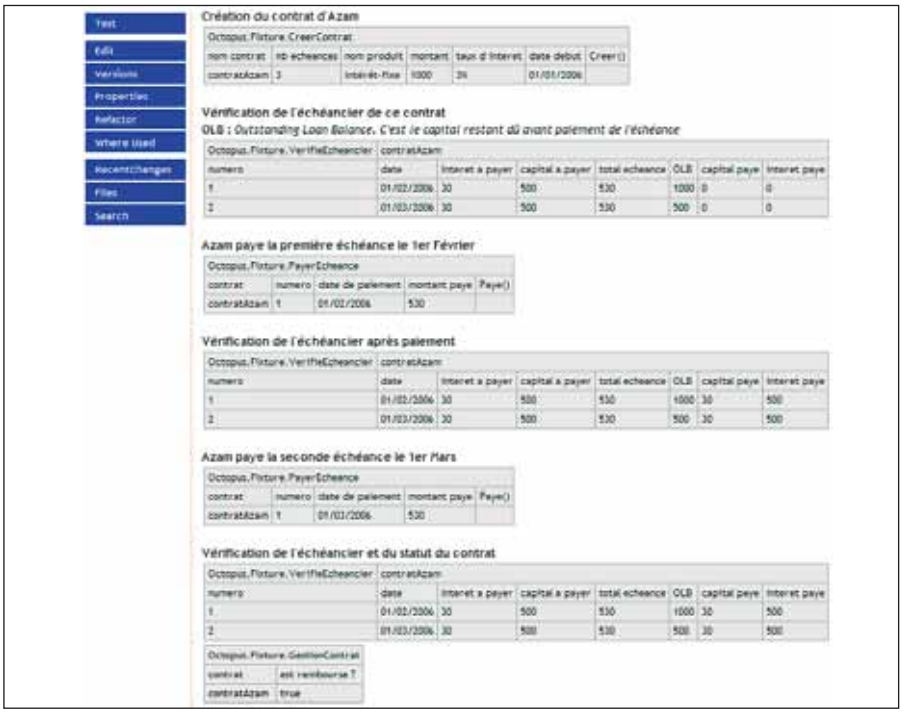


Fig 3 - Ecriture d'un test Fitnesse

sur l'opération effectuée, « OLB » est défini dans la page Fitnesse (c'est le capital restant dû avant le paiement de l'échéance ;-)… Au passage, ces explications mettent en lumière l'exercice de communication entre développeurs et MOA, mais ça, c'est une « autre histoire »… Quoi qu'il en soit, à l'arrivée, le test est exprimé de manière précise et univoque.

Comment exécuter le logiciel lors des tests de recette ?

Pas de miracle, il faut écrire du code de liaison entre le wiki Fitnesse et notre application : simple mais indispensable.

Ce code de liaison est écrit par le développeur, en tenant compte du test de recette proposé par la MOA. Pour cela, FIT (2) intègre une API en C#, JAVA et C++ comprenant à peine 5 verbes. Dans notre cas, comme bien souvent, deux d'entre eux suffiront : ColumnFixture et RowFixture. Une « Fixture » est une classe qui vérifie que le comportement de l'application testée est bien celui décrit dans le wiki.

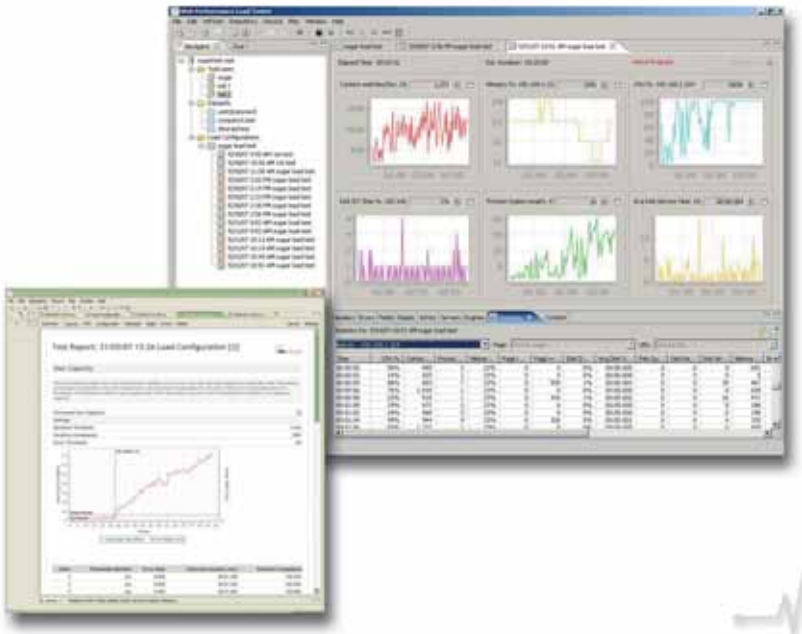
```
public class CreerContract : ColumnFixture
{
    public string nomContrat;
    public int nbEcheances;
    public decimal montant;
    public string nomProduit;
    public double tauxDinteret;
    public DateTime dateDebut;

    public int Creer()
    {
        ContratCredit contrat = new ContratCredit();
        contrat.nom = nomContrat;
                .....
```

(2) Pour la documentation complète, voir http://fitnesse.org ou lire l'excellent ouvrage « Fit for Developing Software - Framework for Integrated Tests » de Rick Mugridge et Ward Cunningham, aux éditions « Prentice Hall ».

webperformance SUITE

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



- **Développement automatisé de cas-tests et Test de Charge**
- **Rapports d'analyse des tests de performance et de charge**
- **nouveau Analyse des serveurs et de l'impact sur les performances**

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

Evaluation gratuite de Web Performance Suite sur http://www.kapitec.com/Pub/WP?id=99

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

kapitec Software

Power Your Web Projects

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

Web Performance est une marque déposée de Web Performance, Inc.

```
        contrat.Creer();
        FixtureDictionnaire.SetValue(nomContrat,
contrat);
        return 1;
    }
}
```

Un exemple de cas de test de type ColumnFixture.

La fixture CreerContrat, de type ColumnFixture, permet d'appeler la méthode applicative qui crée un contrat. On notera 4 points importants dans le code de cette classe :

- Les attributs de la classe possèdent les mêmes noms que les éléments du tableau dans le wiki : l'association est effectuée par convention de nommage.
- Ces attributs possèdent une visibilité public.
- C'est la méthode Créer de la fixture qui effectue l'appel à la méthode applicative qui crée un contrat.
- Le contrat créé est stocké dans un dictionnaire qui permet de retrouver simplement les objets créés via un mécanisme de clé / valeur.

La fixture suivante est de type RowFixture. Elle permet de vérifier l'échéancier d'un contrat après sa création ou après un remboursement. Plus généralement, une RowFixture permet de vérifier le contenu d'une liste.

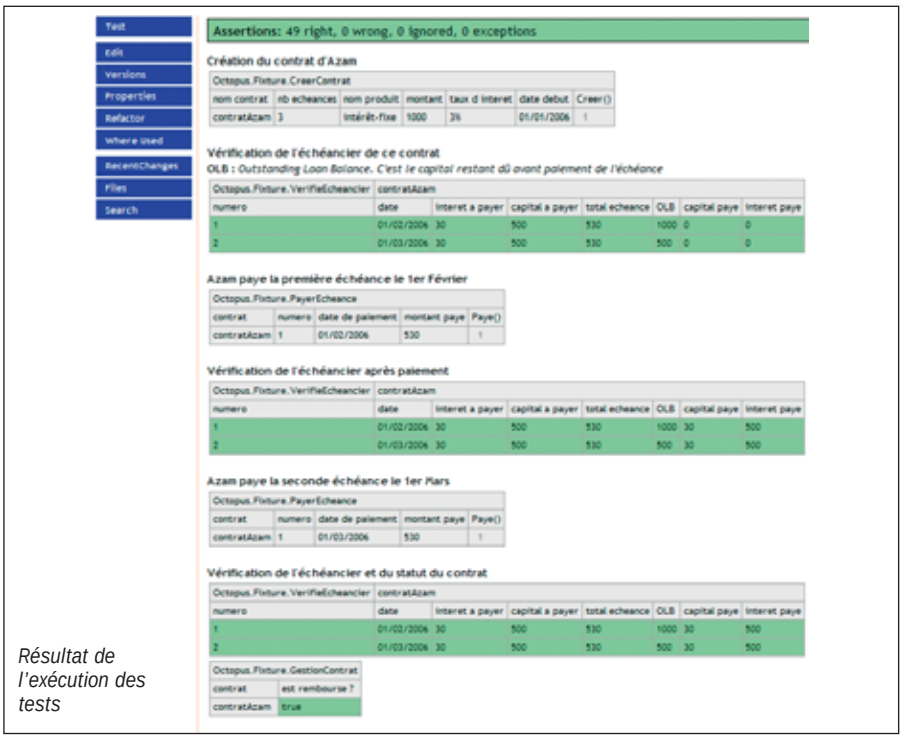
Les comportements de 2 méthodes doivent être redéfinis :

- La méthode Query est surchargée pour définir la liste à retourner. Pour la fixture VerifieEchancier nous souhaitons retourner une liste d'échéances.
- La méthode GetTargetClass est surchargée pour spécifier le type des éléments de la liste.

```
public class VerifieEcheancier : RowFixture
    {
        public override Type GetTargetClass()
        {
            return typeof(FixtureEcheance);
        }

        public override Object[] Query()
        {
            ContratCredit contrat =
                (ContratCredit) FixtureDictionnaire.
GetValue(Args[0]);
            Object[] listeEcheances = new Object
[contrat.ListEcheance.Count];
                    .....
            return listEcheances;
        }
}
```

Un exemple de cas de test de type RowFixture.



Résultat de l'exécution des tests

Le code de liaison est très léger. Il n'est écrit qu'une seule fois et permet ensuite de décliner un grand nombre de tests fonctionnels. Par exemple, Octopus réalise environ 10 000 vérifications (3) sur un peu plus de 60 pages de tests pour à peine 500 lignes de code de liaison. L'investissement en charge de développement est tout à fait acceptable, et le retour sur investissement est non négligeable. Imaginez maintenant une ou plusieurs personnes devant vérifier l'ensemble de ces tests à chaque modification du code du logiciel, avec un cahier de tests par exemple. Connaissez-vous de tels surhommes ? Pour notre part nous n'en avons jamais rencontrés. De plus, grâce au code de liaison, le test Fitnesse n'est pas adhérent à la GUI de l'application. Il sera tout simplement insensible aux modifications de « cosmétique ». Enfin, l'exécution de ces tests s'effectue en un seul clic en se plaçant sur une page de tests. Fitnesse permet également de lancer toute une arborescence de pages de tests. Cerise sur le gâteau, l'exécution des tests est rapide : à peine quelques minutes pour l'ensemble des tests d'Octopus. Fitnesse possède donc un avantage capital pour assurer la non régression de l'application puisqu'il permet de rejouer en permanence l'ensemble des tests de manière automatisée.

(3) Oui, vous avez bien lu : 10 000 vérifications. Ces tests sont consultables sur http://octopusfit.octo.com

Trêve de bavardages, exécutons notre test. Après le développement du code de l'application, et quelques mises au point, le résultat des tests est sans équivoque : tout est vert, ce qui signifie que le code de l'application est conforme au test de recette rédigé.

Quelques limites de Fitnesse

Dire que Fitnesse n'a aucun défaut serait mentir. L'outil ne sait pas produire nativement de reporting orientés tests de recette : vélocité, suivi de leur nombre, pourcentages de succès et d'échec. De plus, l'absence d'un éditeur WYSIWYG rend l'édition des pages wiki plus ardue. Si elles ne sont pas indispensables dans un premier temps, ces fonctionnalités s'avèrent utiles avec l'augmentation du nombre de tests. D'autres outils, à l'instar de GreenPepper, semblent les proposer.

Conclusion

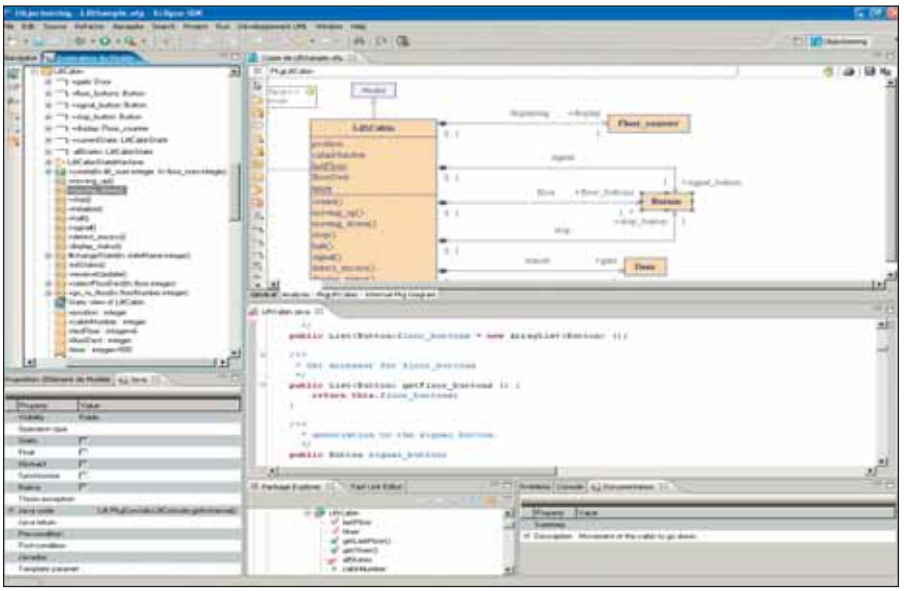
Comme nous l'avons vu sur cet exemple concret, Fitnesse a répondu à notre besoin qui était de rédiger des tests de recette univoques et automatisés. De plus, grâce au code de liaison, les tests sont effectués directement sur l'application. De cette manière, ils permettent d'assurer sa non régression.

Messaoud Oubechou et **Nicolas Baron,**
Architectes - Centre de Compétences .NET
OCTO Technology
www.octo.com

L'expertise Java pilotée par le modèle UML

Objecteering 6 optimise MDA et UML2 pour générer un code Java5 d'un haut niveau d'expertise : il maximise la productivité et la qualité des développements Java.

Comment tirer parti au mieux de la modélisation UML à des fins de production automatisée d'un code de qualité, maintenu en cohérence avec le modèle ?
L'approche MDA qui consiste à exploiter le modèle par des mécanismes de transformation répond précisément à cette problématique en assurant également la traçabilité entre le code généré et les modèles. Avec Objecteering 6, Objecteering Software met à disposition des développeurs Java une nouvelle génération d'outils de développement guidé par le modèle, en s'appuyant sur les dernières avancées de MDA de UML2.1 et de Java5.



Objecteering 6 intégré à la plate-forme Eclipse

Un haut niveau de services de guidage et d'automatisation dédiés Java

Intégré à la plate-forme Eclipse, Objecteering 6 guide et automatise le développement Java en combinant le meilleur des ateliers UML avec le meilleur des IDE Java. Il apporte :

- Une production automatisée du code Java, toujours cohérente avec le modèle,
- Le support de patterns dédiés Java (RMI, event listener, etc.) pour plus de productivité,
- Une production de doc javadoc automatique,
- Un support complet de Java5 à partir de UML2
- Une génération automatisée de la chaîne de production (ANT) à partir du modèle
- Un support dédié du travail en équipe, avec des " composants de modèles " assurant la diffusion de parties applicatives Java/UML vers d'autres équipes.
- Un reverse engineering Java pour reprendre les applications existantes
- Des extensions (modules gratuits) pour cibler des frameworks opensource (Struts, Hibernate, JSF, Spring)
- Un paramétrage puissant et aisé, à l'aide de templates, permettant d'adapter la génération de code Java à son contexte avec la technologie MDA.

La richesse de UML2 traduite en Java

Objecteering 6 exploite au mieux les nouvelles capacités du standard UML 2.1. Les templates UML2 permettent ainsi efficacement de modéliser et générer les generics Java. La gestion des liens d'import, couplée à un service de contrôle de cohérence sur leur bonne utilisation permet de gérer finement les modes d'import Java avec leurs diverses options d'emploi. Les associations sont pleinement exploitées, pour déduire les librairies de container appropriées. Les annotations et metadatas sont supportées à partir du modèle
La prise en compte de UML2 se prolonge jusqu'au déploiement de l'application où les diagrammes de déploiement UML2 prennent tout leur sens en permettant la génération de la chaîne de production (ANT).

Représentation UML d'un Jar déployant un package (puis génération ANT)

Une gestion de cohérence modèle/code permanente

Avec Objecteering 6, le modèle est le code : le développeur complète le modèle UML par l'algorithmique des méthodes, tout en étant guidé dans l'outil. La modélisation est intégralement exploitée, l'utilisateur n'ayant jamais à reformuler le modèle en Java. Les compléments de code sont attachés aux éléments du modèle, stockés dans le référentiel. A chaque instant la cohérence du modèle UML et du code Java est garantie. Le développeur peut ainsi éditer son application au niveau modèle sous Objecteering ou au niveau code sous Eclipse: quand le modèle est modifié, Objecteering re-génère le code de façon incrémentale et quand le code est modifié, il assure la mise à jour du modèle via son référentiel.

Une ergonomie dédiée au programmeur Java

Au sein du modèle UML, le développeur bénéficie d'une interface utilisateur spécifique Java : il retrouve les notions Java (Jar, wrapper, interface, final, etc.), dispose de boites de saisie dédiée, et de diagrammes UML spécifiques.
Il dispose aussi de services dédiés Java (javadoc, patterns, etc.) pour rendre son modèle encore plus productif.

Objecteering SOFTWARE

Objecteering Software, éditeur de l'atelier Objecteering 6, est le spécialiste français UML/MDA pour le développement d'applications guidé par le modèle.
Sa suite d'outils couvre le cycle de vie de la gestion des exigences jusqu'au déploiement de l'application pour les cibles Java/J2ee, C#.Net, C++, SQL, Corba et Fortran.

Pour plus d'informations :

www.objecteering.com
info@objecteering.com
Tél. : 01 30 12 16 60

Web 2.0 : choisir, créer, déployer !

1re partie

Difficile depuis bientôt 2 ans de passer à côté du phénomène Web 2.0. Concept avant tout, ce " nouveau " web est une conjugaison de techniques, de langages, de technologies, de concepts. Il y a deux volets à considérer : l'utilisateur et l'interface riche, la technicité avec le développement. L'utilisateur final veut aujourd'hui une interface dynamique, élégante, bref quelque chose qui puisse ressembler à ce que l'on trouve sur les bureaux des systèmes comme MacOS X, Windows Vista et certains thèmes GNU Linux.

Comme toujours, si le package est beau, la plomberie demeure compliquée. Le Web 2.0 ne simplifie pas le problème du développeur, de l'administrateur, du testeur, du webmaster ! Au contraire, la multiplication des couches, des outils, des langages, des frameworks, complexifie le développement d'un site, d'une application web 2.0.

C'est à partir de ce constat que nous avons voulu faire établir un panorama complet aussi bien au niveau d'Ajax, de .Net, de Ruby que d'Adobe AIR et JavaFX. Le Web 2 suscite bien des questions. Impossible en quelques dizaines de pages de survoler l'ensemble de cet l'univers.

Qu'est-ce que Web 2. ? Quelles architectures choisir ? Quel modèle technique et économique ? Comment choisir son framework Ajax ? Faut-il adapter ou partir de zéro ? Pourquoi et comment mettre en œuvre silverlight ou Rails ? Quelles sont les limites du web 2 aujourd'hui ? Nous verrons aussi les risques d'attaque et les failles du web 2 au niveau sécurité (et ces risques sont malheureusement trop souvent occultés). On parlera aussi de stratégie, de mashup, de gadgets. Nous aborderons quelques techniques avancées sur le XML.

N'hésitez pas à faire partager votre expérience, positive ou négative, à propos de ce nouveau web sur nos forums : www.programmez.com. Nous restons ouverts tout l'été !

François Tonic

Mashup : le concept à tout faire ?

Depuis quelques mois, le terme mashup est partout ou presque. Dans les salons, conférences, dans la presse, mashup revient encore et toujours. Mais finalement, qu'est-ce que le mashup ? Il s'agit, basiquement, d'agréger, de mixer, de rassembler, du contenu de différentes sources, et de différentes natures.

Le mashup concerne tout et n'importe quoi : son ERP, sa base de données, les flux RSS, les images, la vidéo, les animations, le système, la musique, etc. Le mashup s'utilise aussi bien dans un contexte grand public que métier au niveau entreprise. Pour l'utilisateur final, il s'agit de pouvoir accéder rapidement aux informations. Il peut prendre différentes formes et l'une des plus connues reste le widget (ou gadget) comme on peut en trouver chez Google, Yahoo, Dashboard d'Apple ou encore Windows Vista, Live.com.

Portail en ligne, sur le bureau

Une des utilisations du mashup concerne le portail d'entreprise, interne ou non. Couplé aux gadgets, cela permet aux utilisateurs finaux de rapidement créer leur propre page portail avec les informations nécessaires. Pour BEA, le mashup est défini comme cela : " application web qui combine du contenu provenant de plusieurs sources pour créer un nouveau service à valeur ajoutée ". Ce qui n'est pas forcément le cas. C'est un service mais pas forcément à valeur ajoutée. Dans ce contexte, il devient aisé de mixer du mapping (Google Maps ou autre) avec un fichier client par exemple. Les technologies mises en œuvre sont multiples : Java, .net, Rails, PHP, XML, REST, RSS, Ajax, etc.

Le gadget comme mashup ?

Les gadgets constituent une forme de mashup, par exemple : récupérer des informations systèmes sur le trafic réseau, accès à des données de son SAP, accès à une banque d'image, etc. Le gadget possède deux éléments : une partie métier avec le code et une partie présentation (interface). Le plus souvent, il fonctionne en mode connecté. " Le widget est une évolution des portlets. C'est interactif et cela renforce le client léger (le navigateur)", précise François Bourdoncle (PDG et co-fondateur d'Exalead). D'ailleurs, on lie beaucoup mashup et gadget, ainsi qu'avec RSS, Atom ou encore JSON. Le mashup a une dimension d'intégration entre les sources d'information, et même au niveau des applications. Il est aisé d'agréger un web service, un gadget, un flux RSS, etc. L'utilisateur veut aussi agréger et donc créer ses propres mashup. Le développeur fournira le composant technique mais pas forcément l'agrégation. Par contre, à l'administrateur de gérer le mashup au niveau accès, identité, sécurité (car n'oubliez pas d'identifier les sources des contenus et éviter des données non certifiées). D'autre part, un des soucis du mashup sera la disponibilité des données, des services et donc de leur pérennité. Donc soyez vigilants. L'un des soucis du mashup concerne donc la disponibilité des services, des données. Est-on sur de la pérennité, du gadget, du service web, du service, du site que l'on agrège ? Impossible de l'assurer à 100%. D'autre part, quelle est la disponibilité de la donnée agrégée ? Là encore impossible de répondre, cela va dépendre du fournisseur de la donnée, de l'état du réseau. Ensuite, mon mashup est-il stocké localement ou uniquement en ligne ? Faut-il alors prévoir un fonctionnement déconnecté (possible uniquement dans certains scénarios, par exemple avec Google Gears, AIR, Silverlight 1.1 ou JavaFX) ? Ensuite, il faut se méfier de l'intéropérabilité. Dans le cas des widgets, la compatibilité n'est pas toujours assurée. Ainsi un gadget Google ne fonctionnera pas tel quel sous Vista ou sur Dashboard. Et un gadget Google Desktop ne fonctionne pas sur iGoogle… Bref, rien n'est jamais simple. Si pour l'utilisateur final, les questions sont moindres, pour un portail, une application desktop, une application mobile, il faudra bien définir ses besoins de mashup et vérifier la faisabilité selon ses propres scénarios. Pour l'utilisateur, les avantages sont indéniables. On imagine parfaitement faire de la Business Intelligence, des contrôles de trafics, un accès à un SAP, etc. Par contre, pour le développeur et l'architecte, il faut tout d'abord choisir sa plate-forme puis développer les composants. Mais quelle architecture choisir ?

Quelques outils

- **Yahoo Pipes :** outil en ligne pour créer rapidement des mashup. Très visuel et assez pratique à utiliser. Fonctionne avec la majorité des navigateurs. Pratique pour agréger des informations, données. Les objets sont cependant un peu limités.
- **Microsoft Popfly :** outil en ligne permettant de créer des pages web, des mashup et des communautés. La partie mashup fournit une interface graphique très soignée avec de nombreux objets et effets visuels.
- **Kapow Mashup Server :** serveur pour créer et déployer des mashup. Se connecte aux sources de données de l'entreprise, le serveur (avec RSS, Atom, Rest, etc.) et la publication et le déploiement avec des portails, des constructions tiers de mashup, des applications, etc.
- **Google Mashup Editor :** éditeur basé sur Ajax pour créer des mashup. Actuellement en préversion limitée. D'autre part, on dispose de Google Mapplets permettant d'afficher des éléments dans une carte Google Maps. Ce sont en quelque sorte des gadgets.

Pour voir quelques possibilités de mashup, de nombreux exemples avec Google Maps : http://googlemapsmania.blogspot.com/

François Tonic

Le téléphone 2.0 : une timide réalité !

Dans ce dossier, les lecteurs les plus avertis s'étonneront peut être de la grande discrétion de la mobilité. Il faut avouer que les éditeurs la ciblent peu ou discrètement. Cependant, les nouvelles générations de téléphones portables, telle que le iPhone, permettent aujourd'hui de se mettre totalement, ou presque, au web 2.0.

Tout d'abord, sur mobile, les technologies de type RIA ne sont pas encore présentes (ou trop limitées). Par exemple, Adobe AIR ne peut fonctionner car la version de Flash Lite n'est pas à niveau de Flash 9, indispensable à AIR. Concernant Silverlight, pareillement, si Microsoft a prévu une version mobile (uniquement Windows Mobile), pour le moment, le développeur ne peut pas faire grand-chose, surtout que .Net 3.0 n'est pas présent sur les terminaux mobiles, donc pas de WPF. Par contre, on parle de plus en plus d'Ajax mobile. Opera fut un des premiers sans doute à en parler et à le mettre en pratique avec les Opera widgets. Cependant, si le concept Ajax Mobile existe, il ne faut pas oublier que nous sommes dans un contexte mobile, donc, avoir une interface adaptée. Il faudra, mais cela n'est pas nouveau, tester sur les différents navigateurs et systèmes, on connaît tous les problèmes d'affichages… On retrouve les contraintes traditionnelles : taille de l'écran (attention là aussi à prendre en compte les différences de tailles et l'affichage paysage ou portrait…), les performances et la mémoire (même si les derniers mobiles s'en sortent plutôt bien), l'interactivité utilisateur – application diffère d'un ordinateur. Il n'y a pas de souris par exemple et enfin, il y a la connectivité réseau (question : faut-il gérer ou non le mode connecté, déconnecté ?).

De plus, pour faciliter l'usage, il faudra éviter d'utiliser des frameworks, plates-formes nécessitant un runtime, une installation locale. Parmi les navigateurs cibles on peut citer Pocket IE (Windows Mobile, Opera, les navigateurs basés sur WebKit (moteur open source initié par Apple) notamment chez Nokia et Apple (Safari mobile de l'iPhone). Vous l'aurez compris, pour faire de l'Ajax mobile, il faut redévelopper un site et utiliser un framework adapté et compatible mobilité ! Elle constitue donc un enjeu vital pour les concepteurs / éditeurs de frameworks. Ce n'est pas un hasard si en 2006, Nokia et Backbase annonçaient une collaboration sur l'Ajax mobile. Si aujourd'hui Java ME (Micro Edition) connaît un bon succès, l'apparition d'Ajax peut supplanter au moins partiellement Java sur les terminaux mobiles avec un modèle de développement plus simple. Il faudra voir ensuite comment accéder aux services, fonctions du téléphone comme on peut le faire en Java, .Net, etc.

Java défie Ajax !

Concernant Java, JavaFX Mobile est taillé, comme son nom l'indique, pour les terminaux mobiles. On dispose d'API, d'un kit d'interface, d'un moteur graphique, de framework media et téléphonie, d'un système de mise à jour, etc. Pour ne pas casser avec l'existant, il supporte Java ME. Sun a par ailleurs en mai dernier racheté la technologie SavaJE, conçue pour la mobilité. D'autre part, sur la partie RIA, Sun et Laszlo System travaillent ensemble pour que Laszlo (le clone de Flash) soit accessible en environnement Java ME !

XUL de Mozilla est timidement présent sur la plate-forme mobile avec des projets tiers. On citera par exemple thinlet, un kit pour interface graphique en Java et XUL. Il existe aussi Bambookit GUI.

Alors mobile ou pas mobile le web 2 ?

À l'heure actuelle, avouons que la mouvance web 2.0 arrive timidement sur les terminaux mobiles. Le principal problème est l'absence de runtime adapté, de kits de développement. Cependant, les premiers frameworks Ajax ou encore JavaFX côté RIA, permettront sans doute d'amplifier dans les mois à venir l'arrivée des applications riches et web 2 sur son téléphone. Le iPhone d'Apple sera sans aucun doute un facteur important avec l'arrivée massive de gadgets (même si Nokia le propose déjà). Steve Jobs l'a d'ailleurs explicitement indiqué : les applications tiers sur iPhone seront des gadgets (javascript, xml, css, etc.) pouvant appeler des services systèmes tout en fonctionnant dans le navigateur mobile Safari (pour une question de sécurité).

En quête de standard

Le W3C travaille aussi de son côté sur la définition du futur mobile avec l'OpenAjax Workshop on Mobile Ajax (http://www.w3.org/2007/06/mobile-ajax/), événement se tenant le **28 septembre prochain à Santa Clara (USA).** Le but est de parler des défis d'Ajax sur mobile, d'explorer les possibilités, les limites, comment aider et améliorer le support et le développement Ajax sur les navigateurs mobiles. Au printemps dernier en Corée du Sud, le W3C par l'intermédiaire de Philipp Hoschka, à l'occasion du Mobile Web 2.0 Forum,a dévoilé les axes de réflexions : améliorer l'expérience utilisateur, améliorer le contenu et la participation, avoir des environnements interopérables, créer du mashup et une convergence des applications. À l'instar du desktop et navigateur, quid de l'accessibilité et du référencement (sans compter la sécurité) ? Sur ces points, les réponses demeurent timides et floues. Et l'interopérabilité, la compatibilité des applications 2.0 quelle que soit la plate-forme mobile ne sont guère d'actualité (tout du moins en ce moment).

Une bonne synthèse du sujet : http://www.thenewworld.fr/post/2007/06/08/FAQ-sur-Ajax-Mobile

François Tonic

2007 : l'année du RIA

Qu'est ce que le concept du « Rich Internet Application » (ou RIA), comment évolue cette offre sur le marché ? C'est ce que nous allons développer dans les lignes qui vont suivre, puis nous dresserons un tableau comparatif des différentes technologies mises à la disposition des professionnels pour réaliser des applications RIA et nous aborderons quelques exemples concrets.

Les Rich Internet Applications correspondent à une terminologie inventée par Macromedia en Mars 2002 : « A l'époque, l'idée était d'aligner un outil de rendu Flash avec un back office de données » explique Frédéric Massy responsable marketing entreprise Adobe. Depuis, d'autres éditeurs ont repris l'expression, comme Adobe, Microsoft, BEA, etc. Les RIA sont des applications web qui tendent à ressembler aux applications classiques (applications dites desktop ou de bureau). Les RIA favorisent la fusion des logiciels traditionnels et les logiciels client-serveur de type internet. La vitesse d'exécution et les interfaces riches sont les principaux aspects reconnaissables et caractéristiques de ce type d'application.

Comment fonctionnent les RIA ?

Comme énoncé ci-dessus, les applications RIA sont des applications qui s'exécutent du côté du client. Elles utilisent les ressources du processeur du poste client. De ce fait, elles permettent d'orchestrer aisément des interfaces utilisateur en temps réel. Les applications RIA doivent être *pensées client* afin de minimiser le nombre d'allers-retours vers le serveur des différentes informations. Ceci dans le but de favoriser les traitements côté interface. Les applications RIA demandent à être pensées et architecturées (souvent basées sur les services de type SOA) quelque peu différemment des applications web classiques mais ceci permet aux RIA de :

- Ne faire aucune installation pour exécuter l'application. Pas de problématique de mise à jour de l'application, c'est le principe des applications web.
- Les utilisateurs peuvent exécuter leur application depuis n'importe quel poste client équipé d'une connexion internet et d'un navigateur internet.
- Diminuer le trafic sur le réseau

Les RIA, une réalité du marché ?

Selon le Groupe Gartner, 30% des nouvelles applications seront développées sur les architectures RIA et 60% des nouveaux systèmes l'intègreront au moins partiellement d'ici 2010. Cette enquête laisse donc présager un bel avenir à cette « technologie » mais plus précisément, à cette façon d'architecturer vos applications. En effet, partir sur ce type d'application et d'architecture 3 tiers peut s'avérer très rentable. La notion d'orienté service (SOA) permet de séparer la logique métier de la logique « rendue » de votre application. Cela laisse la possibilité à vos clients de faire le choix de l'interfaçage de leurs données.

Les offres RIA du marché professionnel

Ajax, Flex, Open Laszlo, Silverlight, JavaFX, XUL/E, Wazaabi… sont toutes des technologies permettant de réaliser et de développer des applications de type RIA. Nous avons décidé de mettre en avant dans notre propos les technologies les plus révélatrices du marché mais aussi celles qui laissent présager un avenir certain aux applications RIA.

Les technologies que nous allons étudier dans la suite de cet article sont représentatives des grandes tendances actuelles, à savoir : Silverlight, Flex et JavaFX.

Microsoft Silverlight

Silverlight est une technologie issue des laboratoires de Microsoft, et s'installe sous la forme d'un plug-in (4mo) multinavigateur (Internet Explorer, Firefox, Safari) et multi-plate-forme (Windows & MacOS X). Silverlight fonctionne avec n'importe quelle technologie ou plate-forme Web d'arrière-plan, s'intégrant de manière transparente aux infrastructures et applications existantes, y compris Apache et PHP, ainsi que JavaScript et XHTML côté client.

Modèle de programmation

Silverlight se décompose en deux parties : la partie interface et la partie code. La partie interface est orchestrée par le langage XAML (eXtensible Application Markup Langage). Il s'agit d'un langage de description d'interface qui permet de créer des interfaces riches (animations, triggers,…) de façon déclarative, introduit initialement par son grand frère : Windows Presentation Foundation. Pour ce qui est de la partie code, il faut distinguer deux versions de Silverlight : une version bêta 1.0 qui donne la possibilité de coder en JavaScript et la version alpha 1.1 de Silverlight qui permet de programmer avec les technologies .NET, ce dernier embarquant une CLR minimale qui permet d'exécuter du code côté client. Quelle que soit la version du plug-in, il faut savoir que Silverlight n'embarque pas encore de contrôles natifs tel que le « Button », la « Listbox », etc. Par conséquent, vous devrez vous-mêmes créer vos propres contrôles et vos bindings à la main.

Les outils

La version finale n'étant pourtant pas encore divulguée, Silverlight possède néanmoins déjà une panoplie d'outils très intéressants :

Pour les développeurs, **Visual Studio (Orcas)** vous permet d'ores et déjà de créer et débugger des applications Silverlight pour le plug-in en version 1.0 mais aussi en 1.1.

Expression Blend, quant à lui, est dédié au public des graphistes et des intégrateurs, il vous permet de créer très simplement et rapidement des interfaces riches via le langage XAML.

Expression Media Encoder permet de réaliser soi-même son propre lecteur multimédia en Silverlight (avec un déploiement simplifié).

Exemple d'application Silverlight

Une application Silverlight 1.0 est composée au minimum de 3 fichiers :

- 1 fichier HTML qui va permettre d'injecter via JavaScript et balise OBJECT le plug-in Silverlight.

- 1 fichier JavaScript appelé « Silverlight.js » ce fichier sert entre autres à gérer les compatibilités entre les navigateurs et les différentes versions du plug-in.
- 1 fichier XAML qui va permettre d'afficher l'interface d'un point de vue graphique de l'application.

Pour communiquer entre le JavaScript et le fichier XAML il faut renseigner les fonctions JavaScript au sein du document XAML, par exemple :

```
<Canvas xmlns=»http://schemas.microsoft.com/client/2007» xmlns:x=»http://schemas.microsoft.com/winfx/2006/xaml» x:Name=»parentCanvas» Loaded=»setClockTime» >
```

Cela aura pour effet de déclencher les fonctions une fois l'évènement en cours. Dans notre exemple, lorsque le Canvas sera entièrement chargé, la fonction « setClockTime » sera appelée. D'un point de vue JavaScript, il suffira d'utiliser la méthode « findName » pour accéder aux éléments XAML.

Pour modifier une propriété d'un élément XAML vous pourrez via le JavaScript renseigner la nouvelle valeur de cette propriété, voici un exemple pour une animation :

```
control.content.findName(«animation»).from = angle.toString() ;
control.content.findName(«animation»).to = angle.toString() ;
```

Pour ce qui est du plug-in Silverlight 1.1, Visual Studio Orcas et Expression Blend, généreront directement un fichier DLL qui permettra au plug-in d'exécuter votre code .NET du côté du client. Pour ce qui est du traitement des évènements celui-ci s'aborde exactement comme un projet .NET classique.

Côté serveur

Silverlight propose un support de vidéo multimedia dont le contenu peut-être hébergé avec l'aide du service Silverlight Streaming by Windows Live. Ce service permet aux développeurs et aux designers de délivrer du contenu riche dans leur application, à moindre coût, avec pour le moment 4Go par compte et une limite de durée des vidéos fixée à 10 minutes. A noter que vous pouvez utiliser IIS 7 pour faire de même dans votre environnement d'entreprise.

Adobe Flex

Flex est une solution de développement créée par Macromedia en 2004 puis reprise par Adobe en 2006, permettant de créer et de déployer des applications Internet riches multi- plates-formes. Le 26 avril 2007, Adobe a annoncé choisir la licence open source MPL pour sa solution de développement Flex.

Modèle de programmation

A l'instar de Silverlight, son modèle de programmation fait appel à deux langages :

- L'interface, avec MXML (un dialecte XML), permet la réalisation d'interfaces utilisateur interactives avec son approche déclarative.
- Pour le code, ActionScript 3.0. Il repose ECMAScript, et est un langage orienté objet (cette version 3 a été particulièrement remaniée).

Pour sa partie présentation, Flex utilise le plug-in Flash (on estime aux alentours de 97% le nombre de machines équipées du Flash Player), ce qui rend par conséquent les applications Flex multi-plates-formes et facilement déployables.

Les outils

Flex Builder : Adobe propose son propre IDE basé sur Eclipse. Appelé « Macromedia Flex Builder » cet IDE vous permet de développer vos applications Flex mais aussi de les designer. Cet outil est dédié aux développeurs et aux designers. Il vous permet d'accéder au Flex Framework et de skinner vos applications Flex.

Flex SDK : il s'agit ni plus ni moins que du compilateur qui est entièrement gratuit et téléchargeable en ligne.

Exemple d'application Flex

Voici un exemple concret d'utilisation Flex. Cet exemple montre qu'il est possible d'intégrer du code ActionScript 3.0 directement dans la page MXML.

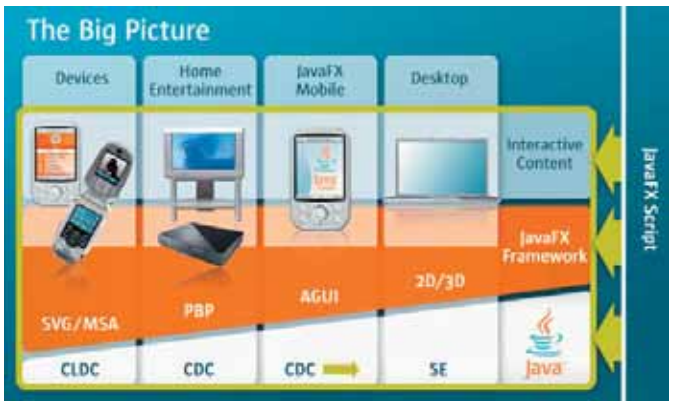
```
<?xml version=»1.0» encoding=»utf-8»?>
<mx:Application xmlns:mx=»http://www.adobe.com/2006/mxml» layout=»absolute»>
  <mx:Script>
    <![CDATA[
      public function bonjour():void {
        monMessage.text = «Bonjour « + maTextbox.text;
      }
    ]]>
  </mx:Script>
  <mx:Panel x=»20» y=»20» width=»500» height=»300» layout=»absolute»>
    <mx:Label x=»10» y=»10» text=»Pseudo :»/>
    <mx:TextInput x=»10» y=»30» id=»maTextbox»/>
    <mx:Button x=»150» y=»30» label=»Go !» click=»bonjour()»/>
    <mx:Label x=»10» y=»60» id=»monMessage»/>
  </mx:Panel>
</mx:Application>
```

Côté serveur

Flex propose une solution de connectivité aux données et de déportation de la logique métier côté serveur. Cette solution de middleware porte le nom de Flex Data Services et s'appuie sur une application JavaEE pour délivrer des services de synchronisation automatique des données entre le client et le serveur, de push de données, … Un service qui peut vite se révéler indispensable dans des projets d'envergure.

Sun Java FX

JavaFX est une nouvelle famille de produits que Sun destine au marché du contenu riche sur desktop et mobile. Lors du JavaOne 2007, Rich Green a introduit la technologie JavaFX Script en exprimant clairement la volonté de tenir tête à Flash et Silverlight.



Au cœur de cette initiative, se trouve le langage JavaFX Script permettant aux développeurs de contenu de créer des médias riches et du contenu pour des déploiements en environnement Java.

Le modèle de programmation

Anciennement nommé F3, JavaFX Script est un nouveau langage de script conçu spécifiquement pour les applications RIA et clients lourds. JavaFX Script est un langage typé possédant une syntaxe déclarative, avec inférence de type, un système de liaison automatique des données et un support pour les graphiques 2D (et composants Swing) et les animations. On notera néanmoins l'absence de l'intégration de multimédia (vidéo/audio) et de graphismes 3D, dans une moindre mesure, pourtant présents dans différentes technologies concurrentes.
Sa syntaxe déclarative, mais qui reste très proche de celle du langage Java, lui apporte une flexibilité et une souplesse par rapport à une syntaxe orientée XML. Par exemple, ce listing permet de créer une fenêtre pleinement fonctionnelle avec un bouton qui affichera un texte dans la console lors de l'appui sur celui-ci:

```
import javafx.ui.*;
import java.lang.*;

Frame {
....title: «JavaFX Script»
....content: Button {
........text: «Cliquez ici !»
........action: operation(){
.................System.out.println(«Bouton cliqué !»);
........}
....}
....visible: true
}
```

L'une des forces de JavaFX Script est sa possibilité d'exploiter nativement les APIs Java. Vous pouvez importer des classes Java, créer de nouveaux objets Java et implémenter des interfaces Java de façon naturelle dans JavaFX Script. Le fait que le langage possède des fonctionnalités d'encapsulation et de réutilisation (packages, classes, héritage, unités de déploiement et compilation séparée) rend possible la création et la maintenance d'applications importantes utilisant la technologie Java, ce qui est un grand avantage pour l'intégration de ce composant dans des projets déjà existants.
La liaison des données avec JavaFX Script est rendue très simple, souple et puissante avec le mot clé bind comme le montre l'exemple suivant qui affiche dans un Label, le contenu d'un Textbox :

```
var lbl = SimpleLabel {
    ...
    text: bind txt.value
}
```

Les outils

Sun met à disposition un plug-in pour Eclipse et un plug-in pour Netbeans 5.5 et 6.0. Tous deux peuvent être téléchargés sur le site OpenJFX, et permettront aux développeurs de rapidement créer leur première interface avec JavaFX.

Conclusion

JavaFX a tout juste été annoncé lors du JavaOne 2007, par conséquent les outils et les projets utilisant cette technologie -qui on l'espère évoluera dans une prochaine version revue en profondeur avec une plus grande interaction 2D/3D/Multimédia – restent encore au stade expérimental, tout comme la technologie. Cependant, les caractéristiques du langage JavaFX Script font de JavaFX une technologie à fort potentiel, fonctionnelle et productive pour les développeurs.
On attendra donc impatiemment de voir l'évolution du produit et des applications, ainsi que de l'écosystème gravitant autour de cette technologie qui semble très prometteuse.

Conclusion générale

Il existe un grand nombre de technologies qui vous permettent de créer des applications RIA. Chacune possède une approche différente au niveau de la définition de l'interface, par exemple JavaFX avec JavaFX Script et Silverlight avec XAML, et des objectifs différents qui se discernent au vu des fonctionnalités proposées par chacune.
Le tableau suivant permet d'avoir un aperçu général des technologies étudiées dans cet article :

	Multi plateforme (1)	Multimédia	Vectoriel	Accès système	Accessible hors ligne	Installation	Langage
Silverlight 1.0 (2)	Oui (sauf Linux)	Oui	Oui	Non	Non	Plug-in (2 Mo)	XAML, javascript
Silverlight 1.1	Oui (sauf Linux)	Oui	Oui	Oui	Non	Plug-in (4Mo)	XAML, .NET
JavaFX	Oui (+ mobile)	Non	Oui	Oui	Oui	Oui	JavaFX Script
Flex	Oui	Oui	Oui	Non	Non	Plug-in	ActionScript 3.0 et MXML

Références :

OpenJFX : https://openjfx.dev.java.net/
Silverlight, le site communautaire officiel : http://silverlight.net/Default.aspx
Laboratoires Adobe Flex : http://labs.adobe.com/technologies/flex/

■ **Guillaume André** - Intégrateur .NET, Designer
http://www.wygwam.com/ - http://www.wpfstorm.com/
■ **Julien Chable** - Consultant/Développeur

(1) Safari 3.0 Windows n'est pas pris en compte, car sorti trop récemment.
(2) La version mobile de Silverlight est prévue courant 2007.

Les limites du web 2.0

Le web 2.0 est un concept arrivé en France fin 2005. Celui-ci recouvre de nouveaux usages souvent associés à des technologies remises au goût du jour.

Les sites web 2.0, généralement reconnaissables à leurs sigles "version bêta", sont centrés sur l'utilisateur. Ils permettent d'afficher un contenu plus riche, plus interactif et plus ergonomique, ce qui permet de rapprocher l'interface limitée HTML classique de l'interface client lourds.
A ce jour les sites " web 2.0 " sont souvent des sites proposant des services (hébergement des favoris, des vidéos, comparaison de produits, etc.).
Le principal reproche fait par les développeurs au web 2.0 porte sur son aspect marketing. D'un point de vue technique il n'apporte pas de nouveautés technologiques, ce serait un terme créé par et pour le marketing. On peut distinguer deux types de limites au Web 2.0 selon que l'on considère les technologies employées ou les usages impliqués.

Les contraintes techniques : limites d'AJAX

L'une des principales nouveautés de l'approche Web 2.0 consiste à utiliser massivement Ajax pour permettre le rafraîchissement d'une partie de la page HTML plutôt que son intégralité. D'une couche d'interface HTML on passe à une couche d'interface utilisant massivement Javascript pour communiquer avec le serveur. Certaines limitations viennent donc directement d'Ajax. L'avenir et les autres technologies qui pointent le bout de leur nez (Flex, XUL, ...) nous diront si le Web 2 offrira autant de facilité que les clients lourds.

L'accessibilité

L'approche traditionnelle permettait d'associer une page à une action. Ainsi il était facile de stocker son adresse et d'y revenir (à quelques exceptions près). L'approche Ajax où tout se fait sur la même URL ne facilite pas cette accessibilité. Tout utilisateur qui souhaite ajouter la page à ses favoris est pénalisé, car il n'obtiendra pas les résultats de ses manipulations en favori, mais juste l'adresse du site web. Google Maps propose un lien "Obtenir l'adresse de la page" qui remplit la barre d'adresse et permet d'enregistrer ses cartes annotées dans les favoris. De la même façon, les boutons *précédent* et *suivant* ne remplissent plus leur fonction première, la navigation s'en trouve rompue.
De plus, toutes les pages de résultats créées par les choix de l'utilisateur sont masquées aux moteurs de recherche. On retombe dans les mêmes travers qu'avec les frames, tant au niveau de l'accessibilité que du référencement des pages.
L'utilisation abusive du Javascript peut aussi être source d'une perte d'accessibilité. Tout baser sur les événements de souris, notamment pour les formulaires, est une mauvaise idée. Un internaute peut valider un formulaire en cliquant sur le bouton 'submit', mais il peut aussi choisir d'y aller par les tabulations, puis de valider par la touche Entrée. Bref le Web 2 pour l'ergonomie, oui, mais avec modération. Mais modération et Web 2 ne vont pas très bien ensemble !

Dépendance au réseau

Si le gain en terme d'ergonomie est potentiellement important qu'en est-il si votre connexion internet n'est pas disponible ? Votre application est inaccessible, or malheureusement, il n'est pas rare que la bande passante vienne à manquer. Cette problématique est extrême dans le cas d'applications " professionnelles ". Peut-on imaginer travailler sans avoir accès à son Webmail ? Quid de ses prises de rendez-vous si votre calendrier n'est pas accessible ? On peut imaginer une version déconnectée avec une synchronisation régulière, mais on tombe tout de suite dans une complexité importante. Conscient de cette limitation, Google dont le modèle est quasi exclusivement basé sur le Web vient de dévoiler une API permettant de travailler aussi bien connecté / déconnecté (Google Gears).

Sécurité

Le Javascript est lisible partout côté client, donc déplacer une partie de la logique d'interaction, voire une partie de la logique métier dans le Javascript ne doit pas être fait à la légère. De plus, avec un envoi multiple de requêtes par pages, vous accroissez la surface d'attaque pour un hacker, puisque les interactions du site avec le navigateur sont multipliées. Du côté serveur, on veillera donc à continuer le filtrage des entrées, que les requêtes transitent en GET ou en POST.

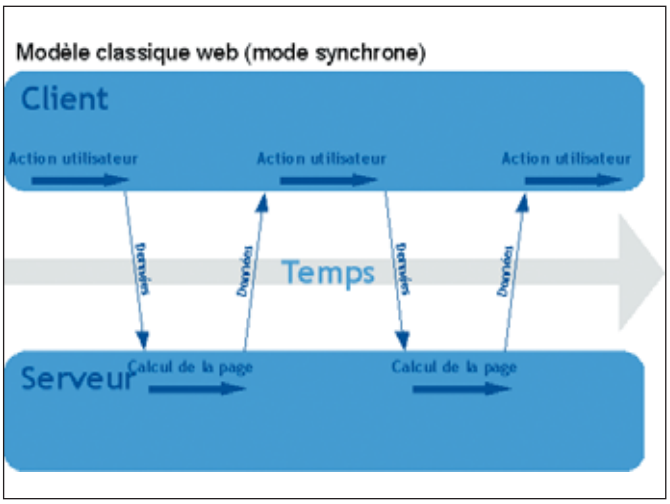
La charge serveur

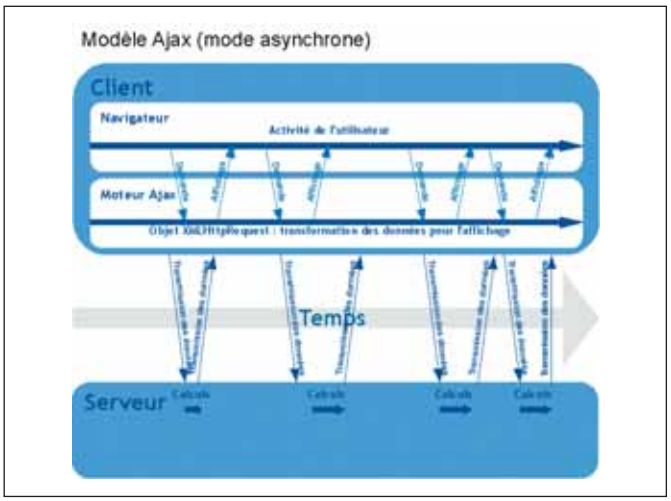
Par ailleurs, la charge serveur est un problème à garder sous contrôle. L'approche page par page impliquant n demandes générant une charge m se voit remplacée par une approche impliquant beaucoup plus de " petites " demandes. Les différents navigateurs n'ont pas le même comportement par défaut pour la gestion du cache. Par défaut, Firefox ne met rien en cache, alors qu'Internet Explorer et Opéra mettent la réponse en cache pour les requêtes en GET.

Les contraintes non-techniques

Standard d'échange ?

Vous avez entendu parler du stockage de favoris sur des sites web, par vos collègues. L'un d'entre eux, très enthousiaste utilise del.icio.us (racheté par Yahoo!), vous recherchez rapidement le site, vous vous inscrivez et





commencez à collecter vos URLs favorites sur le site. Un autre collègue vous indique qu'il utilise un autre site, et que certains liens peuvent vous intéresser, il vous propose de vous les envoyer. Chaque site de ce type propose en effet un marquage "pour" un autre utilisateur à condition qu'il soit sur le site.

Les outils sont multiples, comment savoir lequel choisir ? Et si je souhaite être en réseau avec mes collaborateurs, comment choisir ? Comme pour la messagerie instantanée, la force d'attraction d'un réseau est proportionnelle au nombre de mes connaissances qui y sont raccordées (effet de réseau ou effet-club). On peut souvent exporter et/ou importer des liens, au format RSS entre différents services (del.icio.us vers Blogmarks, par exemple). Par ailleurs, pourquoi, si je veux veiller sur tous ces liens, dois-je me déclarer auprès des différents services ? Pourrait-on imaginer, comme le suggère Karl Dubost(1) avoir un outil qui stocke nos liens dans un format ouvert et qui nous identifie sur tous les réseaux de favoris disponibles ?

Confidentialité des données

Ce point nous amène à un autre problème qui touche en particulier les traitements de texte en ligne et les webmails : la confidentialité des données stockées. Quand vous souscrivez à Gmail (par exemple), ou quand vous utilisez un traitement de texte en ligne (éventuellement collaboratif) avez-vous confiance dans le prestataire chez qui vous laissez vos informations ?

Qualité du contenu

Un autre reproche qu'on peut faire à ces nouveaux sites de partage de vidéos, de liens favoris ou d'avis sur des produits est une " perte d'auteur ". Les rédacteurs d'un journal sont qualifiés pour retransmettre l'information (vérification des sources, traitement équilibré des thèses en présence), ils connaissent les astuces pour influencer, mais également la limite entre information et publicité. Ils sont aussi responsables de ce qu'ils écrivent. Eu égard à ces différents aspects, les sites web 2.0 où les utilisateurs prennent la parole ont-ils un contenu fiable ? Ces sites qui redonnent la parole à l'utilisateur ne produisent pas de contenu. Quand vous hébergez une vidéo sur DailyMotion, c'est vous qui créez le contenu. Si en plus vous le taggez avec des mots-clefs et notez les contenus des autres internautes, vous participez de plus en plus à l'enrichissement du contenu du site web, DailyMotion se contentant de fournir l'infrastructure nécessaire. C'est vous qui en créez le contenu, mais c'est le site qui en retire les bénéfices.

Conclusion

Si on peut voir de la démocratie dans les votes utilisateurs sur les sites web 2.0 et la reprise en main du web par les internautes, il est nécessaire de nuancer la qualité des contenus ainsi mis en avant.

En termes de sécurité, pour Ajax, les interfaces riches ou Web 2 en général, ne doivent pas masquer les principes de sécurité. Les développeurs doivent tester toutes les données reçues en entrée. Pour être réussie, une application Web 2.0 doit être pensée, non en terme de Web2.0, mais pour l'utilisateur final. Il faut choisir vos technologies en fonction de ce qu'elles apportent et non pas " pour être web 2.0 ".

Sarah Haim-Lubczanski
Formatrice "Ajax, Web 2.0 et Javascript" chez Anaska (www.anaska.com) et modératrice sur les forums de PHPFrance

(1) : http://www.la-grange.net/2006/07/24.html (Carnet Web de général en chef)

BOUTIQUE BOUTIQUE BOUTIQUE BOUTIQUE

Achetez les magazines,
les articles en PDF et abonnez-vous en ligne
www.programmez.com

Evoluer vers le web 2.0 : tout ou partie ?

On parle tellement du web 2.0 ces derniers mois que la tentation peut se faire forte parfois d'actualiser son site en y plaçant des gadgets Ajax. Mais il faut prendre garde à ne pas aller ni trop vite ni trop loin, car tout n'est pas bon à prendre dans le monde merveilleux du web 2.0. Au-delà de l'aspect technique, les notions d'interactivité avec les visiteurs sont au centre de l'idée même de web 2.0. Voyons ce qui peut être bon ou non à implémenter selon les situations.

La première grande idée du web 2.0 est l'implication du visiteur dans la vie et le contenu même du site. Ceci entraîne un changement d'optique dans la conception d'un site moderne, ainsi qu'une évolution de vocabulaire. En effet, on peut aujourd'hui rapprocher les sites des applications web, et donc les visiteurs (passifs) des utilisateurs (actifs). Mais pour autant, certains sites doivent, par essence, conserver un caractère informatif, et il serait nuisible à leur efficacité d'y intégrer trop d'interactivité artificielle, c'est-à-dire trop de gadgets superflus.

En revanche, chaque fois qu'il est question de faire intervenir l'utilisateur, il est souvent possible d'améliorer la qualité du service qui lui est proposé grâce aux concepts définis par le web 2.0. Ces améliorations s'expriment sous deux formes : l'une technique, le recours à Ajax notamment pour rendre l'interface utilisateur plus intuitive et ergonomique, l'autre fonctionnelle, en autorisant l'utilisateur à se rapprocher de vos processus, c'est-à-dire en partageant avec lui plus d'informations sur ce qui se passe au-delà de ses propres actions. Il s'agit en un mot de rendre vos relations avec les utilisateurs plus transparentes. Le site web ne doit plus être un hygiaphone, mais un espace ouvert de discussion !

Les modifications techniques

Bien sûr, la face visible de l'iceberg reste l'interface utilisateur. Intégrer des mécanismes Ajax va sans aucun doute rajeunir un site. Et dans bien des cas lui apporter une réelle plus-value en terme de confort d'utilisation. Nous allons voir dans le détail un exemple typique : utiliser Ajax pour mettre à jour une partie seulement d'une page web.

Depuis les tout débuts du web, les concepteurs de site ont cherché à limiter les rechargements de pages, pour deux raisons : éviter des temps de chargements trop longs et rendre l'interface plus agréable, plus proche des applications traditionnelles en réalité. A une époque très reculée, dans la lointaine galaxie du HTML 2.0, il était commun de recourir aux frames pour pouvoir actualiser seulement une partie de l'écran. Mais les problèmes que cela posait pour indexer les contenus dans les moteurs de recherche ainsi que le manque de souplesse de cette technique, les frames obligeant à délimiter des sections au cordeau dans la page, ont rapidement amené à son abandon quasi total.

Ajax permet de résoudre élégamment ce problème. Car vous pouvez désormais actualiser un calque, un tableau, une ligne de tableau, une cellule seulement, etc. selon vos besoins. Voyons un exemple avec un moteur de recherche interne. Avec Prototype, apporter cette modification est véritablement simplissime :

Code original

Page de recherche find.html :

```
<!- affichage du corps de la page ->
<form method='POST' action='search.php'>
<input type='text' name='keyword'> <input type='submit' value='Rechercher'>
</form>
<!- affichage du pied de page ->
```

Page d'affichage des résultats search.php :

```
<?php
// affichage du corps de la page

// on récupére les résultats en fonction de $_POST['keyword']
$query = " ... "
$result = mysql_query($query);
while($row = mysql_fetch_object($result))
{
   echo " $row->titre - $row->description<br><br> ";
}

// affichage du pied-de-page
?>
```

Code " ajaxisé "

Page de recherche find.html :

```
<!- chargement de prototype ->
<script type='text/javascript' src='/js/prototype.js'></script>
<script type='text/javasrcipt'>
function search()
{
   // récupération du mot-clef dans le champ texte
```

```
    // $F est une fonction Prototype qui retourne la valeur du champ correspondant
    var keyword = $F('keyword')

    // validation de sa valeur
    if(keyword.length < 3)
    {
        alert('Le terme de recherche doit faire au moins 3 caractères')
        return false
    }
    else
    {
        // affichage d'un message
        Element.update('search_results', 'Recherche en cours...')
        // envoi du formulaire et récupération des résultats
        // en un seul appel de fonction Prototype
        new Ajax.Updater('search_results','search.php',{parameters: { keyword: keyword }
}

    }
}
</script>
<!-- affichage du corps de la page -->

<input type='text' name='keyword'> <input type='button' value='Rechercher'
onclick='search()'>
<!-- calque vide, pour l'affichage des résultats de recherche -->
<div id='search_results'></div>
<!-- affichage du pied de page -->
```

Et voilà, c'est presque fini : il ne reste plus qu'à retirer de search.php le code correspondant à l'affichage du corps et du pied de la page, car ceux-ci sont déjà chargés par find.html. Tout ce que retourne search.php est désormais automatiquement chargé dans le calque div#search_results. A chaque clic sur le bouton de recherche, le contenu de ce calque sera remplacé par les résultats de la nouvelle recherche. Ce mécanisme pourrait en fait être étendu à toutes les pages du site, pas seulement un résultat de recherche. Il est possible de donner en second argument de new Ajax.Updater l'URL d'un simple fichier HTML, dont le contenu sera alors chargé dans le calque spécifié en premier argument. Mais cela demanderait certains " bricolages " pour contourner les problèmes d'indexation des contenus, qui restent les mêmes qu'avec les frames, car les robots des moteurs de recherche sont incapables d'activer les fonctions Javascript nécessaires au chargement des contenus. Et donc, pour une URL donnée, le robot ne stockera qu'un contenu, alors que vous en avez prévu plusieurs. Ce problème peut être contourné en fournissant un sitemap aux moteurs comme Google, avec une liste d'URL en accès direct à vos contenus, et surtout ne pose aucun problème lorsqu'il s'agit d'un intranet ou d'un back office, qui ne nécessite pas d'être indexé. N'hésitez donc pas à employer cette méthode, qui permet d'accélérer très significativement le confort d'utilisation du site, en réduisant de manière très importante les temps de chargement. La possibilité de communiquer avec le serveur grâce à Javascript, depuis une page déjà chargée, et d'intervenir directement sur cette page ouvre la voie à bien des astuces de ce type. Mais beaucoup sont nettement plus complexes à mettre en oeuvre, c'est pourquoi nous ne les passerons pas en revue ici d'un point de vue technique. Je vous renvoie pour

cela à l'article sur les frameworks Ajax de ce même numéro, qui vous donnera des pistes pour découvrir les autres possibilités en terme d'ergonomie que propose Ajax.

Les modifications fonctionnelles

C'est sans doute en apportant de nouvelles fonctionnalités à votre site que vous rattraperez le plus efficacement l'ère du web 2.0. En effet, il faut désormais se dire que l'utilisateur veut avoir son mot à dire, et veut être tenu informé ! Il ne se contente plus d'être un visiteur passif.
La première expression de cette tendance a été de donner la possibilité aux utilisateurs de sites, notamment commerciaux, de donner leur avis. Ainsi, ils participent au contenu, ils sont acteurs. Permettre aux utilisateurs de commenter des produits, de les noter, de les classer par tags, donc selon leur propre classement et non plus selon une hiérarchie figée de catégories, est sans doute le symbole le plus représentatif du concept de web 2.0.
Mais il est possible, voire nécessaire aujourd'hui, d'aller plus loin. Le web permet une très grande réactivité, et les utilisateurs veulent en profiter, sous peine d'être frustrés. Prenons un exemple : un simple formulaire de contact. A première vue, ce type de formulaire ne peut servir à l'utilisateur qu'à transmettre un message, comme une demande d'informations commerciales, à une entreprise. Et pourtant, il est possible d'en faire plus. De transformer cet outil simpliste, qui sert à la base à inciter les utilisateurs à entrer en contact avec l'entreprise par le biais d'un mail, en véritable outil de relation client.
Comment ? Simplement en remplaçant la page de type " Nous vous remercions blah blah " par une interface permettant à l'utilisateur de savoir où en est sa demande, si elle a été traitée ou non, et par qui, etc. et sur laquelle il peut revenir, modifier sa demande, consulter les archives du suivi de cette demande...

Conclusion

Le web a évolué ces dernières années. Des simples sites vitrines des origines, nous sommes passés aux boutiques en ligne. Il faut désormais ouvrir les portes de ces boutiques, laisser l'utilisateur pénétrer à l'intérieur, s'adresser à quelqu'un d'identifiable, avoir une relation humaine avec son interlocuteur. Le web 2.0, c'est aussi ça : faire du web un outil de communication entre humains, et non pas entre humains et machines seulement. A l'heure où de plus en plus d'entreprises mettent au bout de leurs lignes téléphoniques des serveurs vocaux, le web 2.0 va peut-être paradoxalement offrir un retour à une communication moins désincarnée entre utilisateurs et fournisseurs !

Gauthier Delamarre

La sécurité et le Web 2.0

Désormais, les applications Web 2.0 sont bel et bien une réalité. Elles ont connu un essor considérable ces deux dernières années. Les principales références en la matière sont nombreuses : Google maps, Google docs, YouTube, MySpace. Il est indéniable que les applications Web 2.0 présentent des fonctionnalités novatrices. Cependant, on peut remarquer qu'un point a réellement été omis durant cette adoption massive : la sécurité.

Cet article va présenter les différents composants d'une application Web 2.0 et décrire les principaux vecteurs d'attaque.

Les différents composants d'une application Web 2.0

Les applications de type Web 2.0 sont, de par leur nature, plus complexes et font intervenir de nombreux composants. La complexité peut être attribuée à différents facteurs :

- L'utilisation massive de JavaScript, ActionScript par des technologies telles qu'Ajax, Flash pour l'interface graphique, rend difficile la compréhension du code.
- Des échanges de données nombreux entre les différents composants. Une application peut très bien faire appel à des web services distants qui ont très bien pu être développés et mis à disposition par d'autres développeurs. De même, une application peut agréger le contenu de différents flux RSS ou de différents blogs. Les échanges de données peuvent avoir recours à XML ou JSON. De même, le simple protocole HTTP avec les paramètres GET et POST ne suffit plus, les applications ont désormais besoin de faire appel à SOAP, REST, XML-RPC.

Les principales attaques sur les applications Web 2.0

1) Cross-site Request Forgery (CSRF)

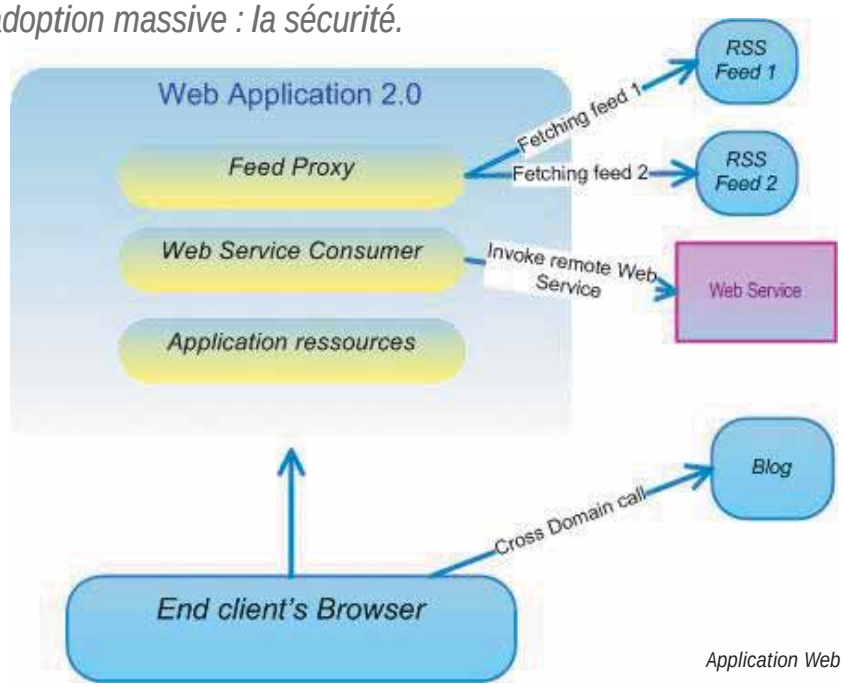
Contrairement aux attaques de types Cross-site Scripting (XSS) qui permettent d'exploiter la confiance qu'a un client envers le serveur, le CSRF consiste à exploiter la confiance que le serveur a envers le client.

1.1) Quelques rappels sur XSS

Concernant XSS, les attaques consistent à injecter du code malicieux en exploitant une faille de sécurité. Par exemple, la fonction phpinfo des versions PHP 4.4.3 et 4.4.6 était vulnérable à ce type d'attaque. Phpinfo est une fonction qui affiche des informations détaillées sur l'environnement PHP en place. En exécutant le code suivant sur un serveur PHP, une pop-up JavaScript s'affiche :

http://www.privatewebsite.com/phpinfo.php?a[]=<script>alert(/XSS/);</script>

Afin de contrecarrer ce type d'attaques, il est nécessaire de faire du filtrage sur les paramètres saisis.



Application Web 2.0

1.2) CSRF

Pour le CSRF, la confiance que le client a envers le serveur, se manifeste par l'utilisation d'un cookie. Concrètement, l'attaque va consister à faire exécuter par l'utilisateur une action de manière involontaire qui sera utile à l'attaquant. Concrètement, avec un lien du type http://myprivatewebsite.com/Page1?action=removeContent, il va être possible pour un utilisateur de supprimer le contenu de la page1. Bien que cette action ne puisse être réalisée que par un utilisateur authentifié, elle peut très bien être exécutée par un attaquant à l'insu de l'utilisateur. Car cette action peut très bien être :

- envoyée dans un email,
- postée sur un site web
- utilisée au sein d'une attaque XSS pour ouvrir automatiquement la page.

Cette action n'a de sens que si l'utilisateur est déjà authentifié, afin de pouvoir profiter du cookie (qui symbolise la confiance que le serveur a envers le client). Les applications qui se basent uniquement sur la validité d'un cookie pour effectuer des opérations sont sujettes à ce type d'attaque.

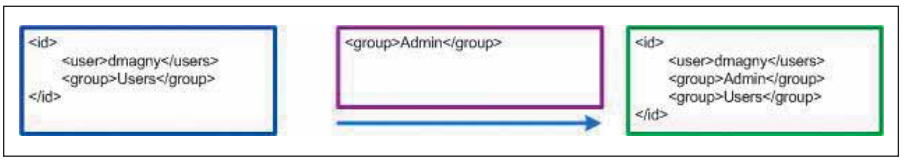
Ce type d'attaque n'est pas propre aux applications Web 2.0. Cependant, celles-ci peuvent faciliter grandement les choses. Prenons par exemple, un utilisateur qui s'authentifie sur son site bancaire, il obtient donc un cookie, il effectue un certain nombre d'opérations. Ensuite, il navigue sur d'autres sites web tout en étant encore connecté sur son site bancaire. L'utilisateur navigue sur un site où un attaquant a écrit du code Ajax silencieux qui va faire des appels sur le compte bancaire de l'utilisateur allant chercher par exemple des informations critiques, et les envoyant sur le site web de l'attaquant.

Concrètement, il n'existe pas de solution universelle pour protéger les applications de ce type d'attaques et elles sont difficiles à contrecarrer. Cependant des protections sont possibles :

- Vérification du header HTTP Referrer, option HTTP qui identifie l'origine de la requête. Cette protection sera utile pour contrecarrer les attaques par emails. Cette solution n'est pas fiable et fonctionnera uniquement dans certains cas. Car bien entendu il est très aisé pour un attaquant de modifier la valeur de ce header HTTP.
- Favoriser les requêtes HTTP de type POST aux requêtes de type GET. Les requêtes POST sont moins exposées car moins prédictibles, en effet dans le cadre de requêtes GET, les informations transitent dans l'URL ce qui n'est pas le cas des requêtes POST.
- Utilisation d'une clé temporelle, il s'agit de la seule méthode efficace. Elle consiste à utiliser un secret (token aléatoire en champ caché) qui sera envoyé au serveur lors de la soumission d'un formulaire ou d'actions critiques. Si l'url envoyée ne contient pas ce nombre aléatoire qui ne peut être prédit par l'attaquant, le serveur web refusera d'effectuer le traitement.

2) Attaques XML

Avec les applications de type web 2.0, du trafic XML est transmis entre le navigateur et le serveur (avec Ajax notamment). Une attaque peut consister à charger plusieurs fois des nœuds XML similaires pouvant provoquer un déni de services sur le serveur si le moteur XML de traitement présente une faiblesse. Des attaquants peuvent très bien produire des documents XML mal formés pouvant déranger les mécanismes de parsing utilisés côté serveur.



Exemple d'insertion de données

De même, il est tout à fait possible d'injecter de nouveaux formats de données.

Avec les applications de type Web 2.0, le recours aux Web Services est courant. Ceux-ci sont amenés à consommer du contenu XML au travers du protocole SOAP. On peut très bien imaginer qu'un attaquant remplace le contenu d'un noeud XML par une technique d'injection (LDAP, XPATH, SQL). De manière à pouvoir pallier ce type d'attaques, il est indispensable de toujours valider les données reçues de Web Services.

3) Injection RSS / Atom

Une application Web 2.0 peut agréger différents flux RSS, et les transmettre au navigateur d'un client de l'application. Il est possible d'injecter du code JavaScript dans les flux RSS afin de générer des attaques au niveau du navigateur du client. Un utilisateur final naviguant sur un site web particulier peut charger un flux RSS contenant du code Javascript malicieux. Le script peut très bien par exemple installer un logiciel, voler des cookies. Les flux RSS et ATOM faisant désormais partie intégrante des applications Web, il est indispensable de filtrer côté serveur un certain nombre de caractères avant la transmission des données à l'utilisateur final.

4) Attaques relatives au WSDL

WSDL (Web Services Definition Language) est, comme son nom l'indique, un langage de description de Web Services au format XML). WSDL propose une interface qui liste l'ensemble des opérations disponibles au niveau du Web Service. En consultant le fichier résultant, il va être aisé de connaître l'ensemble des méthodes exposées ainsi que leur signature. Bien souvent les fichiers WSDL sont générés de manière automatique par des outils. Il est donc tout à fait possible que par inadvertance une méthode se retrouve exposée alors qu'elle ne le devait pas. Il est donc important de vérifier le contenu des fichiers WSDL générés et des méthodes exposées. De plus, il est aussi recommandé de limiter l'accès au fichier WSDL en mettant en place un mécanisme d'authentification par exemple.

Conclusion

Comme nous avons pu le voir tout au long de cet article, les risques sont bien réels avec les applications Web 2.0. Il est important que les développeurs soient conscients des problèmes qui peuvent subvenir, afin qu'ils mettent en place les procédures nécessaires et ainsi éviter tout incident. Mais malheureusement à l'heure actuelle, la sécurité n'est pas toujours prise assez en amont dans le cadre des projets informatiques.

David MAGNY
Consultant IT
www.pragmaconsult.lu
david.magny@pragmaconsult.lu

N°100
IN-DIS-PEN-SA-BLE !!!
Un numéro historique.
Toutes les références du développement :
• Les 100 meilleurs logiciels
• Les 100 termes à comprendre
• Les 25 gourous de l'histoire du soft
• Quel avenir pour le développeur ?

2e partie du dossier
Web 2.0
La suite au prochain numéro !
• Quelle architecture choisir ?
• Comment référencer un site web 2.0 : les blocages et les astuces.
• L'Ajax d'Adobe : Spry.
• Le modèle économique du web 2.0

Comment choisir son framework Ajax ?

Ajax est une méthode et non une technologie. Il s'agit en résumé, d'exploiter l'objet XmlHTTPRequest du langage Javascript afin d'établir des connexions entre une page (déjà chargée) et le serveur. Le choix d'un framework, parmi ceux, déjà nombreux à votre disposition, dépendra donc de la manière dont vous entendez exploiter les résultats de ce type de requêtes. Selon la nature et la complexité des traitements, tous les frameworks ne seront pas adaptés.

En effet, on peut distinguer plusieurs types de frameworks pour faciliter la création de code Ajax. Mais il est cependant vain de vouloir tous les classer dans une case unique. Beaucoup de frameworks couvrent des domaines transversaux, car le web 2.0, ce n'est pas seulement l'utilisation d'Ajax. Ce n'est pas non plus seulement l'usage intensif de gadgets graphiques. Et c'est encore moins l'utilisation de MVC (Model View Controller - motif de conception facilitant la séparation de la logique métier de l'interface homme-machine). C'est le plus souvent tout cela à la fois, selon un dosage qui dépend de l'ergonomie et des fonctionnalités attendues. Aussi faut-il comprendre des trois catégories selon lesquelles nous classons les frameworks qu'elles décrivent la caractéristique principale de l'outil, et non l'intégralité de son champ d'action.

1 Frameworks Javascript

Il s'agit en fait le plus souvent d'une librairie, voire d'une simple classe, purement Javascript, et donc indépendante du langage côté serveur, dont la vocation première est de permettre au développeur de configurer, d'exécuter et de gérer le retour d'appels Ajax. Certains proposent en outre d'étendre le langage Javascript lui-même, et de rendre l'accès aux propriétés DOM des objets manipulés plus confortable.

Ce type de librairie peut d'ailleurs s'avérer très intéressant y compris dans des projets où Ajax en soi n'est pas indispensable, mais dans lesquels Javascript est prépondérant, car les extensions apportées au langage représentent un réel gain de temps à l'écriture, et une lisibilité incomparable.

Parmi les autres avantages de ce type de framework, l'universalité n'est pas le moindre. En effet, la plupart sont écrits de manière à être utilisables sur la majorité, sinon l'ensemble, des navigateurs les plus utilisés, et même certains plus exotiques. Universels, également parce qu'étant écrits uniquement en Javascript, ils permettent au développeur de les employer indifféremment avec n'importe quel langage côté serveur.

Quelques références

Prototype JS - www.prototypejs.org

L'incontournable compagnon du succès de Ruby on Rails (voir partie 3) est sans doute la plus complète de ce genre de librairies. Ses innombrables extensions du langage Javascript (nouvelles fonctions, nouveaux types de données, nouvelles méthodes sur des objets de base, etc.) et sa documentation (enfin !) très complète en font une véritable référence en la matière. Prototype offre à de très nombreux problèmes classiques en Javascript des solutions simples et élégantes. Sans oublier la simplification à peine croyable que Prototype apporte à la manipulation des objets et propriétés DOM !

Sack

http://www.twilightuniverse.com/projects/sack/

Bien plus compact et limité que le précédent, Sack est intéressant si vos besoins se limitent aux transactions Ajax.

Le périmètre très restreint de ses possibilités, en contrepartie, lui permet une légèreté sans commune mesure avec les concurrents, et donc des performances excellentes, sans alourdir le chargement initial des pages.

Cela lui a d'ailleurs valu d'être intégré au moteur de blogs Wordpress 2.0. Le développement de Sack est plus ou moins stoppé, dans la mesure où les fonctionnalités sont arrivées à maturité, et que l'auteur ne semble pas souhaiter changer de philosophie, et donc étendre la librairie.

jQuery - www.jquery.com

Force est de le reconnaître, le vieux maître Prototype trouve en jQuery une relève de très haut niveau, et qui probablement le surclasse sur bien des points, et principalement sur un terrain qui pourtant est resté longtemps la prédilection de Prototype : la manipulation DOM. jQuery est incontestablement le plus complet aujourd'hui dans ce domaine, proposant une syntaxe d'une puissance incroyable. Même si ceci est au prix du sacrifice, relatif, des performances, les possibilités offertes par ce framework sont fantastiques. En outre, son mécanisme de plug-in, dont il possède une impressionnante collection du fait d'une communauté d'utilisateurs très active, permet de contourner très facilement tout manque que vous pourrez éventuellement rencontrer.

2 Bibliothèques d'interfaces graphiques

Le Web 2.0 a rapidement été identifié aux avancées époustouflantes qu'ont connues les sites webs ces dernières années en matière d'ergonomie. Rappelons-nous l'apparition des premières zones de textes avec auto-complétion, notamment dans certaines versions du moteur de recherche Google ! Le résultat était bluffant, et beaucoup de développeurs web se sont dit : « ça y est, on tient enfin l'arme absolue pour annihiler les applications riches lourdes ». Hélas, après avoir jeté un coup d'œil au code nécessaire pour réduire ce simple comportement, nombre d'entre eux ont déchanté, se disant qu'il ne serait jamais possible de créer des applications entières basées sur ce type de mécanismes s'il fallait des centaines de lignes de code (Javascript, qui plus est) pour gérer le moindre contrôle riche. C'était sans compter sur la fertilité de la communauté du logiciel libre, qui en la matière

a su prendre une avance considérable, une fois n'est pas coutume, sur les éditeurs traditionnels. En effet, sont rapidement apparus différents frameworks totalement dédiés à la gestion d'éléments d'interfaces riches et autres effets jusqu'alors réservés au monde, souvent honni par une partie de la population de développeurs web, de Flash.

Quelques références

Scriptaculous - http://script.aculo.us
Bâti par-dessus Prototype, Scriptaculous est lui aussi à considérer en tant qu'ancêtre, et à respecter en tant que tel. Précurseur, Thomas Fuchs l'a été, en proposant ce qui fut probablement l'une des premières librairies Javascript à offrir une gestion du glisser/déplacer, des effets sur les textes et calques HTML, jusqu'au morphing ! Tout n'est pas parfait dans le monde merveilleux de Scriptaculous, et on regrette principalement l'absence cruelle d'une documentation digne de ce nom. Mais le développement se poursuit, aussi ne faut-il pas désespérer de voir le projet s'étoffer de ce point de vue prochainement.

Dojo - www.dojotoolkit.org
Autre framework de renom depuis déjà longtemps dans le monde du web 2.0, Dojo est effectivement séduisant par ses nombreuses possibilités. Ses capacités en termes de manipulation de média SVG sont particulièrement remarquables. Attention toutefois, le moteur de Dojo est plutôt lourd, et lent à démarrer, ce qui peut rendre son utilisation parfois agaçante. Cela dit, bien utilisé, Dojo permet de faire des interfaces vraiment convaincantes. Notons également que le développement reste assez actif.

Yahoo User Interface
http://developer.yahoo.com/yui/
Plus récent que les deux concurrents, Yahoo UI n'en a pas moins rapidement rattrapé son retard, Yahoo oblige. A l'instar de Google, Yahoo a commencé par développer ce framework pour son usage interne. Mais devant l'ampleur de la tâche, et l'intérêt manifeste de la communauté du libre pour cette technologie, Yahoo a très vite décidé de placer son projet sous une licence BSD (quasi libre de droit). La dynamique communautaire, alliée aux moyens du géant américain, a permis à Yahoo UI de s'enrichir d'une façon considérable en un temps record, pour finalement être aujourd'hui l'un des frameworks les mieux notés de sa catégorie. Mais là encore, la richesse fonctionnelle est quelque peu handicapée par la lourdeur du noyau, qui reste le problème majeur de Javascript quand on lui en demande tant…

3 Frameworks web 2.0 complets

Enfin, troisième catégorie majeure, les frameworks qui encadrent vos développements de A à ... Web 2.0 bien sûr. Inévitablement, il faut citer Ruby on Rails, dont on reparlera également ci-après. Bien que basé sur un langage à la diffusion très confidentielle (et qui le reste aujourd'hui encore, par ailleurs), Ruby on Rails à véritablement révolutionné l'approche du développement web. Outre le recours intensif à Ajax (Prototype JS, que nous avons évoqué précédemment, est né avec lui), Ruby on Rails a apporté au monde du développement web un élément clef, et qui lui faisait cruellement défaut : la rigueur.
La plupart des applications web sont basées sur des langages de scripts, tels PHP, Python, JSP et dans une moindre mesure ASP. Ces langages ont permis la démocratisation du développement, car ils sont faciles à apprendre et à mettre en œuvre. En revanche, de par leur conception même, ils se prêtent facilement à la réalisation d'un code parfaitement anarchique. Ruby on Rails a propulsé dans ce monde une notion aujourd'hui incontournable : le Design Pattern, à commencer par le MVC, pierre angulaire de ce type de frameworks. Depuis l'avènement de Ruby on Rails, le développement web peut revendiquer une qualité et un sérieux digne des applications lourdes classiques, ce qui lui était jusque là formellement contesté.
Les frameworks MVC pour le web sont généralement livrés avec une batterie de bibliothèques permettant de couvrir la plupart des fonctionnalités évoquées au sujet des deux catégories précédentes, mais ont la particularité d'être fortement liés à un langage donné.

Quelques références :

Ruby on Rails - www.rubyonrails.com
Inutile de s'étendre sur le sujet, nous en avons déjà parlé, Ruby on Rails est la référence en la matière. S'il n'est probablement pas le plus avancé, et s'il souffre finalement beaucoup d'être bâti sur Ruby, RoR reste le pionnier. Ne serait-ce que pour appréhender le concept, tester RoR est une bonne idée. Et qui sait, peut-être vous laisserez-vous séduire par ce langage déroutant, mais néanmoins très intéressant ?

CakePHP - www.cakephp.org
Impossible lorsque l'on évoque Ruby on Rails de ne pas citer son pendant, PHP, que les lecteurs assidus de Programmez ! connaissent déjà : CakePHP. Il s'agit, au moins à l'origine, d'un portage quasi à l'identique de RoR en PHP. Avec le temps, Cake a pris une certaine indépendance vis-à-vis de son modèle, mais l'esprit reste très proche. Si on peut lui reprocher quelques lacunes, notamment sur les performances et sur son modèle Active Record, qui n'est pas des plus agréables, CakePHP est tout de même très convivial, et permet de rapidement développer de très bonnes applications Web 2.0.

Zend Framework - www.framework.zend.com
Proposant une approche un peu différente, car moins directif que les deux autres, le Zend Framework permet cependant de faire, entre autres, la même chose. Naturellement basé sur PHP lui aussi, ce framework propose un grand nombre de modules avec très peu de dépendances entres eux, ce qui fait son originalité. Côté Ajax, s'il est possible d'utiliser les librairies purement Javascript bien sûr avec le ZF, l'intégration d'un module d'interface pour la librairie Dojo devrait prochainement faciliter encore l'écriture d'applications web 2.0 avec ce framework.

Google Web Toolkit
http://code.google.com/webtoolkit/
Pour ne pas déroger à ses habitudes, Google est encore là où ... on finissait par l'attendre en fait. Depuis des années que la société californienne rendait publique des API d'accès à ses services, il fallait bien se douter qu'elle finirait par proposer un framework complet. Le choix technologique a été Java. Les meilleurs exemples de ce que peut produire GWT sont certains services Google, comme notamment Google Maps ou GMail. L'ensemble est libre, soumis à une licence Apache 2.0.

Conclusion

Le choix d'un framework, quel que soit le domaine, peut être lourd de conséquences. Selon la taille des équipes à former ou l'ampleur des projets à mener, un mauvais choix initial peut entraîner des problèmes parfois catastrophiques pour une entreprise. Ce choix doit donc se faire en pleine connaissance de cause. Si bien sûr il est capital de sélectionner un outil apte à répondre à tous les besoins du projet, tels qu'ils sont définis dans son cahier des charges, il faut aussi s'assurer que le framework saura s'adapter aux évolutions futures du projet. La souplesse est un atout majeur pour un projet d'envergure. Il est également important de vérifier que la communauté d'utilisateurs est active lorsqu'il s'agit d'un produit libre, ou la qualité du support pour les produits commerciaux. Votre projet est trop dépendant de votre framework pour prendre le risque de rester bloqué faute de support adéquat.

Gauthier Delamarre

AVIS D'EXPERTS

Pourquoi faire un site communautaire avec Ruby ?

Les ambitions de notre projet ont imposé 4 contraintes en terme de framework de développement : Open Source, développement rapide par itérations, robustesse, et mise à disposition d'outils pour l'élaboration d'interfaces riches.
Grâce à leur architecture Modèle / Vue / Contrôleur, leur infrastructure de tests performante et leur ORM efficace, turboGears, Django et Rails sont arrivés dans le peloton de tête. Notre choix final s'est porté sur Rails, du fait de la concision du langage Ruby et des nombreuses références de sites en production.

La migration

Une migration vers Rails est difficile en particulier à cause des conventions de nommage. Adapter systématiquement ces conventions afin d'assurer la compatibilité avec un vieux modèle de données est une opération coûteuse et périlleuse. Ré-écrire le modèle selon les bonnes pratiques du framework est plus judicieux.

Les avantages / Inconvénients

La principale force de Rails tient au fait qu'il est le fruit de l'expérience capitalisée sur un vrai projet. Il répond donc à des problématiques bien réelles.
Vient ensuite son langage de programmation, Ruby, qui offre concision et souplesse. Les applications comportent ainsi moins de code et sont de ce fait plus lisibles et source de moins de bogues. Autre avantage du framework : sa communauté. Fédérée et très active, celle-ci participe à l'extension de Rails en développant de nombreux plug-in, dont certains sont de qualité professionnelle. Ce sont des ressources très précieuses venant démultiplier le potentiel du framework.
En revanche, la documentation de l'API de Rails est assez inégale et parfois trop elliptique. Par ailleurs, Rails est à double tranchant : sa facilité de prise en main peut se révéler trompeuse. En effet, elle ne doit pas masquer l'essentiel, à savoir porter ses efforts sur une implémentation de qualité et correctement testée.

Quelques conseils

Le choix d'une technologie "Web 2" doit évidemment prendre en compte l'interface riche, la robustesse en cas de montée en charge et la démarche agile pour intégrer dans de courts délais les retours des utilisateurs. En plus de cela, les problématiques que sont l'internationalisation de l'application, la sécurité et surtout les tests ne doivent pas être occultées.

Cédric Delthel
Denis Brulé

En création d'entreprise, avec le soutien de l'incubateur Agoranov

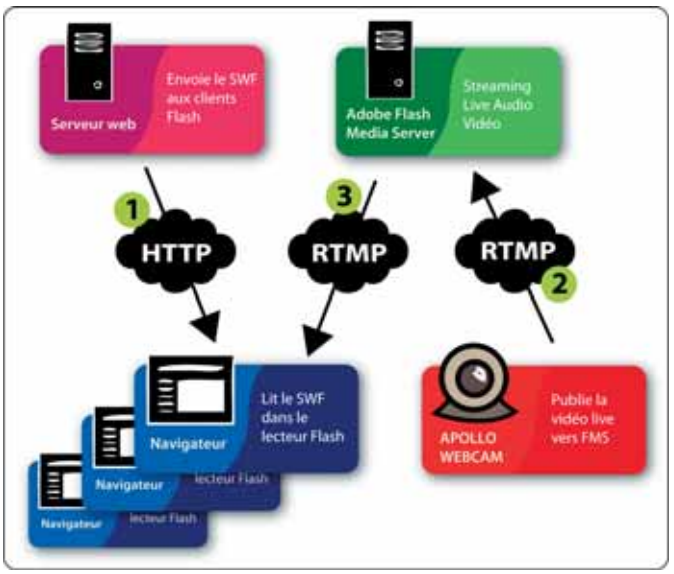
Une webcam " live " pour mon blog 2.0

Le web 2.0 nous fait redécouvrir que nous ne sommes plus seuls sur le net. Les fonctionnalités des sites 2.0 favorisent les interactions communautaires, et les internautes sont devenus de réels producteurs de contenus. J'en veux pour preuve l'explosion du nombre de blogs et la prolifération de vidéos personnelles sur des sites comme youTube ou dailyMotion.

La webcam devient progressivement le périphérique phare de cette révolution. Sur YouTube, vous pouvez laisser un enregistrement en direct avec votre webcam et la vidéo est publiée instantanément en ligne. Des projets comme eyeJot.com proposent de remplacer vos emails classiques par des messages vidéo. J'ai donc eu l'envie de partager avec vous un développement qui met en œuvre la vidéo, la webcam, un blog et une application riche. Les " webcams live " se développent à vive allure sur les blogs pour partager sa vie au quotidien, l'activité d'un bureau ou un panorama.

Le développement consiste en deux modules :

- une application AIR réservée à l'auteur du blog qui capture le flux vidéo d'une webcam. J'ai développé une application AIR persistante sur le bureau pour éviter d'avoir à conserver un navigateur ouvert en permanence sur mon poste.
- une application Flash (développée avec Flex) que l'on pourra facilement insérer dans un blog pour retransmettre l'image et le son en live de la webcam de l'auteur du blog.



Architecture de la solution

Pour capturer le flux d'une vidéo et la retransmettre en temps réel aux visiteurs du blog, j'utilise le serveur Adobe Flash Media Server 2. Une édition gratuite est disponible sur le site d'Adobe (Developer Edition). Flash Media Server utilise le protocole RTMP pour diffuser en streaming ou en direct de la vidéo ou du son. La première étape du développement consiste donc à installer FMS et à créer une application dédiée à notre webcam live. Pour ce faire, il suffit de se rendre dans le répertoire " applications " de FMS et de créer un répertoire " flex_myblog ". Créez ensuite un fichier main.asc dans ce répertoire qui contient cette ligne :

```
load( "components.asc" );
```

Cela chargera tous les composants de base de Flash Media Server. La configuration du serveur est terminée.

Le protocole RTMP

Le RTMP (Real Time Messaging Protocol) est un protocole propriétaire créé à l'origine par Macromedia. Il est utilisé par Flash Media Server pour streamer du son et de la vidéo vers les lecteurs Flash. Le RTMP maintient une connexion persistante entre les clients et le serveur pour permettre des échanges en temps réel. Un projet openSource Red5 fait du reverse engineering pour constituer une architecture similaire. D'autres serveurs comme Flex Data Services et WebORB utilisent le protocole RTMP. RTMPT (RTMP Tunneled) est une variante intéressante qui encapsule les données RTMP dans la couche HTTP et utilise donc le port classique 80. Le protocole peut assurer des échanges sécurisés (RTTPS).

Développement de l'application AIR qui diffuse la vidéo live

Nous avons vu comment réaliser une application AIR avec Flex Builder (Programmez ! 97). Dans Flex Builder, j'ai créé un nouveau projet AIR et ajouté deux composants de base du Flex SDK :

```
<?xml version="1.0" encoding="utf-8"?>
<mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" width="300" height="280" creationComplete="initApp()">
  <mx:Panel title="Ma webcam vers mon blog" borderColor="#000000">
    <mx:Canvas width="100%" height="215">
      <mx:VideoDisplay id="videoLocal" />
      <mx:Button id="bStartWebcam" label="Démarrer ma webcam" click="startCamera()" y="179" horizontalCenter="0" />
    </mx:Canvas>
  </mx:Panel>
</mx:WindowedApplication>
```

Une fois l'application AIR chargée, la fonction initApp() est appelée.

```
private function initApp():void
    {
      nc = new NetConnection();
      nc.objectEncoding = flash.net.ObjectEncoding.AMF0;
      nc.connect("rtmp://localhost/flex_myblog");
    }
```

L'objet NetConnection crée un lien persistant avec Flash Media Server. Les données circulent sur le réseau en binaire AMF. J'indique ici le chemin vers le serveur FMS (en local dans mon exemple). Le composant VideoDisplay sera utilisé pour afficher le flux vidéo issu de la webcam. Un clic sur le bouton déclenche la fonction ActionScript startCamera() qu'il nous reste à coder.

```
private var userId:String = "myblog";

private function startCamera():void
        {
 //Activation de la webcam
            camera = Camera.getCamera();
            microphone = Microphone.getMicrophone();
          videoLocal.attachCamera(camera);

          //Publication sur FMS
            nsPublish = new NetStream(nc);
            nsPublish.attachCamera(camera);
            nsPublish.publish(userId);
            nsPublish.attachAudio(microphone);
        }
```

La fonction cherche d'abord à activer la webcam (getCamera) et le micro (getMicrophone) de l'ordinateur et à l'associer au lecteur Flash qui sait gérer nativement ces périphériques. Le flux vidéo est alors affiché dans le composant VideoDisplay de l'interface. Dans un second temps, je publie les flux son et vidéo au Flash Media Server grâce à l'objet NetStream sur le canal de diffusion " myblog ". L'application de l'auteur du blog est prête. Le flux de la webcam sera envoyé en live à FMS dès qu'il cliquera sur le bouton de son application AIR.

Développement du client Flash qui reçoit le flux vidéo en live

Pour des raisons de productivité, je me sers ici du SDK Flex pour développer l'application Flash qui sera présente sur mon blog.
Le code tient sur à peine 20 lignes !

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="connectStream()" layout="absolute" background
GradientColors="[#11c0df, #165369]">
  <mx:Script>
    <![CDATA[

      private function connectStream():void
      {

        maCamera.source = "rtmp://localhost/flex_myblog/myblog";
      }

    ]]>
  </mx:Script>
  <mx:Panel x="10" y="10" width="250" height="223" layout=
"absolute" title="Le video-chat de Michael">
    <mx:VideoDisplay id="maCamera" width="210" height="158"
autoBandWidthDetection="false" live="true" autoPlay="true" x="10" y="10"/>
  </mx:Panel>

</mx:Application>
```

Je commence par créer un Panel qui contient un composant VideoDisplay. C'est ce dernier qui affichera le flux audio/vidéo en provenance du Flash Media Server. La propriété " live " précise que le composant va afficher du contenu diffusé en direct en provenance de FMS. Je n'utilise pas ici la propriété autoBandWidthDetection. Sachez qu'elle permet d'adapter la qualité de la vidéo en fonction du débit disponible sur la bande passante au moment de la diffusion. Sa mise en place demande du paramétrage sur FMS, je le passe donc à 'false' pour simplifier la mise en place de cet exemple. Dès le chargement de l'application, la fonction ActionScript connectStream() est appelée. Elle indique au composant vidéo l'URL vers le canal Flash Media Server qui diffuse du contenu en live. Comme vous le voyez, la partie cliente est très simple ! Une fois ce code compilé, vous obtenez un fichier Flash.swf très simple à intégrer dans votre blog. Vous pouvez désormais vous amuser à étendre l'application pour réaliser un chat vidéo multi-utilisateurs. Pour vous aider, je vous conseille le blog de Jean Philippe de MediaBox qui est une mine d'informations : http://jeanphiblog.media-box.net/dotclear/index.php
Le site http://www.flashcomguru.com/ (en anglais) est le portail référence pour trouver des tutoriaux et des articles sur ce sujet.

Michaël Chaize - *Consultant Avant Vente chez Adobe*

Le kit pour développer une application "webcam live"

- Flex SDK
Kit de développement gratuit et openSource composé de deux langages (MXML, langage de description d'interface et ActionScript 3.0). Disponible en libre téléchargement sur www.adobe.fr
- Flex Builder
Environnement de développement Flex basé sur Eclipse. Disponible en version d'évaluation 30 jours sur www.adobe.fr
- AIR SDK
Environnement d'exécution multi-plate-forme disponible gratuitement en bêta en attendant la version finale sur labs.adobe.com
- Flash Media Server
Serveur de streaming Vidéo et Audio pour les clients Flash. Version gratuite pour les développeurs (Developer Edition) disponible sur www.adobe.fr
- Hébergement Flash Media Server : Pour mettre en production cette application, l'offre Streamedia (http://www.streamedia.eu/) semble prometteuse. A l'heure où j'écris cet article, seul un teaser est visible.

Silverlight 1.1 : créez vos applications riches et multimédia sur Internet

Silverlight est la nouvelle technologie de présentation Web de Microsoft, portant le nom de code : WPF/E (Windows Presentation Foundation / Everywhere). Silverlight se base sur XAML (eXtensible Application Markup Language) pour créer des interfaces graphiques riches améliorant l'expérience utilisateur. Pour cela, il reprend les fondements de WPF en utilisant aussi bien le XAML, que ses couches de gestion multimédia, ainsi que celles d'affichage et d'animation 2D.

Silverlight a pour but de sortir de ces frontières en proposant une solution multi-plate-forme et multi-navigateur. Il se présente sous la forme d'un plug-in pour le navigateur, est disponible pour Windows et MacOS X et pour les navigateurs Internet Explorer, Firefox et Safari. Une version mobile apparaîtra dans les prochains mois. Le monde GNU Linux n'est pas oublié. Le projet Mono a annoncé la mise en chantier d'une version Linux, Moonlight (http://www.mono-project.com/Moonlight).

Silverlight pour une nouvelle expérience utilisateur sur Internet

Les sites Internet utilisant la technologie Silverlight vont pouvoir offrir de nouvelles fonctionnalités pour leurs utilisateurs. Tout d'abord, grâce au moteur de rendu vectoriel, il sera possible d'avoir des interfaces utilisateur qui s'adapteront automatiquement à la dimension de l'écran du visiteur, mais aussi de dynamiser l'utilisation des sites Internet par le biais des animations. Le multimédia est de plus en plus présent autour de nous et le Web en est la principale source, c'est pour cela que Silverlight est capable de gérer différents formats comme le MP3, WMA et le format vidéo VC-1. Ce codec de Microsoft est basé sur WMV9 et standardisé par la SMPTE (Society of Motion Picture and Television Engineers), il permet de supporter les médias de haute définition. A l'attention des fournisseurs de contenu, Silverlight 1.1 sera capable de gérer les droits numériques (DRM) en s'appuyant sur la technologie d'accès aux contenus appelée Microsoft PlayReady. Pour aller dans le sens du média, Microsoft a ajouté à sa plate-forme Live, un nouveau service spécifique au streaming de vidéo haute définition via Silverlight Streaming (http://silverlight.live.com). Ce nouveau service sera facilement accessible par les applications utilisant la technologie Silverlight. Microsoft a créé un site internet exclusivement dédié à Silverlight, sur lequel on pourra télécharger les dernières versions des plug-ins, ainsi que des exemples et des informations sur la nouvelle technologie de Microsoft (http://www.silverlight.net).

Silverlight 1.1 : La révolution du développement Web côté client

Cette nouvelle version rend maintenant possible l'exécution côté client de code .net en lieu et place de l'actuel JavaScript de la version 1.0. Surtout, avec la version 1.1, l'application pourra s'exécuter sur son bureau grâce à l'intégration dans Silverlight de .net. Cela devient possible via un plug-in Silverlight pesant moins de 5Mo et existant pour plusieurs types de plates-formes et de navigateurs. Ainsi, il sera possible

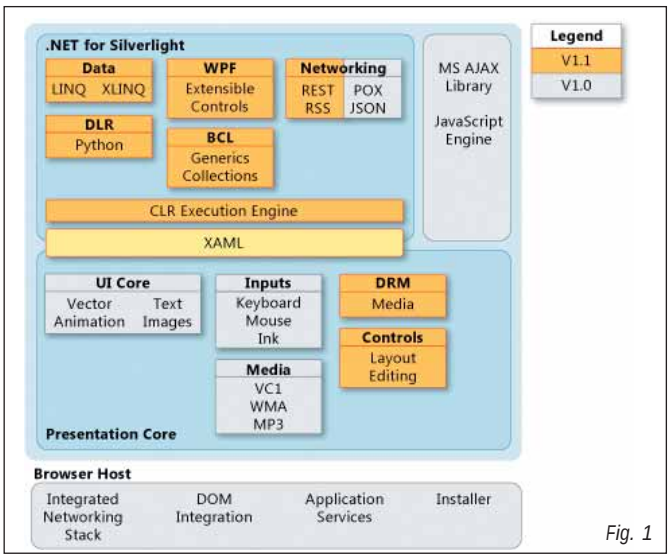


Fig. 1

de naviguer sur des sites Internet utilisant Silverlight pour la partie présentation, mais surtout de pouvoir gérer toutes les interactions coté client via le runtime .net sous Windows et MacOS X, Moonlight promet la même chose sous Linux. Avec cette nouvelle version, différentes librairies du Framework seront en partie ajoutées et même complétées pour répondre plus spécifiquement aux besoins d'une application Web, par exemple la prise en charge de la communication avec des Web Services. (Figure 1) On peut ainsi voir sur ce schéma que différents éléments ont été rajoutés avec la version 1.1 et le runtime .net, en effet, on y retrouve les classes de base, comme les Collections ou les Generics. Cependant, on constate que la partie données pourra être gérée via LINQ et XLINQ, ce qui implique que le CLR .net utilisé dans Silverlight est un sous-ensemble de celui de la version 3.5 du Framework .net. C'est d'ailleurs pour cette raison que le runtime ne pèse que quelques Méga-Octets. Mais l'on remarque aussi l'arrivée d'un élément nouveau au sein du Framework .net, il est nommé DLR pour " Dynamic Language Runtime ", celui-ci rend possible l'exécution de langage dynamique de type Python, Ruby ou même du futur VB X.

Quels sont les outils disponibles pour développer avec Silverlight 1.1

Cela nécessite l'utilisation d'outils de nouvelle génération. En effet, la nouvelle version de Silverlight s'appuyant en partie sur des éléments du Framework .net 3.5, elle a besoin de la version de Visual Studio qui lui

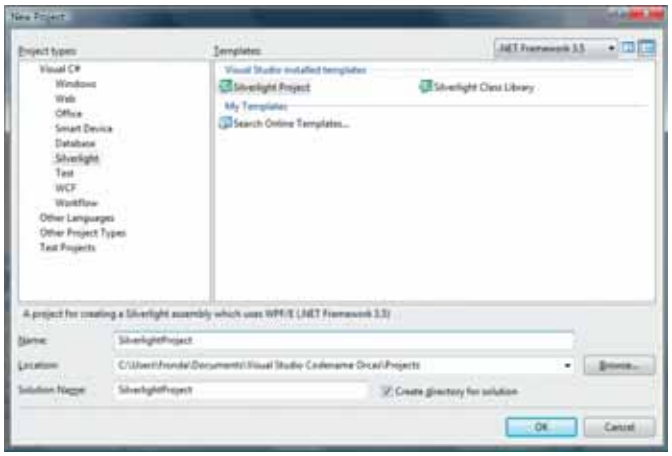


Fig. 2

est dédiée, c'est-à-dire " Orcas ", de plus il faudra lui rajouter une extension afin que celui-ci soit capable de créer des projets Silverlight. Pour pouvoir plus aisément éditer les fichiers XAML des interfaces graphiques de vos applications, il faudra utiliser la nouvelle version de l'outil destiné aux designers, Expression Blend 2, car seule cette version est capable de générer des fichiers XAML compatibles avec Silverlight et le nouveau Visual Studio. Tous ces outils sont disponibles en téléchargement sur le site de Microsoft (voir aussi : http://silverlight.net/GetStarted).

Créer une application Silverlight 1.1

Nous allons voir comment réaliser une application simple, afin de mieux se rendre compte des bénéfices que peuvent apporter Silverlight 1.1. Pour cela, nous prendrons l'exemple d'une application affichant un UserControl permettant d'afficher une liste d'images, l'une après l'autre. La première étape consiste à créer un projet de type Silverlight dans Visual Studio. Pour cela, on doit en premier lieu sélectionner la version du Framework sur laquelle notre projet va se baser. Dans notre cas on choisira la version 3.5, car c'est la seule permettant la réalisation d'application Silverlight. (Figure 2). Une fois le projet créé, on constate que notre projet contient déjà quelques fichiers. Tout d'abord, nous avons un fichier " TestPage.html " qui sert de point d'entrée à notre application et son fichier "TestPage.js" pour l'exécution de scripts spécifiques à cette page. On retrouve aussi un autre fichier JavaScript nommé " Silverlight.js " qui contient tout le code JavaScript nécessaire à la détection du plug-in Silverlight sur la machine cliente. Et enfin, un fichier " Page.xaml " contenant le code XAML de notre application, ainsi que le fichier " Page.xaml.cs " qui représente le code C# attaché au fichier XAML. C'est dans ce fichier que l'on définira le code à exécuter, lors d'événements envoyés suite à des interactions avec l'utilisateur. Enfin, pour réaliser notre application, nous allons rajouter un UserControl à notre projet, afin que cet élément soit réutilisable plus facilement. On obtient donc un nouveau fichier XAML qu'on aura nommé " SimpleSlideShow.xaml " dans lequel on définira l'interface du UserControl et un fichier " SimpleSlideShow.xaml.cs " qui contiendra le code C# nécessaire à la réalisation de notre diapositive. (Figure 3)

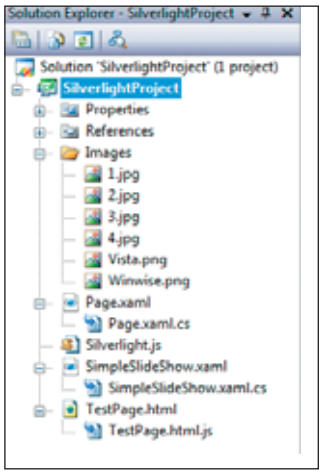


Fig. 3

```
[./SimpleSlideShow.xaml]
<Canvas     xmlns="http://schemas.microsoft.com/client/2007"
            xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
            Width="300" Height="240" x:Name="ParentCanvas">

  <Canvas.Resources>
    <!- Storyboard permettant le fondu entre l'ancienne et la nouvelle image ->
    <Storyboard x:Name="ImagesBlurAnimation">
      <DoubleAnimationUsingKeyFrames Storyboard.TargetName="ImageNew"
                  Storyboard.TargetProperty="(UIElement.Opacity)">
        <SplineDoubleKeyFrame KeyTime="00:00:00" Value="0"/>
        <SplineDoubleKeyFrame KeyTime="00:00:02" Value="1"/>
      </DoubleAnimationUsingKeyFrames>
      <DoubleAnimationUsingKeyFrames Storyboard.TargetName="ImageOld"
                  Storyboard.TargetProperty="(UIElement.Opacity)">
        <SplineDoubleKeyFrame KeyTime="00:00:00" Value="1"/>
        <SplineDoubleKeyFrame KeyTime="00:00:02" Value="0"/>
      </DoubleAnimationUsingKeyFrames>
    </Storyboard>
  </Canvas.Resources>

  <!- Fond du UserControl avec un dégradé ->
  <Rectangle Width="300" Height="240" StrokeThickness="1" RadiusX="0"
RadiusY="0" Stroke="#FF000000">
    <Rectangle.Fill>
      <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
        <GradientStop Color="#FFA29DCF" Offset="0"/>
          <GradientStop Color="#FF49465D" Offset="0.15"/>
          <GradientStop Color="#FF3D3C63" Offset="0.95"/>
          <GradientStop Color="#FF121320" Offset="1"/>
        </LinearGradientBrush>
      </Rectangle.Fill>
  </Rectangle>

  <!- Titre de notre SlideShow ->
  <TextBlock Width="160" Height="20" Canvas.Left="70" Canvas.Top="5"
          FontWeight="Bold" Foreground="#FFEEEEEE"  Text="Silverlight SlideShow" />

  <!- Images utilisés pour afficher les images avec l'effet de fondu ->
  <Image x:Name="ImageNew" Width="280" Height="200" Canvas.Left
="10" Canvas.Top="30" Stretch="Uniform"/>
  <Image x:Name="ImageOld" Width="280" Height="200" Canvas.Left
="10" Canvas.Top="30" Stretch="Uniform"/>

</Canvas>
```

Comme on peut le voir, le fichier XAML contient les éléments graphiques de notre UserControl. Il a été nécessaire de nommer certains de ces éléments, soit car ils sont référencés dans une autre partie du fichier XAML, par exemple dans le Storyboard, ou alors parce qu'ils doivent être utilisés dans la partie code. En les nommant ainsi, il sera plus aisé de récupérer l'instance de chacun dans le code afin de les manipuler.

```
[./SimpleSlideShow.xaml.cs]
using System;
using System.IO;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media;
using System.Windows.Browser;

using System.Collections.Generic;
using System.Windows.Media.Animation;

namespace SilverlightProject
{
    public class SimpleSlideShow : Control
    {
        private HtmlTimer timer;
        private int currentImageIndex;
        private List<string> imageList;

        private FrameworkElement rootElement;
        private Image imageElementOld;
        private Image imageElementNew;
        private Storyboard blurAnimation;

        public List<string> ImageList
        {
            get { return imageList; }
        }

        public SimpleSlideShow()
        {
            // Chargement du fichier xaml et recupération de l'élément racine
            Stream s = this.GetType().Assembly.GetManifestResourceStream("SilverlightProject.SimpleSlideShow.xaml");
            rootElement = this.InitializeFromXaml(new StreamReader(s).ReadToEnd());

            // Via m_rootElement, on recupère les instances des éléments que l'on va manipuler
            imageElementNew = rootElement.FindName("ImageNew") as Image;
            imageElementOld = rootElement.FindName("ImageOld") as Image;
            blurAnimation = rootElement.FindName("ImagesBlurAnimation") as Storyboard;

            // Initialisation des variables internes
            currentImageIndex = 0;
            imageList = new List<string>();

            // Initialisation du Timer
            timer = new HtmlTimer();
            timer.Interval = 5000;
            timer.Tick += new EventHandler(timer_Tick);
        }

        /// <summary>
        /// Lance le slideshow avec les images présentes dans la liste
        /// </summary>
        public void StartSlideShow()
        {
            currentImageIndex = 0;
            if (currentImageIndex < imageList.Count)
                imageElementOld.Source = new Uri(imageList[currentImageIndex], UriKind.RelativeOrAbsolute);
            timer.Start();
        }

        /// <summary>
        /// Affiche l'image suivante avec la gestion de l'effet de fondu
        /// </summary>
        public void ShowNextPicture()
        {
            if (currentImageIndex < imageList.Count)
            {
                // Initialise la source avec le chemin de l'image précédente
                imageElementOld.Source = new Uri(imageList[currentImageIndex], UriKind.RelativeOrAbsolute);

                // Mise à jour de l'index de l'image à afficher
                currentImageIndex += 1;
                if (currentImageIndex >= imageList.Count)
                    currentImageIndex = 0;

                // Initialise la source avec le chemin de l'image suivante
                imageElementNew.Source = new Uri(imageList[currentImageIndex], UriKind.RelativeOrAbsolute);

                // Lance l'animation qui fera la transition entre l'ancienne et la nouvelle image
                blurAnimation.Begin();
            }
        }

        private void timer_Tick(object sender, EventArgs e)
        {
            ShowNextPicture();
        }
    }
}
```

Ainsi grâce au code C# de notre UserControl, il est très simple de développer toute la logique de notre composant. Reste maintenant à voir comment utiliser notre UserControl dans notre application. Par défaut, lors de la création du projet, la page " TestPage.html " héberge le fichier " Page.xaml " dans lequel nous allons utiliser notre nouveau contrôle.

```
[./Page.xaml]
<Canvas xmlns="http://schemas.microsoft.com/client/2007"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:Local="clr-namespace:SilverlightProject;assembly=ClientBin/SilverlightProject.dll"
      x:Class="SilverlightProject.Page;assembly=ClientBin/SilverlightProject.dll"
      x:Name="parentCanvas"
```

```
    Loaded="Page_Loaded"
    Width="600"
    Height="400"
    >

  <Local:SimpleSlideShow x:Name="slideShow" Width="100" Height="100"
                         Canvas.Left="50" Canvas.Top="25"/>

</Canvas>
```

Afin de pouvoir utiliser notre contrôle, il faut tout d'abord rajouter le namespace de la librairie générée contenant notre UserControl, de cette façon, nous pouvons insérer notre objet dans l'interface. Il nous faut maintenant configurer notre contrôle, en lui donnant la liste des images à afficher. Nous allons faire cela lors de l'événement du chargement de notre fichier " Page.xaml ". Ainsi comme on peut le constater, il suffit d'ajouter le code dans la fonction " Page_Loaded ". C'est bon, notre UserControl affiche maintenant nos images les unes après les autres avec un effet de fondu entre chaque images (Fig. 4).

```
[./Page.xaml.cs]
using System;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;

namespace SilverlightProject
{
    public partial class Page : Canvas
    {
        public void Page_Loaded(object o, EventArgs e)
        {
            // Required to initialize variables
            InitializeComponent();

            this.slideShow.ImageList.Add(@"Images/1.jpg");
            this.slideShow.ImageList.Add(@"Images/2.jpg");
            this.slideShow.ImageList.Add(@"Images/3.jpg");
            this.slideShow.ImageList.Add(@"Images/4.jpg");

            this.slideShow.StartSlideShow();
        }
    }
}
```

Déployer une application Silverlight

Nous l'avons vu, les applications Silverlight sont des applications Internet. Comme toute application Internet, il faut disposer d'un serveur Web qui hébergera les fichiers HTML/Javascript que vous aurez créés. Si vous développez une application Silverlight 1.1, vous remarquerez qu'un répertoire ClientBin a été créé lors de la compilation de votre projet. Ce répertoire contient l'assembly qui abrite le code .NET que vous avez écrit pour votre application et qui sera " téléchargée " par votre navigateur afin d'être exécuter sur le poste client : il vous faut donc déployer également ce répertoire et son contenu sur le serveur. En ce

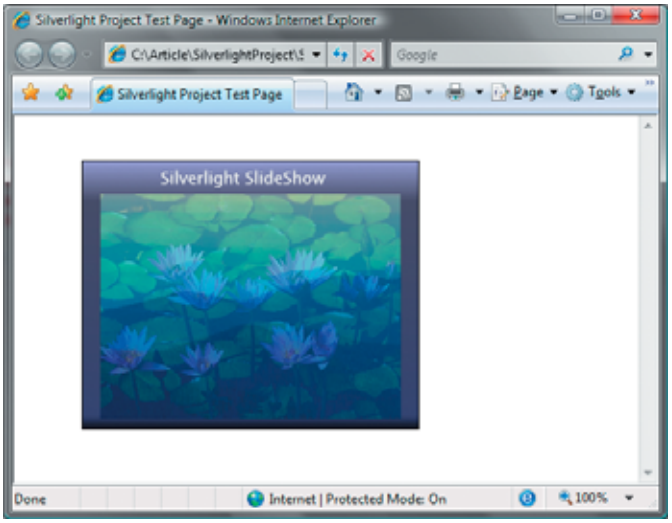


Fig. 4

qui concerne le déploiement à proprement parlé, il s'agit d'une simple copie des fichiers sur le serveur Web : étant donné que Silverlight est un plug-in qui s'exécute côté client, il n'y a rien besoin de faire sur le serveur pour que cela fonctionne. Si vous préférez, vous pouvez utiliser la plate-forme mise à disposition par Microsoft : http://silverlight.live.com. Bien que celle-ci possède des limitations (4 Go maximum de stockage, taille des vidéos limitée, Silverlight 1.0 uniquement, etc.), elle s'avère très complète et performante pour faire des tests. En ce qui concerne la sécurité, il faut savoir que les applications Silverlight s'exécutent dans un " bac à sable " (Sandbox) qui permet de restreindre les droits utilisés et d'éviter les problèmes de sécurité.

Conclusion

Comme on vient de le voir avec cet exemple, la mise en place d'une application riche pour Internet devient très simple grâce à cette nouvelle version de Silverlight. Le point fort de cette future version est de permettre l'utilisation des différents langages .net côté client, cela permet tout d'abord d'utiliser des langages et un environnement de programmation connus par un grand nombre de développeurs, mais cela laisse aussi enfin une réelle alternative à l'utilisation du JavaScript pour ce genre d'applications. La sortie officielle de Silverlight 1.1 n'est pas encore annoncée, pour le moment nous savons seulement que la version sans le runtime .net, c'est-à-dire Silverlight 1.0, sortira durant cet été. Il va donc encore falloir patienter un peu, mais cette version Alpha de Silverlight 1.1 donne et laisse présager beaucoup de nouveautés dans nos applications Web de demain.

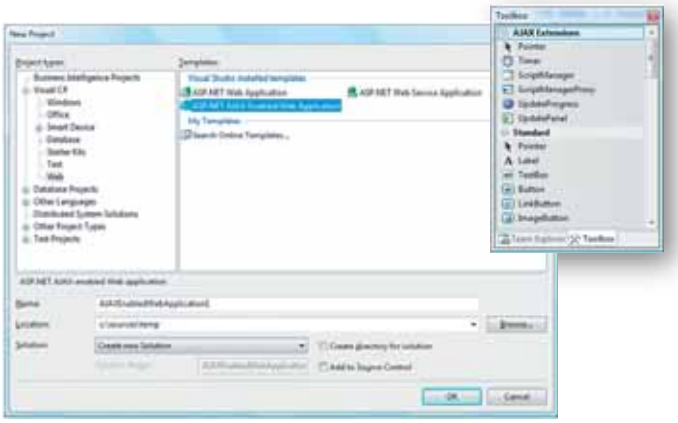
Davy Frontigny
Consultant en technologies .NET
Email : davy.frontigny@winwise.fr
Site : http://www.winwise.fr

Daniel Tizon
Microsoft Regional Director
Responsable du Pôle Génie Logiciel à Winwise
Email : daniel.tizon@winwise.fr
Blog : http://blogs.codes-sources.com/daniel/

Créer et déployer son site ASP.Net Ajax

Dans la galaxie AJAX, il existe de nombreux frameworks. Certains partent d'initiatives personnelles, d'autres sont portés par des entreprises. La plupart sont open source, quelquefois non maintenus, voire même abandonnés.

Microsoft s'est lancé très tôt dans la course à AJAX. D'aucuns diront qu'il en est même le créateur avec le support de XmlHttpRequest dans Internet Explorer 4.0 en 1998. Quoi qu'il en soit, la firme de Redmond a annoncé très tôt son positionnement sur le marché avec Atlas en 2005. Le projet Atlas est devenu ASP.NET Ajax v1.0, un produit officiel Microsoft, bénéficiant à ce titre du support complet de la part de l'éditeur pour les 10 prochaines années. Ce Framework est abouti, on y reconnaît la qualité de finition des produits Microsoft : une installation simplissime, une intégration aux outils, de nombreux exemples, une documentation touffue et une forte communauté, consolidée depuis les versions bêta.



Démarrage

ASP.NET Ajax est juste une surcouche à ASP.NET 2.0. On parle des extensions AJAX pour ASP.NET. Il n'est pas nécessaire de changer d'outil ou de revoir radicalement la façon de développer un site web. L'architecture ASP.NET ne change pas.

L'installation se fait tout simplement via un installeur classique. Cet installeur configure Visual Studio pour supporter les nouvelles fonctionnalités Ajax. A noter que les extensions Ajax fonctionnent avec toutes les versions de Visual Studio, même les gratuites.

ASP.NET Ajax est composé de 3 modules :

- Des extensions serveurs : elles vont prendre en charge les fonctionnalités AJAX dans les pages ASPX côté serveur
- Des contrôles qui encapsulent des comportements ajaxifiés pour dynamiser une page
- Microsoft AJAX Library : Une bibliothèque de scripts JavaScript pour prendre en charge AJAX côté client (manipulation de XMLHttpRequest, JSON, DOM, etc.)

La bibliothèque de scripts client est disponible de façon indépendante, ce qui permet de bénéficier des efforts de développement des équipes de Microsoft sans avoir l'obligation d'utiliser la technologie ASP.NET. De nombreux articles sur Internet expliquent comment l'utiliser dans un site, en PHP par exemple.

Le code source complet d'ASP.NET Ajax est disponible. Cela s'avère utile pour regarder comment Microsoft a implémenté son Framework mais aussi pour avoir accès à un débogage avancé.

Les extensions Ajax sont elles-mêmes extensibles. On parle de développer des extenders. Cet article ne couvre pas ce vaste sujet mais il faut savoir qu'ASP.NET Ajax propose une API simple et élaborée pour étendre et exploiter au mieux la technologie AJAX, et bien sûr, tout cela du côté serveur avec par exemple des services ou une sérialisation JSON personnalisée et/ou côté client avec des animations DHTML ou des effets visuels en CSS !

Associée aux extensions Ajax d'ASP.NET, on évoque souvent la Control Toolkit. Développée par des Microsoftee, c'est un ensemble de contrôles qui tirent parti des fonctionnalités d'ASP.NET Ajax. On trouve par exemple un calendrier, une fenêtre pop-up modale, des boîtes de saisie intelligentes, etc. Ils sont eux aussi proposés avec leur code source. Attention, contrairement aux extensions elles-mêmes, ces contrôles n'entrent pas dans les cycles de support de Microsoft.

Création du site

Les extensions Ajax mettent à disposition de nouveaux contrôles dans la boîte à outils et un nouveau type de projet Web. C'est ce type de projet, ASP.NET Ajax-enabled Web Application, qu'il faut choisir pour créer un site web AJAX. Une fois créé, il n'y a visuellement aucune différence avec un site ASP.NET classique, ce qui n'est pas tout à fait faux. En effet, seul le fichier de configuration web.config a été modifié, ce qui est nécessaire pour indiquer au moteur ASP.NET les extensions à utiliser lors du fonctionnement du site.

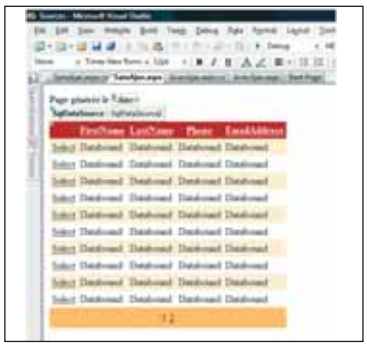
Il n'existe pas de moyen de transformer de façon automatisée un site web existant en un site web AJAX. Il faut modifier manuellement le fichier de configuration du site. Cette opération est décrite plus loin dans l'article.

Première page

Nous allons " ajaxifier " notre première page. Je vous propose de partir sur un exemple simple et qui parle à tout le monde. La page présente une GridView affichant les contacts de la base de démonstration AdventureWorks, livrée avec SQL Server Express. On active les fonctionnalités classiques de pagination, d'ordonnancement et de sélection.

A l'exécution, à chaque clic, l'ensemble de la page est rechargé par le mécanisme de Postback d'ASP.NET. En termes d'ergonomie, cela peut apporter un inconfort à l'utilisateur. En termes de performances, on renvoie l'ensemble du code HTML de la page au client, ce qui est consommateur en bande passante.

Grâce à AJAX, nous allons faire en sorte que seule la GridView soit rechargée dans le navigateur client. Pour cela, nous allons utiliser le contrôle central des extensions Ajax à savoir l'UpdatePanel. Ce contrôle conteneur permet d'envoyer au client uniquement son contenu. Le fonctionnement est complètement transparent, il suffit juste de mettre le contenu à ajaxifier dans l'UpdatePanel.



```
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
   <ContentTemplate>
[...]
<asp:GridView ID="GridView1" runat="server" AllowPaging="True" Allow
Sorting="True">
      [...]
      </asp:GridView>
   </ContentTemplate>
</asp:UpdatePanel>
```

Cette simple manipulation suffit pour ne plus avoir de rechargement complet de la page lors d'un événement sur la GridView. La date affichée atteste du rechargement d'une partie de la page seulement.
De nombreuses options sont disponibles pour configurer le comportement d'un UpdatePanel. Par exemple, si on a plusieurs UpdatePanel dans la page, on a moyen de définir de façon déclarative ou impérative lesquels doivent être envoyés au client pour mise à jour de l'interface.
Voila pour un exemple simple des extensions AJAX dans un site ASP.NET classique. Cela n'a nécessité aucun code côté serveur ni côté client, juste l'insertion d'un contrôle conteneur. Cependant, l'UpdatePanel n'est pas suffisant pour faire fonctionner ce scénario. La page doit toujours contenir un contrôle non visuel : le ScriptManager.

```
<asp:ScriptManager ID="ScriptManager1" runat="server" />
```

Ce contrôle sert de chef d'orchestre des fonctionnalités AJAX de la page entre le client et le serveur. Si ce contrôle est manquant et qu'un UpdatePanel se trouve dans la page, une erreur à l'exécution vous indiquera qu'un ScriptManager est requis sur la page. Si vous avez créé votre projet de site Web avec le type de projet AJAX-enabled, vous aurez noté que la page par défaut contient déjà un ScriptManager.

Amélioration de l'ergonomie

Le problème de ne pas avoir de rechargement complet de la page c'est que si le serveur met un peu de temps pour exécuter la requête, l'utilisateur peut être tenté de cliquer plusieurs fois car il n'a aucun acquittement visuel de son action. Il est important dans un site web de faire un retour visuel à l'utilisateur pour lui indiquer que son action a été prise en compte. ASP.NET Ajax nous propose un contrôle permettant de signaler à l'utilisateur qu'un traitement AJAX a été envoyé au serveur. Ce contrôle est l'UpdateProgress. C'est aussi un conteneur qui s'affiche pendant un traitement asynchrone. Il suffit donc de le placer sur la page et d'y insérer du contenu, un GIF animé par exemple.

```
<asp:UpdateProgress ID="UpdateProgress1" runat="server">
   <ProgressTemplate>
      <img src="loader-square.gif" />
   </ProgressTemplate>
</asp:UpdateProgress>
```

Et bien évidemment, vous pouvez afficher des UpdateProgress différents pour différents UpdatePanel.

Ajouter un extender

Précédemment dans cet article a été évoquée la possibilité d'étendre ASP.NET Ajax. C'est ce que font les contrôles de la Toolkit. Nous allons ajouter un de ces extenders pour ajouter du comportement à notre GridView, à savoir un texte en surimpression quand on passe sur une colonne. Tout d'abord, il faut référencer la Control Toolkit dans notre projet. Après avoir téléchargé le projet (cf. lien en annexes), il faut faire une référence à AjaxControlToolkit.dll qui se trouve dans un des répertoires d'installation. On peut aussi copier directement cet assemblage dans le répertoire bin du site. Il faut ensuite référencer l'espace de noms des contrôles de la Toolkit, avec une directive <%@Register%>

```
<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControl
Toolkit"    TagPrefix="ajaxToolkit" %>
```

Ensuite, on ajoute la colonne avec le texte qui doit réagir au passage de la souris.

```
<asp:TemplateField>
   <ItemTemplate>
      <asp:Label runat="server" ID="LabelMenu">ICI</asp:Label>
   </ItemTemplate>
</asp:TemplateField>
```

Dans l'ItemTemplate, on ajoute le panneau contenant le texte en surimpression :

```
<asp:Panel CssClass="popupMenu" ID="PopupMenu" runat="server">
   <div style="border: 1px outset white; padding: 2px; background: lightgreen;">
      HoverMenu
   </div>
</asp:Panel>
```

On ajoute ensuite l'extender que l'on configure. Ici on utilise un HoverMenuExtender qui référence le panneau à afficher et le contrôle qui réagit à la souris.

```
<ajaxToolkit:HoverMenuExtender ID="hme2" runat="Server" Popup
ControlID="PopupMenu" TargetControlID="LabelMenu"
   PopupPosition="Left" PopDelay="25" />
```

Bien sûr, les aficionados du Javascript et du CSS ont l'habitude de développer ce genre de comportement mais ici, tout est encapsulé dans un contrôle générique que l'on pose sur la page. Cela simplifie le code et améliore sa maintenabilité.

Configurer Ajax manuellement

Pour disposer des extensions Ajax dans un site web existant, il faut préciser différentes choses dans le fichier web.config. Il faut tout d'abord inclure les extensions Ajax dans la compilation dynamique du site afin que les références vers les nouveaux contrôles puissent être résolues.

```
<compilation debug="false">
  <assemblies>
    <add assembly="System.Web.Extensions" />
  </assemblies>
</compilation>
```

Pour éviter de référencer la bibliothèque de contrôles dans chaque page avec un <%@ Register %>, on peut ajouter cette référence directement dans le fichier de configuration. On en bénéficiera de facto dans toutes les pages.

```
<pages>
  <controls>
      <add tagPrefix="asp" namespace="System.Web.UI" assembly=
"System.Web.Extensions" />
  </controls>
</pages>
```

Si vous utilisez des contrôles de la Toolkit, vous devez aussi ajouter la référence suivante.

```
<add assembly="AjaxControlToolkit" namespace="AjaxControlToolkit"
 tagPrefix="ajaxToolkit" />
```

Il faut ensuite " brancher " le moteur AJAX sur ASP.NET. Pour cela, il faut ajouter les handlers Http et le module Http qui interceptent et traitent les requêtes AJAX.

```
<httpHandlers>
  <remove verb="*" path="*.asmx" />
   <add verb="*" path="*.asmx" validate="false" type="System.Web.
Script.Services.ScriptHandlerFactory, System.Web.Extensions" />
    <add verb="*" path="*_AppService.axd" validate="false" type=
"System.Web.Script.Services.ScriptHandlerFactory, System.Web.
Extensions" />
    <add verb="GET,HEAD" path="ScriptResource.axd" type="System.
Web.Handlers.ScriptResourceHandler, System.Web.Extensions" validate
="false" />
</httpHandlers>
```

Vous remarquerez que cette configuration supprime le support classique des fichiers ASMX (ie. des Web Services). Ce n'est pas tout à fait vrai car le handler proposé par les extensions Ajax encapsule le handler classique via le pattern de conception Decorator.

```
<httpModules>
  <add name="ScriptModule" type="System.Web.Handlers.ScriptModule,
 System.Web.Extensions" />
</httpModules>
```

Pour éviter de retenir par cœur cette configuration fastidieuse, je vous conseille vivement de récupérer ces sections de configuration depuis un site web Ajax existant ou depuis un nouveau site créé avec le nouveau type de projet disponible.

Déploiement

Pour déployer notre site web, il n'y a pas de procédure spéciale. Un site web Ajax est un site ASP.NET classique avec juste quelques extensions. Vous pouvez donc utiliser les techniques de déploiement classique à savoir l'assistant de publication de Visual Studio ou la simple copie de fichiers. L'unique vérification à faire est de s'assurer que les extensions AJAX soient disponibles sur le serveur où le site sera déployé. Pour cela vous pouvez très bien utiliser l'installeur présenté au début de cet article pour installer les extensions. Cependant, on préfèrera déployer sur un serveur le strict nécessaire. Pour fonctionner, les extensions AJAX s'appuient sur un unique assemblage : System.Web.Extensions. Vous pouvez donc ne déployer que cet assemblage. Par défaut, l'installeur le met dans le Global Assembly Cache de la machine et en met une copie à disposition dans le répertoire C:\Program Files\Microsoft ASP.NET\ASP.NET 2.0 AJAX Extensions\v1.0.61025. Si vous avez accès au GAC du serveur, vous pouvez l'y installer. Cette installation se fait par un Drag & Drop de l'assemblage dans le répertoire C:\Windows\assembly ou bien par l'utilisation de l'utilitaire gacutil qui se trouve dans le SDK du Framework .NET : gacutil /i System.Web.Extensions.dll.

Si vous n'avez pas accès au GAC (dans le cas d'un serveur mutualisé par exemple), vous pouvez aussi copier cet assemblage dans le répertoire bin de votre site web. Attention, retenez bien que la version du GAC est prioritaire sur la version dans le répertoire bin d'un site web.

Si vous utilisez des contrôles de la Toolkit, l'assemblage de celle-ci doit déjà se trouver dans votre répertoire bin du projet et sera donc déployé avec l'ensemble du site.

Conclusion

Les extensions Ajax vous permettent d'ajouter facilement une touche d'Ajax dans votre site web ASP.NET. Grâce à Ajax, vous pourrez, entre autres, améliorer l'ergonomie de vos applications ou diminuer la bande passante consommée. Ce Framework est gratuit, facile d'installation et supporté par Microsoft. Il est même accessible aux développeurs utilisant les versions Express des produits Microsoft. Simple à mettre en œuvre, ce Framework est aussi ouvert et extensible, ce qui le rend très puissant et évolutif. Quelques remarques toutefois, Ajax n'est pas synonyme de performances coté serveur ni d'un développement simplifié. Il faut être vigilant à l'architecture que vous mettez en place et veiller à ne pas complexifier à outrance les pages, notamment en évitant le syndrome de la page unique.

Annexes

- Le site officiel ASP.NET Ajax : http://ajax.asp.net/ - téléchargement, documentation, exemples, ressources
- Le site de la Control Toolkit : http://ajax.asp.net/toolkit/
- L'édition gratuite de Visual Studio : http://www.microsoft.com/france/msdn/vstudio/express/default.mspx
- L'édition gratuite de SQL Server : http://www.microsoft.com/france/msdn/vstudio/express/sql/telechargez.mspx

Jean-Pierre Riehl
Jean-pierre.riehl@bewise.fr - Bewise

Création et déploiement d'une application Web 2 avec Rails

Rails est né dans la mouvance du Web 2. Il est par essence un outil de prédilection pour développer dans les meilleures conditions un site de dernière génération.

Quel meilleur exemple pour montrer la création d'un site Web 2 qu'un site communautaire ? Plutôt que de reprendre le perpétuel exemple du système de blog, créons ensemble la base d'une application du type "Yahoo Answers". Le principe est simple : des utilisateurs postent des questions et les autres utilisateurs y répondent. La plupart des applications Web 2 utilisent des requêtes AJAX. Ces requêtes permettent de modifier les éléments de la page sans la recharger complètement. Enfin, les sites de cette génération tendent vers une plus grande ouverture et proposent l'accès à leurs informations à d'autres sites via des Webservices. Nous profiterons de la création de l'application pour y intégrer des exemples illustrant ces deux points.

Mise en place

Comme pour toute application Rails, on commence par créer la structure de l'application, que l'on appelle pour l'occasion CoolAnswers. Les commandes Shell de cet article fonctionnent sous Windows (commande MS-DOS) ou via le Shell d'un système Unix/Linux.

```
$> rails coolanswers
```

Une fois la structure créée, utilisez votre éditeur de texte favori, par exemple TextMate sur MacOS X ou Vim sur Linux, pour éditer les fichiers du projet. Rails crée un fichier de configuration d'accès à la base de données par défaut. Par convention, la base de données MySQL coolanswers_development sera utilisée sur l'hôte local (localhost) via l'utilisateur root et le mot de passe '' (chaîne vide). A vous d'éditer le fichier database.yml dans le dossier config autrement... Pour rappel, Rails est basé sur l'architecture de développement M.V.C (Modèles Vues Contrôleurs). Les modèles interagissent avec les données, les vues forment l'interface utilisateur (l'HTML pour un développement Web) et les contrôleurs traitent les requêtes des utilisateurs. Pour notre application, nous pouvons définir deux modèles. Le premier (Question) définira une question et le deuxième (Réponse) définira une réponse d'un utilisateur.

```
$> cd coolanswers
$> ruby script/generate model Question
$> ruby script/generate model Reponse
```

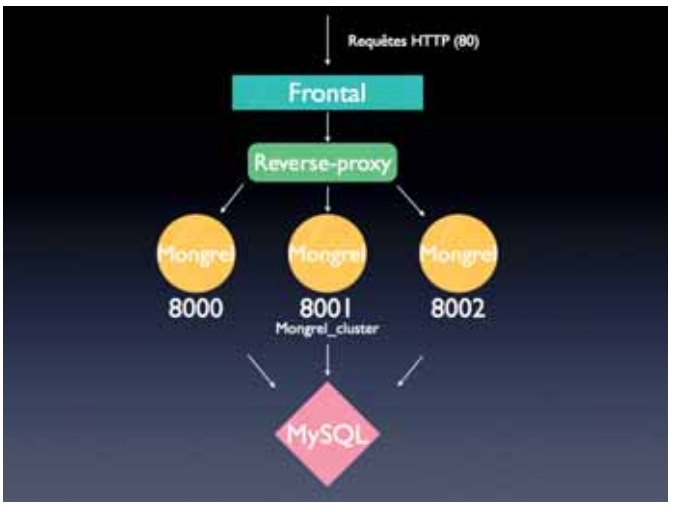
Ces deux commandes génèrent, en plus des modèles en question, deux fichiers de migration qui vont nous permettre de définir en Ruby les deux tables questions et réponses.

```
class CreateQuestions < ActiveRecord::Migration
  def self.up
    create_table :questions do |t|
      t.column 'titre', :string
      t.column 'auteur', :string
      t.column 'contenu', :string
      t.column 'date', :datetime
    end
  end

  def self.down
    drop_table :questions
  end
end

class CreateReponses < ActiveRecord::Migration
  def self.up
    create_table :reponses do |t|
      t.column 'question_id', :integer
      t.column 'auteur', :string
      t.column 'contenu', :string
      t.column 'date', :datetime
    end
  end

  def self.down
    drop_table :reponses
  end
end
```



Une fois les migrations écrites, migrez-les dans la base en croisant les doigts pour que la connexion à la base s'opère avec succès.

```
$> rake db:migrate
```

Si tout se passe comme prévu, les deux migrations sont appliquées et

les deux tables créées. En cas de problème, vérifiez la connexion au SGBD et la configuration dans database.yml.

Développement

Les liens entre les modèles sont simples : une question a des réponses et une réponse appartient à une question. Ecrivons directement ces deux liaisons dans les modèles reponse.rb et question.rb.

```
class Reponse < ActiveRecord::Base
  belongs_to :question
end

class Question < ActiveRecord::Base
  has_many :reponses
end
```

Grâce à ces deux liaisons, il sera possible dans les vues de récupérer les réponses d'une question très simplement :

```
  <%= render :partial => 'reponse', :collection => @question.reponses.
find(:all, :order => 'date desc') %>
```

Ce bout de code affiche les réponses d'une question en bouclant sur le partiel "reponse" pour chaque réponse à la question (paramètre "collection"). C'est encore plus lisible que de faire une boucle standard. Créons désormais un contrôleur Questions. Ce contrôleur permettra de lister les questions, mais aussi d'afficher le détail d'une question avec ses réponses.

```
$> ruby script/generate controller Questions
```

Après avoir lancé le serveur de développement (commande ./script/server), vous pourrez tester l'application en pointant votre navigateur sur http://localhost:3000/questions.
Voici le contenu du contrôleur principal de l'application Questions.

```
class QuestionsController < ApplicationController

  def index
    @questions = Question.find(:all, :order => 'date desc')
  end
```

L'action index récupère toutes les questions dans l'ordre inverse de leur date de création afin d'afficher les questions les plus récentes en premier.

```
  def creer
    @question = Question.new(params[:question])
    @question.date = Time.now
    @question.save
    redirect_to :action => 'index'
  end
```

Cette action crée la nouvelle nouvelle question selon les valeurs retournées par le formulaire et redirige l'utilisateur vers l'index.

```
  def details
    @reponse = Reponse.new
    @question = Question.find(params[:id])
    respond_to do |format|
      format.html
      format.xml { render :xml => @question.to_xml(:include => :reponses) }
    end
  end
```

La plupart des sites Web 2 proposent l'utilisation de leurs services sous forme de Webservices. Rails permet de facilement créer des applications permettant de répondre à différents clients. Dans notre action "details", nous voulons afficher à l'utilisateur les réponses de la question au format HTML. Mais nous voulons aussi permettre à d'autres développeurs de récupérer facilement une question donnée ainsi que ses réponses. Le bloc respond_to permet de définir les différentes réponses possibles et notamment de renvoyer un fichier XML. Dans ce cas, nous renvoyons la question ainsi que ses réponses sous forme d'un XML dans le cas où le client demande un XML.
On peut tester la demande d'un XML en rajoutant ".xml" à l'ID de la question dans l'URL.
Exemple :

```
http://localhost:3000/questions/details/1 => retour HTML
http://localhost:3000/questions/details/1.xml => retour XML

<% form_remote_for :reponse, @reponse, :url => { :action => 'creer_
reponse', :id => @question } do |f| %>

  <table>

    <tr><td>Auteur :</td><td><%= f.text_field :auteur %></td></tr>
    <tr><td>Votre réponse :</td><td><%= f.text_area :contenu %></td></tr>
    <tr><td><%= submit_tag 'Créer' %></td></tr>

  </table>

<% end %>
```

Le formulaire de l'ajout d'une réponse (dans la vue) est relativement simple. Nous utilisons ici la méthode form_remote_for afin de créer un formulaire AJAX autour de l'objet @reponse. Le formulaire sera envoyé à l'action creer_reponse qui se chargera d'insérer la réponse dans la table.

```
  def creer_reponse
    @question = Question.find(params[:id])
    @reponse = @question.reponses.build(params[:reponse])
    @reponse.date = Time.now
    @reponse.save
    redirect_to :action => 'details', :id => @question unless request.xhr?
  end

end
```

L'action creer_reponse récupère la question courante et crée une réponse attachée avec les données récupérées par le formulaire. Lorsque vous développez des applications AJAX, essayez toujours de ne pas casser la compatibilité avec les navigateurs sans JavaScript. Cela vous permettra aussi d'améliorer le référencement de votre application car les robots des

moteurs de recherches ne gèrent pas le JavaScript. Pour garder cette compatibilité, nous testons à la fin si la requête est de type XmlHttpRequest (XHR). Sinon, nous redirigeons de manière standard l'utilisateur. Une fois l'action exécutée, et dans le cas où la requête est AJAX (XHR), Rails enverra au client la réponse JavaScript de la vue RJS ci-dessous.

```
page.insert_html :bottom, 'reponses', :partial => 'reponse', :object => @reponse
page.visual_effect :highlight, 'reponse_' + @reponse.id.to_s
```

Ce code RJS insère la réponse qui vient d'être tapée par l'utilisateur en haut de la liste et ajoute un effet spécial de flash jaune (effet Script.aculo.us highlight) à la réponse. Afin de pouvoir ajouter l'effet sur la réponse qui vient juste d'être ajoutée, il est essentiel de spécifier chaque réponse avec une balise HTML id correspondant à l'id de la réponse dans la base. Lors d'un développement AJAX, il est intéressant d'utiliser avec parcimonie des effets spéciaux sur les blocs et contrôles. Les utilisateurs sont habitués à ce que la page soit rechargée à chaque envoi d'un formulaire ou clic sur un lien. En AJAX, rien ne se passe par défaut... En plus de faire joli, les effets spéciaux permettent donc d'avertir l'utilisateur d'un changement dans la page.

Déploiement de l'application

Le déploiement d'une application Ruby on Rails n'a pas toujours été facile. Durant longtemps, la seule solution était d'utiliser le module FastCGI d'Apache. C'est encore une des manières de déployer une application. Mais la communauté Rails a rapidement voulu trouver une solution plus pratique et plus flexible. Le mini-serveur Ruby Mongrel peut être lancé en "cluster". L'administrateur peut ainsi lancer un nombre indéfini de processus Mongrel. Il met en place un serveur en frontal indépendant qui va rediriger les requêtes vers ces processus Mongrel. Ce procédé de reverse-proxy permet en plus l'équilibrage de charge entre les différents processus en back-office. Cette manière de déployer une application Rails a de nombreux avantages, dont une plus grande souplesse : on peut facilement déboguer les pannes et pallier une éventuelle montée en charge (il suffit de lancer de nouveaux processus Mongrel). Les développeurs de Rails préconisent tous de déployer sur une plate-forme Linux ou Unix. Le déploiement sur un système Windows reste possible mais peu commun et sans grand support de la part de la communauté. La première étape consiste à installer sur le serveur le même environnement Ruby que sur la machine de développement. Après l'installation du système de gestion de base de données et des gems de Rails, il ne reste qu'à installer Mongrel et Mongrel_Cluster.

```
$> gem install mongrel
$> gem install mongrel_cluster
```

Pour cet exemple, nous allons déployer l'application avec Pound. Pound est un reverse-proxy simple, indépendant de tout serveur Web et très simple à configurer. Bien sûr, à vous de choisir votre frontal préféré tout en retenant que le déploiement via Apache 2.2+mod_proxy_balancer reste une méthode très souvent employée. Le principe de Pound est basique. Sa configuration lui indique quel(s) port(s) écouter et vers quoi renvoyer les requêtes. On commence par configurer le cluster de Mongrels de cette manière :

```
$> mongrel_rails cluster::configure -p 8000 -N 3
$> mongrel_rails cluster::start
```

La 1re commande configure le cluster à 3 instances et le début des ports à 8000. Et la deuxième commande lance le cluster. L'application Rails se retrouve disponible sur les ports 8000, 8001 et 8002. Il ne reste plus qu'à configurer Pound pour rediriger les requêtes du port 80 vers ces 3 processus. Il s'occupera lui-même de faire l'équilibrage des charges...

```
ListenHTTP
  Address 0.0.0.0
  Port    80
  Service
    BackEnd
      Address 127.0.0.1
      Port    8000
    End
    BackEnd
      Address 127.0.0.1
      Port    8001
    End
    BackEnd
      Address 127.0.0.1
      Port    8002
    End
  End
End
```

Selon la configuration, il ne vous reste plus qu'à lancer Pound et tester l'application sur le port 80.
Il reste possible à tout moment de manipuler les objets de l'environnement de production. Il suffit pour cela de lancer une console Ruby avec le paramètre production.
Toujours dans le dossier de votre application, lancez :

```
$> ruby ./script/console production
```

Cela est très utile pour trouver la cause de certains problèmes survenant lors de la mise en production...

Aller plus loin...

Voici quelques exercices à faire pour améliorer l'application ou son déploiement...

- Rendre le formulaire de l'ajout d'une question AJAX de la même manière que le formulaire d'ajout d'une réponse.
- Programmer un système d'enregistrement et de connexion pour les utilisateurs. Vous pouvez par exemple utiliser le générateur de acts_as_authenticated (http://technoweenie.stikipad.com/plugins/show/Acts+as+Authenticated).
- Faciliter le déploiement de l'application en mettant en place Capistrano (http://manuals.rubyonrails.com/read/book/17). Capistrano permet de facilement réitérer le processus de déploiement en utilisant un dépôt.
- Ecrire les tests unitaires...
- Tester différents frontaux/reverse-proxy, dont le fameux nginx (http://nginx.net) qui permet d'obtenir d'excellentes performances.

Nicolas Blanco - *slainer68@gmail.com*

Des RIA en Java avec JavaFX Script

Sun Microsystems a dévoilé une nouvelle technologie nommée JavaFX lors de JavaOne 2007. Sous ce nom se cachent en réalité plusieurs produits bien distincts dont le plus abouti, JavaFX Script, permet de développer des applications riches plus rapidement.

JavaFX Script désigne un nouveau langage de script imaginé et réalisé par Chris Oliver chez Sun Microsystems. JavaFX Script appartient à une nouvelle famille de produits qui permettront aux développeurs de réaliser rapidement et facilement des applications pour appareils mobiles, desktops et sites Internet. JavaFX Script, autrement connu sous le nom de code "F3" (pour Form Follows Function), est donc un nouveau langage dynamique et interprété, conçu pour créer des interfaces graphiques en Java. Le langage, qui possède des caractéristiques très particulières, est accompagné d'une API reposant sur les API Swing et Java2D de la plate-forme Java SE. JavaFX Script est donc portable sur tout système supportant Java SE et bâti sur des composants fiables et éprouvés. Le runtime JavaFX Script, ainsi que des plug-in pour NetBeans et Eclipse, est disponible gratuitement et sous licence Open Source sur le site officiel, http://openjfx.dev.java.net.

Pourquoi un nouveau langage ?

Après l'annonce à JavaOne, de nombreux développeurs ont critiqué Sun Microsystems pour avoir choisi un nouveau langage plutôt que d'utiliser un produit existant, comme JavaScript ou encore Groovy. Malgré leurs qualités, ces langages n'ont jamais été pensés pour la réalisation d'applications riches et présentent donc de nombreuses faiblesses par rapport à JavaFX Script. Bien qu'il soit impossible d'énumérer ici toutes les fonctionnalités de ce langage, nous allons nous pencher sur certaines des plus intéressantes au travers d'un exemple relativement simple. Le CD-Rom accompagnant ce magazine contient un projet NetBeans intitulé JavaFxViewer. Ce projet, également exécutable dans Eclipse ou directement avec le runtime JavaFX, propose un visualiseur d'image animé.
La caractéristique majeure de JavaFX Script est sa capacité à pouvoir définir des interfaces graphiques de manière déclaratives, un peu à la manière de XML. Cela permet de s'affranchir d'appels de méthodes, de créations d'instances, etc. Le petit exemple ci-dessous déclare une nouvelle fenêtre aux dimensions fixes, contenant un bouton affichant un message dans la console, lorsqu'on le presse :

```
Frame {
    title: "JavaFX"
    width: 500
    height: 390
    content:
    [Button {
        text: "Hello"
        action: operation() { System.out.println("Hello World!); }
    }]
    visible: true
}
```

La déclaration de l'interface graphique permet de mieux représenter la hiérarchie des composants tout en offrant une syntaxe plus claire. Comme vous le voyez ici, JavaFX Script permet d'assigner des valeurs aux propriétés JavaBeans des composants Swing JFrame et JButton. Le runtime JavaFX Script propose une API pour tous les composants Swing, ainsi que plusieurs composants supplémentaires provenant de divers projets Open Source (comme StackLayout, venu du projet SwingX).
Cette capacité ne suffit malheureusement pas à rédiger des programmes complets. JavaFX Script permet donc de créer des classes, des méthodes et des fonctions pour soutenir la logique de votre application.

Afficher des images

Pour afficher des images nous allons créer un nouveau composant ImageViewer. Un composant JavaFX peut dériver de la classe Widget ou de CompositeNode. Les Widgets peuvent êtres assimilés aux composants Swing tandis que les CompositeNode sont ajoutés dans un Canvas. Les CompositeNode ont l'avantage de gérer des fonctionnalités de dessin avancées, comme l'opacité ou encore des filtres (par exemple, un flou). Notre ImageViewer sera donc défini comme un CompositeNode :

```
class ImageViewer extends CompositeNode {
    attribute imageList:String+;
    attribute imageUrl:String;

    private attribute imageIndex:Number;
    private attribute animation:Number;

    operation next();
    operation previous();
}
```

La définition d'ImageViewer introduit le concept d'attribut, ou variable d'instance, et d'opération, ou méthode. Cet exemple introduit également la notion de cardinalité. Le "+" dans la déclaration de l'attribut imageList indique qu'il s'agit d'une liste de String qui doit contenir au moins un élément (on peut également utiliser les cardinalités vide, ? et*). Contrairement à Java la valeur des attributs et le code des méthodes ne peut être intégré dans la définition de la classe, mais doivent êtres placés en dehors. L'exemple suivant présente la définition des attributs et méthodes d'ImageViewer :

```
attribute ImageViewer.animation = 0;
attribute ImageViewer.imageIndex = 0;
attribute ImageViewer.imageUrl = bind imageList[imageIndex];

operation ImageViewer.next() {
    imageIndex++;
}

operation ImageViewer.previous() {
    imageIndex-;
}
```

Ces définitions contiennent une des fonctionnalités les plus importantes et intéressantes de JavaFX Script, le binding. L'attribut imageUrl de la classe ImageViewer est ici lié à l'expression imageList[imageIndex]. Cela signifie que chaque fois que vous changerez imageIndex ou le contenu d'imageList, JavaFX Script mettra automatiquement à jour la valeur d'imageUrl. Cette capacité vous affranchit des systèmes d'événements habituels et d'une grande partie de la logique des applications traditionnelles. Ainsi, lorsque les opérations next() et previous() sont invoquées, la valeur d'imageUrl sera mise à jour. L'implémentation de ces opérations serait bien plus longue dans un programme écrit en Java/Swing. La liaison de données est ainsi utilisée très couramment dans les applications JavaFX Script. Notre ImageViewer définit une autre méthode, composeNode() surchargée de CompositeNode, pour créer le composant lors de l'affichage :

```
operation ImageViewer.composeNode() {
    return ImageView {
        image: Image { url: bind imageUrl }
    }
;}
```

Cette opération retourne un composant ImageView capable d'afficher une image chargée depuis une URL quelconque. Puisque nous disposons d'un attribut imageUrl contenant à tout instant l'URL de l'image désirée, il nous suffit de lier cet attribut à l'attribut url de la classe Image. Lorsque nous changerons imageIndex, imageUrl sera mis à jour, ce qui provoquera alors la mise à jour d'Image.url et finalement la mise à jour de l'affichage. Nous pouvons enfin instancier notre composant avec une liste d'images et l'intégrer à notre fenêtre :

```
var viewer = ImageViewer {
    imageList: [ "urlImage1", "urlImage2", "urlImage3" ]
};

Frame {
    title: bind "JavaFX: {viewer.imageUrl}"
    content:
    [BorderPanel {
        center: Canvas {
            content: [ viewer ]
        }
    }]
    visible: true
}
```

La fenêtre définie ici contient un Canvas qui contient lui-même notre visualiseur d'images. Il est intéressant de noter la définition de l'attribut title de la fenêtre. La liaison de données est ici réalisée avec une chaîne de caractères comprenant une expression JavaFX Script. Une fois de plus, le runtime fera les mises à jour nécessaires sans aucune intervention de votre part.
Notre application nécessite enfin des boutons pour passer d'une image à l'autre. Une fois de plus, le mot-clé bind est utilisé pour activer ou désactiver les boutons lorsque nécessaire. Le bouton ci-dessous est grisé lorsqu'ImageViewer affiche la première image et un clic dessus exécute l'opération ImageViewer.previous().

```
Button {
    text: "Previous"
    enabled: bind viewer.imageIndex > 0
    action: operation() { viewer.previous(); }
}
```

Et les animations ?

Bien que le visualiseur d'image soit fonctionnel, le passage d'une image à l'autre est un peu brutal. Grâce à JavaFX Script, nous pouvons très facilement ajouter un fondu enchaîné lors d'une transition entre deux images. Le code nécessaire pour cela repose sur un mot-clé original de JavaFX Script appelé dur. Ce mot-clé permet de faire varier la valeur d'un attribut dans le temps. Par exemple, la ligne suivante fait varier la valeur de "animation" entre 0 et 100 sur une période de 1000 millisecondes :

```
var animation = [0..100] dur 1000;
```

Puisque notre classe ImageViewer contient un attribut animation, nous allons nous en servir pour faire varier l'opacité de l'image affichée. Pour cela, il nous suffit de définir l'attribut imageOpacity dans le composant ImageView créé par l'opération composeNode(). Cet attribut accepte une valeur entre 0.0 et 1.0 définissant le pourcentage d'opacité de l'image. En faisant varier animation entre 0 et 100, nous pouvons définir imageOpacity de la sorte :

```
imageOpacity: bind animation / 100
```

La dernière étape consiste à faire démarrer notre animation quand nous passons d'une image à l'autre. Cette opération a lieu lorsque nous modifions la valeur de l'attribut imageIndex. For heureusement, JavaFX Script permet de définir des triggers, similaires à ceux qui sont utilisés dans les SGBDR, qui permettent d'intercepter des modifications de valeur. Le trigger suivant modifie la valeur de l'attribut animation sur une période d'une seconde chaque fois qu'imageIndex est modifié :

```
trigger on ImageViewer.imageIndex = value {
    animation = [0..100] dur 1000;
}
```

Les fonctionnalités très spécialisées de JavaFX Script ont permis d'écrire un visualiseur d'images animé en 70 lignes de code. Les triggers, bind et dur ne sont que quelques-unes des fonctionnalités fabuleuses de ce langage qui vous permettront de réaliser vos applications très rapidement.

Romain Guy - *romain.guy@mac.com*

L'information à la carte

1 1 an ECO

Recevez le magazine chaque mois et ***économisez 20 €***

-30%

11 Numéros

45 €

Au lieu de 65,45 €
(Prix au numéro)

(Prix France métropolitaine)

2 1 an TOUT NUMERIQUE

Lisez chaque mois le magazine seul
au format PDF
(téléchargement)

sur la base tarif France
-46%

11 Numéros

35 €

Tarif Monde entier

Inscription :
www.programmez.com

3 1 an ETUDIANT

Vous devez justifier de votre statut d'étudiant
Economisez 26 €

-40%

11 Numéros

39 €

Au lieu de 65,45 €
(Prix au numéro)

(offre réservée France métropolitaine)

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

☐ 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

Programmez! LE MAGAZINE DU DÉVELOPPEMENT

Offre limitée, valable jusqu'au 31 août 2007

PROG 99

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.

Epita ou Epitech ?

Epita ou Epitech sont deux écoles du groupe Ionis, mais ne veulent rien avoir en commun sur le plan de la formation et de la pédagogie. Et même si l'informatique est le point commun de ces deux formations qui conduisent en 5 ans au diplôme, **Fabrice Bardèche,** directeur général du Groupe Ionis insiste sur les deux profils d'étudiants qui se répartissent entre ces écoles : *" L'Epitech forme des experts techniques à dimension internationale. L'expérimentation est la source même de l'enseignement, ce qui permet à des passionnés d'être opérationnels à très haut niveau.*

La philosophie d'Epita est un peu différente puisqu'elle forme des ingénieurs informaticiens qui, s'ils travaillent également beaucoup en projets, devront également justifier d'un bon niveau scientifique et en particulier en mathématiques pour suivre les cours. " En d'autres termes, un passionné en informatique qui n'aurait pas un bon niveau en mathématiques a toutes ses chances à l'Epitech où il trouvera un enseignement adapté.

Epita : former des ingénieurs informaticiens

L'Epita a fait le pari de former des ingénieurs informaticiens, aussi doués en équations qu'en codage. Défi réussi : elle obtient une excellente cote lorsqu'elle doit les placer dans le monde professionnel. L'école vient enfin d'obtenir le droit de délivrer le diplôme d'ingénieur.

Art mineur l'informatique ? C'est une affirmation que réfute **Joël Courtois,** directeur de l'Epita (Ecole pour l'informatique et les techniques avancées), dont la première promotion d'ingénieurs informaticiens est sortie en 1989 : ils étaient alors une quarantaine d'étudiants entrés en 1984, passionnés par cette discipline du vingtième siècle et dotés de sérieuses bases scientifiques. *" Un bon expert doit avoir de nombreuses connaissances. Il doit avoir un socle scientifique pour trouver un langage commun avec un ingénieur agronome par exemple. Les mathématiques sont un moyen de communication, une base indispensable pour pouvoir exercer des responsabilités dans le secteur informatique, comme mener à bien un projet. "* Le mythe de l'école privée (6100 € par an en cycle préparatoire pendant deux ans, 7700 € par an pour les trois années du cycle ingénieur-informaticien), dans laquelle les parents fortunés inscrivent le rejeton pour qu'il obtienne un " diplôme maison ", tombe : à l'Epita, une première sélection est faite sur le dossier scolaire. Certes, des accidents de parcours peuvent arriver, mais Joël Courtois ne laisse que peu d'espoir à ceux qui obtiendraient le bac scientifique avec un carnet scolaire médiocre. " Avec le jeu des coefficients, on peut avoir le bac avec une moyenne de 7/20 en mathématiques. C'est bien de pouvoir se rattraper ailleurs, mais ce n'est pas exactement le profil de nos étudiants qui vont devoir pendant 5 ans manier des outils scientifiques! "

Une sélection drastique pour conserver un haut niveau

Sur 2 000 candidats qui se présentent chaque année environ 250 d'entre eux vont intégrer le centre. Et pour les heureux élus, pas question de se reposer sur cette première sélection : *"nous leur apprenons à travailler "*, insiste Joël

Courtois. Fi donc des étudiants qui penseraient passer quelques années sans s'investir, le travail est le maître-mot et la programmation a sa place dans les cours, même si le directeur de l'Epita reconnaît qu'à la sortie de l'école, ses élèves " auront davantage un rôle d'intégrateur que de développeur. Mais nous nous devons de former des architectes capables de monter des briques. "

Sans le dire, l'Epita semble aussi réfléchir aux mutations inhérentes à la discipline en général et aux métiers liés à l'informatique en particulier. Aujourd'hui, nombre de seniors sont sur la touche. Dotés de solides connaissances scientifiques, les futurs diplômés auront davantage de possibilités de rebondir si la situation sur le marché du travail venait à être difficile ou s'ils devaient se reconvertir dans d'autres domaines, puisqu'une carrière rectiligne, avec un métier unique dans une seule entreprise ne semble plus être aujourd'hui la norme dans le secteur privé.

D'où un rythme particulièrement soutenu : le socle commun scientifique, complété par un enseignement en informatique dont les méthodes pédagogiques privilégient projets et tests. Différents laboratoires coexistent : SRS (systèmes, réseaux et sécurité), SIGL (systèmes d'information et génie logiciel), MMA (multimédia et les technologies de l'information), SCIA (sciences cognitives et informatique avancée), TCOM (télécommunications), GISTR (génie informatique des systèmes temps réel) ou CSI (calcul scientifique et image). Et pour ce faire, la convivialité est de rigueur puisqu'à l'instar des campus anglo-saxons, l'école et les salles d'ordinateurs restent ouvertes... 7 jours par semaine et 24h/24h !

" Jamais seul "

Interdiction donc aux petits génies de jouer les artistes du code en solo. *" Dans un projet, c'est l'équipe qui prime. Il faut donc apprendre à travailler en groupe, seul on n'est rien ",* martèle, convaincu, Joël Courtois.

Pas facile pour des métiers où, selon une étude commandée début 2007 par le Syntec Informatique et ayant abouti à une campagne d'attractivité du secteur informatique auprès des collégiens et lycéens, l'individualisme est souvent de mise et les " comportements autistiques " particulièrement ressentis.

Epita veut donc casser ces habitudes dont on taxe les informaticiens et favoriser la communication par la multiplication d'expériences qui permettront aux futurs chefs de projets de s'adapter au travail sur-mesure et aux interlocuteurs multiples. Pour ce faire, un projet d'école presque terminé aura un zéro pointé. Car le " presque " n'existe pas. " Il faut apprendre l'impératif client. En entreprise, il faut de la rigueur ", poursuit le directeur de l'Epita. Et surtout, au cours de leurs années d'études, les futurs diplômés auront des stages à faire en entreprise et des expériences à l'étranger. " En 2010, 100% de nos étudiants auront été en stage hors de nos frontières. C'est une grosse logistique à mettre en place, mais dans le monde qui se dessine, il est désormais indispensable d'apprendre à travailler ailleurs. Avec d'autres cultures et d'autres méthodes. "

■ **Claire Padych**

Un salaire compris entre 32 et 35 K€

Une étude de l'Epita indique que le taux d'insertion en entreprise pour la promotion 2006 était de 100% : 92% des étudiants avaient trouvé un emploi avant la fin de leurs études, 4% moins d'un mois après leur sortie de l'école et 4% dans un délai compris entre un et trois mois. Quant aux salaires, la même source indique qu'ils sont compris entre 29 K€ et 38 K€ avec 59% de diplômés touchant entre 32 K€ et 35 K€ et 21% se situant dans une fourchette de salaire comprise entre 35 K€ et 38 K€.

Un travail en rapport avec leur spécialisation

En 2004, l'Epita a présenté une enquête à l'occasion de ses 20 ans. Il en résulte que 50% des anciens travaillent toujours dans le secteur informatique : 25% en SSII, 14% chez les éditeurs, 8% dans l'industrie informatique et 2% dans la net-économie. Mais l'évolution vers d'autres secteurs est nette : 10% travaillent dans le domaine "banquefinances-assurances" et 10% chez des industriels (hors informatique). Enfin, 60% des anciens (promotions 89 à 95) ont un travail en rapport avec la spécialisation suivie à l'Epita.

Le Graal : l'obtention du titre " d'ingénieur " par la CTI

La Commission des Titres d'Ingénieurs (CTI) est la clé de voûte du dispositif de délivrance du titre d'ingénieur diplômé. Placée auprès de la Direction des enseignements supérieurs, elle a un statut d'instance juridictionnelle pour les formations privées et consultative pour les formations sous tutelle publique. Elle donne ainsi un avis sur toute création de formation d'ingénieur et elle effectue des inspections périodiques tous les six ans, pour prendre position sur la poursuite de l'habilitation.

L'Epita ne possèdait pas ce fameux " Graal " CTI, qu'elle demandait depuis des années. Le dossier a été enfin accepté en juin 2007.

Ne pas l'obtenir était-il un handicap ? Si pour ses étudiants une reconnaissance CTI représente un indéniable atout supplémentaire, l'Epita a toujours bénéficié d'une aura réelle dans le monde professionnel (dans le Top 5 des écoles les plus cotées selon plusieurs classements). Une stricte sélection de ses étudiants et un niveau de formation suffisamment élevé sont deux critères majeurs qui lui ont permis de rivaliser avec les écoles d'ingénieurs labellisée CTI. Ses ingénieurs informaticiens sont recherchés par les entreprises et les cabinets de recrutement car les précédentes générations ont fait leurs preuves et ouvert la voie à ceux qui arrivent. A noter encore que la reconnaissance par l'Etat du diplôme de l'Epita au niveau 1 (Bac+5) est régulièrement reconduite depuis plus de 10 ans.

Epitech : "former des informaticiens capables de s'adapter tout au long de leur vie".

L'Epitech est née en 1999. Ses premières " promos " commencent donc à sortir, avec des jeunes diplômés qui ont la particularité de rejoindre tous les domaines de l'entreprise. Attentive aux suggestions des entreprises, l'équipe pédagogique a pourtant façonné sans l'aide de personne son propre modèle, afin de faire de ses diplômés, des informaticiens capables de s'adapter tout au long de leur vie.

Nicolas Sadirac est le directeur de l'Epitech, une école qui pourrait se définir comme une structure à la vocation première d'être ouverte aux passionnés d'informatique, et de les faire progresser dans leur domaine de prédilection en vue de leur permettre d'exercer un métier. *" L'avantage d'être une école privée est de pouvoir expérimenter. Nous avons une vraie capacité à être interactifs avec les réalités, de se donner les moyens de mettre en œuvre un modèle atypique mais fonctionnel, "* indique son directeur.

Et pour ce qui est de l'originalité, il ne semble pas exister de limites pédagogiques dans cette école.

A commencer par le bizutage qui suit les élèves qui viennent de s'inscrire en première année : *" pendant quatre semaines, ils sont dans le bain en faisant quinze heures de programmation par jour, indique Nicolas Sadirac. Le but est de leur montrer le rythme et la nature de ce qu'on peut leur demander ! "*

Ce premier épisode n'est qu'un aspect d'une pédagogie qui est fondée sur l'autonomie afin que l'étudiant, soumis à un problème, expérimente les solutions qui s'offrent à lui de manière proactive " en ne raisonnant surtout pas avec des certitudes. "

Un informaticien ne doit pas apprendre les solutions… mais les trouver selon les problèmes donnés

" Les ingénieurs formés il y a vingt ans avec des réponses toutes faites, ne parviennent pas à se reconvertir lorsque leur environnement technologique ou scientifique se modifie. Contrairement à ce qui est souvent prôné, nous sommes persuadés que ce ne sont pas les bases intangibles qui permettent d'évoluer, mais le raisonnement et la capacité à trouver les solutions, " poursuit Nicolas Sadirac. Et pour mettre à l'épreuve les étudiants, rien ne leur est épargné : des projets à mener en un temps limité et qui sont modifiés en cours de route, des pièges à différentes étapes, une profusion de problèmes technologiques… " S'ils sont dans les métiers informatique, il ne faut pas oublier qu'ils sont aussi en contact avec l'humain, une équipe de techniciens d'un côté et des clients pour lesquels ils travaillent, de l'autre. En entreprise, ils devront apprendre à gérer les différents paramètres et plus tôt ils seront confrontés à ces réalités, mieux ils réagiront. " Mais l'Epitech a tout prévu pour ne pas laisser ses étudiants désemparés face à cette multiplicité de problèmes : des coachs analysent tous les quinze jours l'avancée des projets, des cours de communication et de gestion des conflits sont au programme, ainsi que des cours de théâtre qui ont pour vocation d'apprendre la négociation.

De la polyvalence

L'Epitech prône aussi la polyvalence pour que ses étudiants puissent évoluer dans les différents secteurs. Les certifications ne sont pas préconisées, car elles risquent de réduire les possibilités de l'informaticien. L'école a également conscience du poids de l'international dans une carrière : les cours sont donnés en anglais dès la quatrième année, " à la fin de la troisième année, ils doivent avoir 750 au TOIC, " insiste Nicolas Sadirac. L'Epitech a aussi signé des partenariats avec la Chine, l'Inde et la Russie où ses étudiants découvrent les réalités de l'offshore. Ainsi, les cinquante étudiants qui sont en Chine font trois heures de chinois par jour. Ils partagent des dortoirs avec les Chinois et travaillent avec eux. Selon Nicolas Sadirac, une quinzaine d'entre eux parlent couramment chinois, une vingtaine le parlent sans l'écrire et le reste est capable de se débrouiller dans la vie courante. " Toute cette formation leur permettra d'apporter une incontestable valeur ajoutée à l'entreprise qu'ils rejoindront à la sortie de leurs études, dans la mesure où ils auront vécu sur place une confrontation culturelle et intellectuelle, " conclut Nicolas Sadirac. ■ **Claire Padych**

Un travail en entreprise dès la troisième année

L'Epitech encourage ses étudiants à travailler en entreprise. Toutes les solutions sont possibles : CDI, temps partiel, convention de stage ou junior entreprise… Cela permet à 40% des étudiants de financer leurs études dès la troisième année, de 70% à 80% de le faire en quatrième année et 100% financent leur scolarité en cinquième année.

Un salaire moyen de 34 817€

Selon l'Epitech, 94% des étudiants sont embauchés dans l'entreprise qui les a reçus en stage, 95% en CDI et 93% sous le statut " cadre " ou " cadre supérieur " (contre une moyenne de 86% pour les étudiants des autres écoles d'ingénieurs et 45% pour les étudiants des universités). Leur salaire moyen est de 34 817€ (contre 28 840 € en moyenne pour les étudiants des autres écoles).
www.epitech.eu/

Ma petite entreprise...

Serges Lages, Fabien Rioli et Nicolas Remia ont monté leur structure en décembre 2006, Tharsis Software, destinée à promouvoir les applications de la technologie Magrathea qu'ils ont développée. Ils peuvent proposer une solution personnalisée aux entreprises à la recherche d'innovations en 3D, car Magrathea est un moteur graphique capable de reconstituer l'espace et ses planètes en trois dimensions. Il est multi-plate-forme (Windows, Unix/Linux, Mac OS), adapté à la simulation terrestre, aéronautique, aérospatiale, mais aussi à l'urbanisme, à l'architecture, à la géo-localisation ou au ludo-éducatif.

Leur particularité : ils sont toujours étudiants et obtiendront leur diplôme en septembre, à l'issue de leur cinquième année à l'Epitech. Contrairement à leurs camarades

de promotion qui sont en stage de fin d'études dans des sociétés de tailles diverses, ils font le leur au sein de leur petite entreprise. Cette liberté aurait été difficile dans une autre école qui aurait pu briser les ailes de ces trois chercheurs très motivés et particulièrement innovants. Mais leur projet ayant séduit les enseignants de l'Epitech, ceux-ci ont validé la performance qui a dépassé le cadre expérimental pour devenir un projet professionnel d'envergure. Aujourd'hui, avec la tête dans les espaces multidimensionnels, mais la tête sur les épaules, les trois techniciens ont commencé à signer des contrats !
http://www.magrathea-engine.org
http://www.tharsis-software.com/

" Pour bien comprendre l'informatique, il faut mettre les mains dans le cambouis ! "

" Le secret, c'est d'être un passionné d'informatique "

Guillaume Denoix, jeune diplômé (promotion 2006) a intégré le groupe Axa.

Programmez ! : Votre stage vous a-t-il servi de pré-embauche ?

G.D : Je suis entré à Axa lors de mon stage de fin d'études mais je ne l'ai pas envisagé comme une pré-embauche. j'ai réalisé les avantages à travailler au sein d'un grand groupe, avec des SI que l'on voit rarement ailleurs tant au niveau des contraintes que des technologies et moyens mis en œuvre.

Programmez ! : Quels conseils donneriez-vous à ceux qui veulent s'engager dans une carrière informatique ?

G.D : Il faut être passionné ! C'est le secret pour bien réussir sa scolarité à l'Epitech. Le piège est d'intégrer une école parce qu'on aime jouer aux jeux vidéo. Il faut savoir que l'Epitech ne forme pas des utilisateurs mais des informaticiens aptes à créer, inventer et trouver des solutions. Aujourd'hui, je mets en application ce que j'ai appris pendant cinq ans : analyser et comprendre un problème qui a une incidence sur la vie économique de l'entreprise, et trouver les moyens pour le résoudre.

Mettez du carburant dans votre mobile

Le développement sur téléphone portable suscite l'intérêt et la curiosité, tant par le côté défi technique que par la cible privilégiée qu'offre cet appareil : tellement de personnes sont équipées de mobiles… Et pourtant rares sont les développeurs qui franchissent le pas. L'objectif de cet article est de vous montrer les pistes qui vous amèneront peut-être, à créer votre première application mobile sur la série s40 de Nokia.

Pour cela, voici mon expérience personnelle en la matière : l'idée est née de créer une application afin d'avoir un suivi de la consommation d'essence de mon véhicule. En effet, j'utilisais la calculatrice de mon téléphone portable pour faire le ratio entre le nombre de litres consommés sur le nombre de km effectués. Voila comment NokiaEssence est né…

Côté matériel, je disposais de 2 téléphones portables Nokia : un 3100 et un 6136. Les mobiles chez Nokia se décomposent en 3 grandes séries : s40, s60 et s80. Mes 2 portables font partie de la série s40. Le site http://forum.nokia.com m'a rapidement renseigné sur les possibilités de ces 2 téléphones. Aux vues des spécifications, mon choix c'est rapidement porté vers le modèle 6136.

Nokia 6136 Technical Specs	EXPLICATIONS	Nokia 3100 Technical Specs
Operating System: Nokia OS Developer Platform: Series 40 3rd Edition, Feature Pack 1 Java Technology: MIDP 2.0 CLDC 1.1 JSR 120 Wireless Messaging API JSR 135 Mobile Media API JSR 172 Web Services API JSR 184 Mobile 3D Graphics API JSR 185 JTWI JSR 205 Wireless Messaging API JSR 226 Scalable 2D Vector Graphics API JSR 75 FileConnection and PIM API JSR 82 Bluetooth API Nokia UI API	La base de la plate-forme java pour développement mobile : - MIDP (Mobile Information Device Profile) et - CLDC (Connected Limited Device Configuration) Pour information, CLDC 1.0 ne permet de gérer que les nombres entiers, pas les nombres flottants, contrairement à CLDC 1.1. JSR : Java Specification Request JSR 75 : Package optionnel qui fournit des APIs utiles pour l'accès au système de fichiers du mobile.	**Operating System :** Nokia OS **Developer Platform :** Series 40 Developer Platform 1.0 **Java Technology :** Nokia UI API MIDP 1.0 CLDC 1.0 JSR 120 Wireless Messaging API

Tableau 1

Les logiciels

J'ai donc commencé par installer tous les logiciels nécessaires (Tableau 2). J'ai été ensuite très agréablement surpris de m'apercevoir qu'un matériel de formation était à disposition de tous : trois vidéos permettent de comprendre et de se familiariser avec ce qui va devenir l'environnement de développement au sens large. Par la suite, des tutoriaux indispensables (installés automatiquement avec Carbide.j) expliquent pas à pas comment construire ses premières applications mobiles et comment utiliser Eclipse.

Logiciels nécessaires :

- Jdk 1.4 http://java.sun.com
- Eclipse 3.1 http://www.eclipse.org
- Carbide.j 1.5 http://forum.nokia.com
- SDK S40 3rd Edition Feature Pack 1 http://forum.nokia.com
- Nokia pc suite (pour transférer les fichiers entre le PC et le téléphone portable)

Tableau 2

L'environnement de développement

L'IDE utilisé, Eclipse, propose un éditeur agréable avec le confort habituel des IDE : coloration syntaxique, complétion de code, assistants etc. Le " refactoring " (possibilité par exemple de renommer intelligemment une méthode, un attribut ou paramètre partout dans le code) est un plus agréable. Le plug-in " Carbide.j " avec le " SDK S40 3rd Edition Feature Pack " permet de créer et régénérer un package application. Le SDK met à disposition un émulateur très proche du téléphone réel. Et chose très utile, il est possible d'utiliser le débuggeur d'Eclipse pour tracer le code de l'application qui s'exécute sur l'émulateur.

Au commencement était la MIDlet…

Le début de tout développement mobile Java est de créer une MIDlet : une MIDlet est un programme java pour l'informatique embarquée, plus spécifiquement pour les machines virtuelles Java ME (Mobile édition – J2ME en abrégé). Une MIDlet doit se conformer au MIDP (Mobile Information Device Profile) qui est un ensemble d'API J2ME définissant la façon dont les applications se connectent à l'interface des téléphones cellulaires. La classe principale de l'application doit être une classe dérivée de la classe " javax.microedition.midlet.MIDlet ". Cette classe permet d'implémenter 3 méthodes : startApp, pauseApp et destroyApp qui seront appelées respectivement au démarrage, à la mise en pause, ou à l'arrêt de l'application. Les APIs disponibles pour le développement J2ME sont à rechercher dans le package javax.microedition. Ensuite la MIDlet doit être packagée dans un fichier " .jar ", ce que Carbide.j permet de faire aisément.

L'affichage

L'interface utilisateur étant réduite, même si plusieurs MIDlet s'exécutent simultanément, une seule affiche son écran. La hiérarchie de classe mise en place pour l'interface utilisateur est basée sur la classe Displayable. Deux classes abstraites dérivent de cette précédente :

Screen et Canvas. Les classes concrètes dérivées de Screen fournissent une API de haut niveau afin de permettre l'affichage de fenêtres. Voici le code pour afficher une boîte de dialogue :

```
public class HelloWorldMIDlet extends MIDlet
{
   public void ShowMessage(String aTitre, String aAlertText, AlertType aAlertType)
   {
     // Création de la boite de dialogue " Alert "
     Alert myAlert = new Alert(aTitre, aAlertText, null, aAlertType);
     // Affichage de la boite de dialogue
     Display.getDisplay(this).setCurrent(myAlert);
   }

   protected void startApp( )
   {
     ShowMessage("Hello world","Hello world",AlertType.INFO);
   }
   protected void pauseApp( )
   {
   }

   protected void destroyApp( boolean p1 )
   {
   }
}
```

Les composants

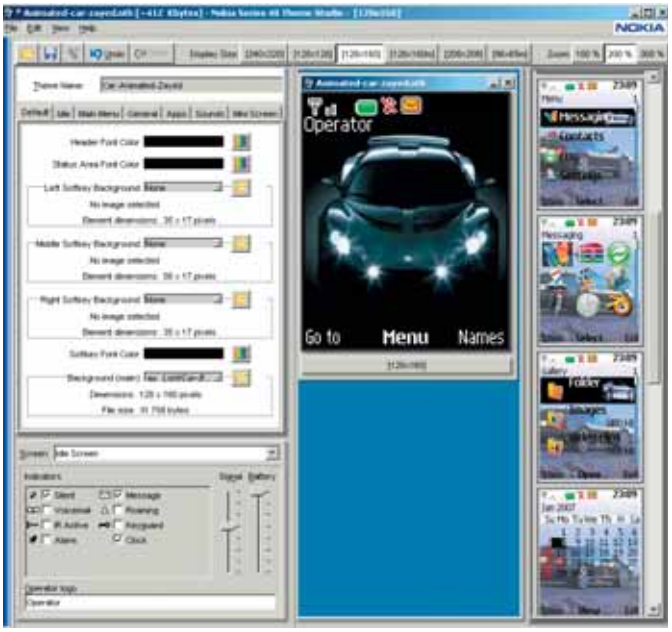
La classe Form permet de construire les écrans de l'application. Grâce au plug-in Carbide.j, un " Designer " de formulaire (Voir image IDENokia-DesignerLite) permet de créer visuellement les écrans en déposant (depuis la palette) les composants sur le formulaire. Un inspecteur d'objet permet d'éditer les valeurs des propriétés.
Voici les objets proposés dans la palette de composants :

Composants	Fonctionnalité
ChoiceGroup	Sélectionner un ou plusieurs éléments de son choix.
DateField	Sélectionner une date. En fonction de l'interface utilisateur, peut proposer un calendrier pour aider à la sélection de la date.
Gauge	Gauge de progression.
ImageItem	Insérer une image dans le formulaire.
StringItem	Afficher du texte, des boutons, des liens hypertexte…
TextField	Saisie de chaîne de caractères, de nombres, décimaux etc.
Ticker	Texte défilant.
Command	Représente une opération utilisable depuis l'interface utilisateur.

Tableau 3

Le composant Command a pour particularité de proposer à l'utilisateur d'interagir avec l'application. Pour que la commande soit utilisable, le formulaire doit implémenter l'interface CommandListener. On retrouve typiquement la programmation événementielle…

```
// Implémentation de l'interface CommandListener :
public void commandAction(Command aCommand, Displayable aDisplayable)
{
if ( aCommand == commandOk ) ActionOk();
```




```
if ( aCommand == commandSauver ) ActionSauver();
// etc.
}
// Au préalable il faudra enregistrer ces commandes à l'initialisation de la form grace à ces instructions :
this.addCommand(commandOk);
this.addCommand(commandSauver);
// Cette instruction définit l'interface comment listener d'événement
this.setCommandListener(this);
```

Le graphisme

Des APIs de plus bas niveau sont aussi disponibles : la classe Canvas est à la base de tout nouveau graphisme à implémenter. Elle permet de représenter des données sous forme de diagrammes ou de créer des animations, voire même des jeux. La méthode centrale de la classe Canvas à implémenter est la méthode paint(Graphics g). L'objet graphique passé en argument à cette méthode permet les opérations graphiques, comme le tracé de lignes, de cercles, l'affichage de texte, le remplissage par une couleur, etc. La classe Image complète ces APIs de bas niveau pour l'affichage. L'image devra être dans un format reconnu par le mobile. MIDP reconnaît en standard le format PNG.

La persistance des données

Il est possible de stocker les données de 2 manières : soit en utilisant le RMS (Record Management Store), soit en utilisant le système de fichiers. Dans le premier cas (RMS), le package javax.microedition.rms livré avec MIDP 2.0 donne accès à une sorte de mini base de données. Chaque MIDlet aura accès à ses données et uniquement à ses données. Aucune autre application ne peut y avoir accès. Les jeux notamment stockent les " high scores " avec ce système. L'autre solution est d'utiliser un système de fichiers classique. Pour cela, il faut que le téléphone portable supporte l'option " JSR 75 FileConnection ". L'avantage de cette solution est que les données sont ensuite récupérables sur votre PC. Un des objectifs de l'application que je voulais créer, était de pouvoir récupérer les données saisies sur PC, afin de pouvoir les retrai-

ter via Excel par exemple… Le chemin complet d'un fichier est de la forme " file:///c:/nokia/ Images/Image001.jpg ". Le préfixe file:/// indique que l'on travaille sur un fichier, et "c:/" correspond à la mémoire interne. La carte mémoire correspond au lecteur "e:/ ". Ecrire un fichier, oui, mais dans quel dossier ? Il faut avoir la permission. La solution que j'ai choisie est d'utiliser le dossier photo de mon mobile afin d'y déposer mon fichier de données. Ce dossier est un dossier " public ".
Le code suivant permet de récupérer une connexion en lecture et écriture au fichier de données :

```
// Récupère le dossier des photos
String url = System.getProperty("fileconn.dir.photos");
// Rajoute le nom du fichier
url = url + "NokiaEssence.txt";
// Récupère la connexion
mainFileConnection = (FileConnection) Connector.open( url, Connector.
READ_WRITE );
```

La classe FileConnection permet l'accès au système de fichier d'une manière générale : elle permet de lister, supprimer, obtenir des informations sur les fichiers, etc. L'accès au fichier s'effectue à l'aide des flux : InputStream en lecture et OutputStream en écriture. Voici un exemple de code pour créer le fichier :

```
public void WriteToFileString(String aStringToWrite)
{
// Crée le fichier
mainFileConnection .create();
// Récupère le flux en écriture
OutputStream os = mainFileConnection .openOutputStream();
// Ecrit dans le fichier
os.write(aStringToWrite.getBytes());
// Vide le cache
os.flush();
// Ferme le fichier
os.close();
}
```

L'option JSR 75 ne se limite pas à l'accès au système de fichiers. Elle fournit une 2e API nommée PIM (Personal Information Management). Cette API permet l'accès aux données personnelles (Contacts/Liste des numéros de téléphone, Agenda/Calendrier, Liste des choses à faire) depuis la mémoire interne du téléphone ou la carte SIM.

API Multimédia

Si votre téléphone portable supporte la spécification JSR 135 Mobile Media API, vous pourrez goûter à la joie de piloter l'appareil photo ou caméra de votre mobile. 4 classes principales sont là pour utiliser l'API :
- Manager
- Player
- Control
- Datasource

La classe Manager contient uniquement des méthodes statiques. La méthode Manager.createPlayer crée un player avec un DataSource associé pour lui fournir les données. La source de données peut être construite à partir d'un flux (InputStream) ou d'un " URI-style locator ".

En voici quelques exemples :

URI-style locator	Source de données
file:///c:/nokia/Images/Image001.jpg	Un simple fichier
http://adresseweb.com/fichier.wav	Une adresse web
capture://image	Capturer une photo depuis l'appareil photo
capture://video	Capturer une vidéo
capture://audio	Enregistrer du son

Tableau 4

Voici un exemple de code pour afficher la capture en temps réel de ce qui est visualisé par l'objectif de l'appareil photo. Le rendu est affiché directement sur la Form.

```
Form f = new Form("JPP Form"); // Crée la form principale
Display.getDisplay(this).setCurrent(f); // Affiche la form
Player p = Manager.createPlayer("capture://video");
p.realize(); // Initialise le player
VideoControl vc = (VideoControl)p.getControl("VideoControl");
if (vc != null)
{
  Item it = (Item) vc.initDisplayMode (VideoControl.USE_GUI_PRIMITIVE, null);
  f.append(it); // Rajoute l'item à la form.
  p.start();    // Affiche la vidéo dans l'item.
}
```

La ligne de code suivante permet de prendre une photo :

```
videoControl.getSnapshot("encoding=bmp");
```

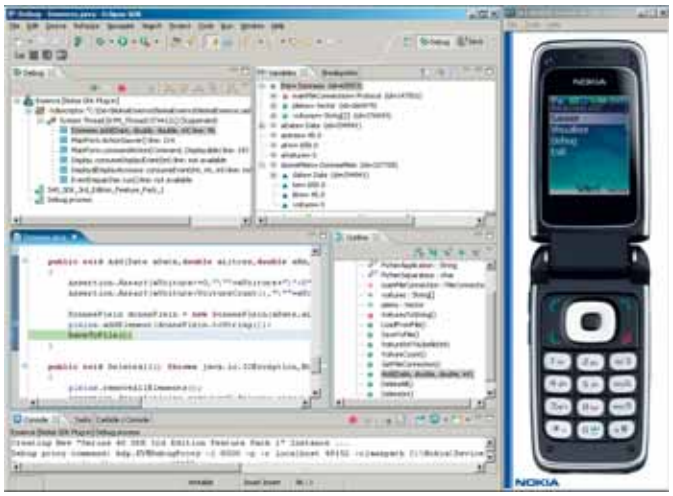
Cette architecture très ouverte permet aussi d'afficher des vidéos au format 3gp, de jouer des sons très facilement, etc. La méthode getControl de la classe Player renvoie une interface abstraite que plusieurs classes concrètes implémentent. Suivant les cas, le transtypage vous permettra de manipuler tel ou tel type de contrôle. Voici quelques exemples de classes concrètes :
- ToneControl
- VolumeControl
- VideoControl
- StopTimeControl
- RecordControl

Le développement de jeux

Nokia fournit avec son SDK tout un ensemble de fonctionnalités pour la création de jeux. Un tutorial installé avec Carbide.j permet de créer votre premier jeu sur mobile en quelques clics, tout en expliquant les bases. Carbide.j propose aussi un designer afin d'éditer vos propres cartes pour vos jeux 2D. Les nostalgiques se rappelleront le temps des sprites et défilement d'écrans. Mais le plus impressionnant reste les jeux 3D. L'option JSR 184 (M3G, Mobile 3D Graphics API) propose tout le nécessaire en la matière. Trois classes sont la clé de voûte de cette option :

Graphics3D	Contexte 3D graphique : effectue le rendu
Loader	Permet de charger les objets individuellement et les graphiques de la scène entière (Fichiers M3G et PNG)
World	Nœud racine de la scène. La scène est une arborescence d'objets

Tableau 5



Nokia a pour l'instant une longueur d'avance en matière de jeux vidéo pour mobile. Le " smartphone " N-Gage en est un bel exemple. Mais Nokia ne se limite pas à la production de terminaux : la société a lancé un SDK nommé SNAP (Scalable Network Application Package). Il s'agit d'une solution " clé en main " pour fournir une infrastructure de services pour la communauté de jeux en ligne multi joueurs.
SNAP permet d'établir une communication réseau entre le terminal et le serveur de jeu afin de rejoindre une communauté de jeu. Les fonctionnalités proposées sont l'authentification, la liste de contacts, l'échange de messages en cours de jeu, la création d'événements et enfin le classement des joueurs. Ce SDK particulier est conçu pour fonctionner avec les standards MIDP 2.0 et CLDC 1.0

Communications / Web services

Par définition, un téléphone est un objet communiquant Java, bien plus que vous ne pouvez l'imaginer : la variété des moyens de communication et des protocoles est impressionnante. Jugez-en par vous-même :

Type de communication	Url spécifique	Option
Infrarouge	comm:<port>	MIDP
Web	http://<adress>	MIDP
Web sécurisé	https://<adress>	MIDP
Datagrams	datagram://:<port>	MIDP
Socket	socket://<port>	MIDP
Ssl	ssl://<address>	MIDP
Sms	sms://	JSR 120
Mms	mms://	JSR 205
Bluetooth	btspp://<server address>	JSR 82

Tableau 6

Le package javax.microedition.io.Connector est le point d'entrée de l'utilisation de ces moyens de communication.
Reste un autre moyen de communication très populaire : les Web services. L'option JSR 172 permet de s'appuyer sur un serveur ou fournisseur de Web services déjà existant. Le but de ce package est de fournir un parsing XML, et une API standard pour accéder et consommer des Web services. Un des avantages de créer un Web service est de permettre à différents clients (application client riche, application web via un navigateur, ou application mobile) d'accéder au même service. Carbide.j facilite le travail en proposant un générateur de code afin d'accéder aux Web services. La logique " métier " est donc centralisée et réutilisable pour différents types de clients.
Une autre option, qui couplée avec les web services semble vraiment prometteuse est l'option JSR 179 (Géolocalisation). Aujourd'hui, cette option n'est pas encore très répandue. Elle fournit des informations comme le positionnement géographique (latitude, longitude et altitude), la vitesse du terminal mobile en déplacement, etc. Ces informations peuvent provenir de différentes sources : GPS, réseau cellulaire, ou pour des localisations de proximité Bluetooth positionning system par exemple. Cette option fournit aussi un ensemble de classes afin de se connecter à une base de données de " landmark ". Un landmark est une localisation physique connue qui est associée avec un nom représentant cette localisation. Les mobiles sont aussi équipés de navigateur web. La norme actuelle pour le développement de pages web pour ces navigateurs est XHTML MP (Mobile Profile). XHTML est très proche du HTML que l'on connaît. Il suffit d'être un peu plus rigoureux sur quelques points comme dans la fermeture des balises… Cependant, il faudra adapter les pages à une taille d'écran à laquelle le web ne nous a pas habitués. Cela réduit forcément le nombre d'informations affichées simultanément… Cependant beaucoup d'opérateurs font payer au prix fort l'accès à Internet via leur réseau. Mais à ne pas en douter, la demande croissante d'accès à Internet via ces terminaux sera forte. A quand verrons-nous la guerre commerciale chez les opérateurs mobiles pour nous proposer un accès Internet mobile illimité à un prix tout à fait compétitif ? L'exemple des " box " que nous connaissons aujourd'hui chez les opérateurs téléphoniques classiques pourrait donner quelques idées aux opérateurs mobiles…

Test et débogage

Après avoir installé Carbide.j et le SDK s40, rien n'est plus simple pour tracer votre code : posez vos points d'arrêt, régénérez votre package, et vous pouvez tester votre application en cliquant sur le bouton debug. Eclipse basculera automatiquement en perspective de débogage (Voir image IDEDebug). Si vous créez plusieurs projets mobiles, vous aurez intérêt à créer différentes configurations. Coté test, un framework de test unitaire (J2MEUnit) basé sur le célèbre JUnit et adapté au monde mobile offre l'environnement nécessaire pour capitaliser ses tests (http://j2meunit.sf.net).

Le déploiement

Le transfert de l'application peut s'effectuer sur le téléphone de différentes manières. Les moyens les plus courants sont le câble USB, l'infrarouge ou le bluetooth. Par câble USB, un logiciel comme Nokia Pc Suite (qui propose parmi ses nombreuses fonctionnalités d'installer une application par simple copie du fichier .jar) sera nécessaire. L'envoi du fichier par infrarouge installe automatiquement l'application sans avoir besoin d'un logiciel tiers.
Un autre moyen très répandu pour installer et diffuser des MIDlet est d'utiliser la technique nommée OTA (Over The Air) : le distributeur de votre application installe la MIDlet sur un serveur web et fournit un lien hypertexte. L'utilisateur active ce lien sur le téléphone portable qui va déclencher le téléchargement de la MIDlet. Le lien pointera comme l'indique l'exemple vers le fichier .jad créé lors de la génération du package par Carbide.j.

```
<A HREF="MaMIDlet.jad">Cliquez ici</A> pour installer l'application
```

L'un des gros avantages du langage Java est sa portabilité. La portabilité signifie que le programme fonctionnera sur une grande quantité de plates-formes différentes, sans nécessiter d'être recompilé. Même si on développe avec les outils fournis par Nokia, le fruit du développement peut tout à fait être utilisé sur d'autres marques de téléphones, sans recompilation. Il suffit que le téléphone supporte les spécifications qui ont été à la base du développement (MIDP, CLDC et JSR…) et le fichier .jar sera exécuté…

Sécurité

L'accès aux fichiers via l'API JSR 75 n'est pas autorisé par défaut. En effet, on ne plaisante pas avec la sécurité sur téléphone portable : une MIDlet non signée se verra par défaut refuser l'accès au système de fichier (cf. Tableau 7). L'utilisateur pourra paramétrer cela, mais il devra quand même donner son autorisation (" Ask every time ") à chaque ouverture de fichiers. Pour que votre application puisse se passer de l'autorisation de l'utilisateur, elle devra être signée.

Comment donner l'accès aux fichiers à une application non signée :

Dans la liste des applications, positionnez-vous sur NokiaEssence. Sélectionnez " Options ",

1. Sélectionnez " App. access ",
2. Sélectionnez " Data access ",
3. Sélectionnez " Add and edit data ",
4. Sélectionnez " Ask every time ",
5. Un message vous confirme votre choix.

Pour information dans l'écran n°5, les options (" Ask first time only " et " Always allowed " se dégrisent uniquement si l'application est signée)

Tableau 7

De plus il faudra donner les permissions suivantes à la MIDlet :

```
javax.microedition.io.Connector.file.read
javax.microedition.io.Connector.file.write
```

Cela se paramètre lors de la génération du package avec Carbide.j

Signature des applications

Signer une application consiste à envoyer des clés générées depuis Carbide.j à une autorité de certification. Après avoir décliné votre identité et donné quelques éléments d'identification personnelle, cette autorité vous renverra un certificat numérique qu'il faudra importer grâce au plug-in Carbide.j. L'autorité de certification vous facturera cela. Comptez environ 199$ pour 1 an. En effet, les certificats ont une durée de validité. De plus votre téléphone doit aussi posséder un " certificat d'autorité " de l'autorité qui vous aura délivré le certificat. Mon choix a donc été de ne pas signer mon application (question de coût), et de laisser l'utilisateur autoriser l'application à accéder aux fichiers. Cependant, à chaque ouverture de fichiers, cette autorisation est demandée à l'utilisateur. Cela m'a contraint à stocker toutes mes données dans un unique fichier afin qu'une seule autorisation soit demandée durant l'exécution de l'application.

Personnalisez votre téléphone avec : Nokia Theme Studio

La personnalisation des téléphones mobiles est rendue possible grâce à l'outil Nokia Theme Studio (Voir image NokiaThemeStudio). Au départ, vous pouvez personnaliser vos sonneries, écran de veille ou fond d'écran. Mais pas seulement : vous pouvez personnaliser l'interface graphique pour avoir un look complètement à votre goût. La personnalisation va très loin, tous les composants graphiques peuvent être redessinés. Le résultat sera sauvegardé dans un fichier à l'extension .nth qui pourra être appliqué ensuite à votre mobile. Cependant, même si c'est de l'art, pensez à ce que votre thème soit utilisable. Les textes et les icônes doivent être suffisamment contrastés afin que l'on puisse continuer à utiliser le téléphone aisément.

Widget sur mobile

Les Widgets sont de petits programmes permettant d'afficher des informations (météo locale, brèves sportives, économiques…) en provenance d'Internet. Ce sont en fait des lecteurs de flux XML d'information. La société Widset (http://www.widsets.com) fournit une Midlet à installer sur votre mobile. Après avoir créé un compte sur le site, vous pourrez sélectionner les thèmes qui vous intéressent. Vous pouvez aussi créer vos propres Widgets. Tout cela gratuitement. Cependant, pensez que votre opérateur vous facturera le trafic de données généré…

Bilan & difficultés rencontrées

L'utilisateur doit confirmer à chaque lancement l'autorisation d'accès au système de fichiers. Ce coté restrictif de la sécurité a été une contrainte. Une incompatibilité entre Eclipse 3.2 et Carbide.j 1.5, m'a empêché d'utiliser le débuggeur d'Eclipse. La solution a été de basculer sur Eclipse 3.1. Le codage a été assez rapide, et les forums de discussion Nokia (http://www.forum.nokia.com) permettent de poser les questions bloquantes. Ces mêmes forums constituent aussi une base de connaissance très utile. Le résultat de mon développement est visible sur les captures d'écrans (Voir image NokiaEssenceEcrans). Les sources, l'application, la documentation utilisateur sont disponibles sur le cdrom ou sur le site http://jp.planas.free.fr/nokiaessence où vous retrouverez les mises à jour.

Conclusion

Dans un environnement où l'ordinateur n'est pas toujours présent, il est envisageable de se tourner vers ce genre d'application à moindre coût. L'interface d'entrée privilégie essentiellement la saisie numérique. C'est ainsi que l'adaptation à ce type de plate-forme nécessite de repenser la manière d'appréhender les applications : il peut parfois être plus pratique de substituer la saisie par la prise d'une photo qui servira d'identifiant, quitte à compléter les informations ultérieurement sur ordinateur. En conclusion, les outils (IDE, sdk, plug-in, outils de design, etc.…), les normes, le matériel de formation, les documentations sont là. Nokia propose de manière claire et efficace tout le nécessaire pour vous lancer sur le sujet. Ce type de développement est beaucoup plus simple qu'il n'y paraît. Les projets sont de petite taille, donc rapides à réaliser. Les possibilités sont vastes. Pourquoi ne pas vous lancer cet été dans la création d'une petite application ? C'est fun, ludique et rapide.
A vous de trouver la bonne idée…

■ **Jean-Pierre Planas** *(Sybaris@Club-Internet.Fr)*
Ingénieur en développement chez VcsTimeless (www.vcstimeless.fr), développe principalement en Delphi.

Rédaction interactive
www.programmez.com

2e partie

Plate-forme Mozilla : XulRunner & XPCom

Nous allons aujourd'hui découvrir comment faire une application standalone, ne dépendant pas de Firefox. Nous verrons aussi le système de composants et d'interfaces pour comprendre comment s'orchestrent les différents modules fonctionnels de Gecko et de la plate-forme Mozilla.

XulRunner

Nous allons développer un mini navigateur Internet, TinyBrowser. Cet outil sera basé sur une balise XUL qui nous mâchera une partie du travail <browser/>. Nous voulons une application standalone. Nous allons donc utiliser une version de Gecko extraite de Firefox: XulRunner. XulRunner est une application qui se charge d'exécuter un paquet chrome. On peut faire une analogie rapide entre Java et XulRunner. Java est une machine virtuelle qui va lancer des applications codées en Java, XulRunner est un programme qui va interpréter nos paquets chrome, codés en XML et en Javascript. Il s'utilise ainsi :

```
$ xulrunner application.ini
```

où 'application.ini' est un fichier décrivant l'application. Firefox 2 (ainsi que Thunderbird 2) est basé sur Gecko 1.8.1. La version de Gecko de Firefox 1.5 est 1.8.0. La version de XulRunner suit la version de Gecko. La dernière version officielle de XulRunner est la version 1.8.0.4, donc équivalente au moteur de Firefox 1.5. Mais vous trouverez sur le site de Mozilla des versions bien plus récentes. Je vous conseille de travailler avec la 1.8.1 qui apporte plus de fonctionnalités.

Hello World !

Commençons par le traditionnel 'Hello World'. Nous allons retrouver notre arborescence classique :

```
|- application.ini
|- chrome
|   |- content
|   |   `- helloworld.xul
|   `- helloworld.manifest
`- defaults
    `- preferences
        `- pref_helloworld.js
```

On découvre 2 fichiers que l'on ne connaissait pas. 'pref_helloworld.js' est un fichier de préférences, comme le fichier 'prefs.js' dans votre profil Firefox. C'est ce fichier qui définit les préférences par défaut de votre application. Ici, nous n'en préciserons qu'une seule :

```
pref_helloworld.js:

pref("toolkit.defaultChromeURI",  "chrome://helloworld/content/hello
world.xul");
```

C'est une préférence qui permet à XulRunner de savoir quel fichier Xul lancer au démarrage.
Le fichier 'application.ini' est la " carte d'identité " de notre application :

```
application.ini:

[App]
;nom et version de l'application
Name=helloworld
Version=0.1
;identifiant unique de la compilation de l'application, en général, la date de
construction de l'application
BuildID=20070606
;l'identifiant unique de l'application
ID=helloworld@www.xulfr.org
;Le nom du fournisseur de l'application
Vendor=xulfr.org

[Gecko]

;les versions de XulRunner compatibles avec l'application
MinVersion=1.8
MaxVersion=1.9.*
```

Notre fichier 'manifest' sera très simple :

```
helloworld.manifest:
content helloworld content/
```

Ainsi que notre fichier XUL :

```
helloworld.xul:
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window xmlns="http://www.mozilla.org/keymaster/gatekeeper/there
.is.only.xul" title="Hello world !">
 <button label="Hello..." oncommand="alert('... world !')"/>
</window>
```

Tout est prêt. Si vous avez correctement suivi l'arborescence, un simple

```
$ xulrunner application.ini
```

vous lancera notre première application XulRunner. Vous pouvez aussi lancer XulRunner avec l'option '-jsconsole' :

```
$ xulrunner application.ini -jsconsole
```

qui vous affichera en plus une console javascript. Vous pouvez aussi ajouter à votre fichier de préférences les lignes :

```
pref("javascript.options.strict", true);
pref("javascript.options.showInConsole", true);
```

qui permettront à la console d'être plus verbeuse.

Les profils

Le but du profil est de stocker pour une application les préférences spécifiques à un utilisateur. Par exemple, vous trouverez le profil de Firefox dans :

```
(Linux) ~/.mozilla/firefox/<NomDuProfil>/
(Windows) C:\Documents and Settings\<NomDeL'User>\Application Data\Mozilla\Firefox\Profiles\<NomDuProfil>\
```

Un utilisateur peut avoir plusieurs profils. Une application XulRunner a aussi son profil :

```
(Linux) ~/.<VendorName>/<ApplicationName>/<NomDuProfil>/
```

TinyBrowser

Nous voulons coder un simple navigateur qui propose les fonctionnalités de base: pouvoir charger une URL, naviguer dans l'historique (précédent, suivant) et gestion de la page d'accueil. Voilà à quoi ressemblera l'application :

On remarque 8 éléments :

- les boutons précédent/suivant
- les boutons " mettre en tant que page d'accueil " et " aller à la page d'accueil "
- le <textbox/> qui accueillera l'URL à charger
- le bouton pour charger l'URL dans le navigateur
- le navigateur, soit la balise <browser/>
- le titre de la page en cours au-dessus du navigateur

Nous n'allons pas dérouler le code source complet de l'application. Basez vous sur ce que nous avons vu dans l'article précédent et le helloworld que nous venons de coder. Nous allons nous focaliser sur le code XUL et Javascript. Voyons notre code XUL :

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<?xml-stylesheet href="chrome://tinybrowser/content/tinybrowser.css" type="text/css"?>
<window xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul" title="TinyBrowser" width="800" height="600" persist="width,height,screenX,screenY" onload="gInit()">
[...]
```

On remarque ici un nouvel attribut, 'persist', qui permet de préciser ici qu'entre différentes sessions nous voulons que Gecko conserve en mémoire où était placée l'application et quelle était sa taille dans la session précédente.

```
[...]
  <commandset>
      <command id="cmd_goBack" oncommand="gGoBack()"/>
      <command id="cmd_goForward" oncommand="gGoForward()"/>
      <command id="cmd_goHome" oncommand="gGoHome()"/>
      <command id="cmd_setHomePage" oncommand="gSetHomePage()"/>
      <command id="cmd_loadURI" oncommand="gLoadURI()"/>
  </commandset>
[...]
```

Voici un nouvel élément que nous n'avions absolument pas abordé précédemment. Ce n'est pas un élément qui a une représentation graphique. Une balise <command/> est associée à des attributs liés à des actions utilisateur ('oncommand', 'onselect', 'onclick', etc.). Cela permet de centraliser la définition d'actions. Par exemple, on va voir immédiatement après qu'il y a 2 manières de déclencher un retour en arrière dans l'historique. On va raccrocher les déclencheurs de cette action à la balise <command id="cmd_goBack"/>.

```
[...]
  <keyset>
      <key modifiers="alt" keycode="VK_LEFT" command="cmd_goBack"/>
      <key modifiers="alt" keycode="VK_RIGHT" command="cmd_goForward"/>
  </keyset>
[...]
```

Voici la définition des deux raccourcis clavier : alt-gauche, aller à la page précédente, alt-droite, aller à la page suivante. Le code est assez explicite. On remarque comment on relie les balises <key/> qui peuvent déclencher des actions à une balise <command/> : via un attribut command. Cela fonctionne de la même manière pour un bouton :

```
[...]
  <script type="text/javascript" src="tinybrowser.js" />
  <toolbar>
      <toolbarbutton id="tb-goback" command="cmd_goBack"/>
      <toolbarbutton id="tb-goforward" command="cmd_goForward"/>
      <toolbarbutton id="tb-sethomepage" command="cmd_setHomePage"/>
      <toolbarbutton id="tb-goHome" command="cmd_goHome"/>
      <textbox flex="1" id="tb-URI"/>
      <toolbarbutton id="tb-loadURI" command="cmd_loadURI"/>
  </toolbar>
[...]
```

Le code ici parle de lui même.

```
[...]
  <groupbox flex="1">
      <caption id="title"/>
      <browser id="tbrowser" flex="1" type="content-primary"/>
  </groupbox>
</window>
```

Et pour finir, nous embarquons notre navigateur <browser/> dans une

boîte <groupbox/> dont la légende <caption/> contiendra le titre de la page en cours.
Dans notre fichier Javascript, implémentons les méthodes de base :

```
tinybrowser.js:
var gBrowser;
function gInit() {
   gBrowser = document.getElementById("tbrowser");
}
function gLoadURI() {
   var uri = document.getElementById("tb-URI").value;
   gBrowser.loadURI(uri);
}
function gGoBack() {
   if (gBrowser.canGoBack)
      gBrowser.goBack();
}
function gGoForward() {
   if (gBrowser.canGoForward)
      gBrowser.goForward();
}
```

L'API de la balise <browser/> se trouve sur xulplanet.com.

XPCom

Que pouvez vous faire en Javascript ? Pas grand-chose de plus que dans une simple page Web, en gros : DOM et XMLHttpRequest (WebServices). Il existe au sein de Mozilla un ensemble de modules, fournissant chacun des fonctionnalités. Par exemple, accéder au système de fichiers, manipuler (créer) un protocole particulier, jouer avec les préférences de votre application, ouvrir un socket TCP/IP, etc. Et bien, ces fonctionnalités sont accessibles via ce que l'on appelle des composants XPCom. Vous pouvez en voir la liste exhaustive ici. Ces composants XPCom sont des modules que vous pouvez instancier en Javascript. Une instance d'un composant XPCom est manipulable à travers une certaine API, que l'on appelle interface. Pour l'exemple, nous allons dévier un instant de notre TinyBrowser : La plate-forme Mozilla met à disposition un composant XPCom permettant de lire des fichiers zippés. Le nom de ce composant est :

```
@mozilla.org/libjar/zip-reader;1
```

Le nom d'un composant est en général: @mozilla.org/xxx/xxx;1
Il implémente les interfaces (une interface est une API) :

```
nsISupports
nsIZipReader
```

Le nom d'une interface est en général : nsIXxxxx .
C'est-à-dire que ce composant implémente les fonctions décrites dans nsISupports et dans nsIZipReader. Tout composant XPCom implémente l'interface nsISupports. C'est l'interface mère de toutes interfaces. Dans le cas où nous ne sommes qu'en Javascript, nsISupports n'expose qu'une seule méthode :

```
QueryInterface(uneInterface);
```

On voit une instance d'un composant XPCom à travers une interface. La méthode QueryInterface permet de basculer entre les différentes interfaces d'un composant. Voyez QueryInterface comme un cast dynamique.
On crée une instance d'un composant XPCom ainsi :

```
var unZipReader = Components.classes["@mozilla.org/libjar/zip-reader;1"].createInstance(Components.interfaces.nsIZipReader);
```

'Components.classes' est un tableau de toutes les classes XPCom existantes. Nous demandons donc la classe '@mozilla.org/libjar/zip-reader;1', et nous réclamons ensuite une instance de type nsIZipReader. 'Components.interfaces' est une structure contenant toutes les interfaces.
Automatiquement, l'objet 'unZipReader' est vu comme 'nsIZipReader'. Mais on aurait aussi pu l'instancier ainsi :

```
var unZipReader = Components.classes["@mozilla.org/libjar/zip-reader;1"].createInstance(Components.interfaces.nsISupports);
```

et ensuite charger l'API sur laquelle on travaille :

```
unZipReader.QueryInterface(Components.interfaces.nsIZipReader);
```

On remarque dans l'API de 'nsIZipReader' que l'on doit l'initialiser avec un nsIFile. De la même manière, vous pouvez créer un objet de type nsIFile à travers le composant '@mozilla.org/file/local;1', qui, lui-même, peut être initialisé à partir d'un chemin sous forme d'une chaîne de caractères, objet nativement géré en Javascript.
Ceci est une utilisation classique des composants XPCom de Mozilla. Vous trouverez la liste de composants XPCom et les interfaces qu'ils implémentent sur XulPlanet. À partir de là, vous savez comment les instancier et comment les utiliser.
Parlons d'un type particulier de composants XPCom : les services. Les services ont la particularité de partager la même instance. Un service est un singleton. Ainsi, si vous réclamez le service IOService par exemple (un module qui permet de travailler entre autres avec les URL), vous obtenez toujours la même instance de ce composant. D'ailleurs on n'instancie pas exactement de la même manière un service XPCom. Au lieu d'utiliser createInstance, on utilisera getService.

Gestion des préférences

Nous avons prévu de gérer une page d'accueil. Nous allons, comme le fait Firefox, stocker l'URL de notre page d'accueil dans le *préférences :*

```
pref_tinybrowser.js:
pref("tinybrowser.homepage", "http://xulfr.org/");
```

Pour lire et modifier les préférences, nous avons besoin d'un composant XPCom, 'preferences-service'. On voit qu'il implémente 'nsIPrefBranch'. Comme son nom l'indique, c'est un service. On le récupère ainsi :

```
tinybrowser.js:
var gPrefService;
function gInit() {
   gBrowser = document.getElementById("tbrowser");
   gPrefService = Components.classes["@mozilla.org/preferences-service;1"].
                  getService(Components.interfaces.nsIPrefBranch);
   gGoHome();
}
```

On voit dans l'API 'nsIPrefBranch' comment manipuler les préférences.

Nous pouvons donc implémenter les méthodes de gestion de la page d'accueil :

```
tinybrowser.js:
function gSetHomePage() {
   gPrefService.setCharPref("tinybrowser.homepage",  gBrowser.current
URI.spec);
//Faites le test. Modifiez la page d'accueil. Fermez TinyBrowser, et allez
vérifier dans votre profil. La page d'accueil est bien enregistrée. Attention
! Le fichier pref de votre application n'est pas modifié.
}
function gGoHome() {
   var homePageURI = gPrefService.getCharPref("tinybrowser.homepage");
//On charge l'URL sauvée dans les préférences dans le browser
   gBrowser.loadURI(homePageURI);
}
```

Voilà. Nous gérons maintenant la page d'accueil. Dernière étape, mettre à jour l'URL et le titre de la page actuelle dans la barre d'URL et dans le <caption/>.

WebProgressListener

Si on regarde de plus près l'API de la balise <browser/> on remarque une méthode qui sonne bien : 'addProgressListener'. D'après la documentation, il faut passer en paramètre à cette méthode un objet implémentant 'nsIWebProgressListener' et 'nsISupportsWeakReference'. On remarque que deux méthodes de 'nsIWebProgressListener' peuvent nous intéresser :

- onLocationChange: qui mettra à jour l'URL dans notre <textbox/>
- onStateChange: qui nous dira quand la page aura fini de charger. On pourra ainsi récupérer le titre de la page Web pour le mettre dans le caption.

Ces méthodes sont donc apparemment appelées par le browser aux différents stades du chargement de la page Web. Ne cherchons donc pas un objet XPCom implémentant ces interfaces et donc ces méthodes, mais codons nous même un objet Javascript implémentant ces interfaces ET nsISupports.

Seules ces méthodes nous intéressent. Nous n'implémenterons donc pas les autres. Ce qui nous donne :

```
function gInit() {
 [...]
   gBrowser.addProgressListener(gListener);
   gGoHome();
}
[...]
const gListener = {
//N'oublions pas qu'une instance d'objet XPCom doit implémenter nsI
Supports, donc, en Javascript, implémenter QueryInterface
     QueryInterface: function (aIID) {
          if (aIID.equals(Components.interfaces.nsIWebProgressListener) ||
          aIID.equals(Components.interfaces.nsISupportsWeakReference) ||
          aIID.equals(Components.interfaces.nsISupports)) {
               return this;
          }
          throw Components.results.NS_ERROR_NO_INTERFACE;
     },
     onLocationChange: function ( webProgress, request, location) {
//Si l'URL change, on met à jour la <textbox/>
          document.getElementById("tb-URI").value = location.spec;
     },
     onStateChange: function (webProgress, request, stateFlags, status) {
//Si l'état est STATE_STOP (cf. [7]), la page est chargée. Donc on peut y
récupérer le titre et mettre à jour le <caption/>
          if (stateFlags & Components.interfaces.nsIWebProgressListener
.STATE_STOP) {
               document.getElementById("title").setAttribute("label",
gBrowser.contentTitle);
          }
     },
//Le reste ne nous intéresse pas.
     onProgressChange: function () {return null;},
     onSecurityChange: function () {return null;},
     onStatusChange: function () {return null;},
     GetWeakReference: function () {return null;}
};
```

Notre code est complet. Le <browser/> appellera nos callbacks que l'on a abonnés à travers un objet de type 'nsIWebProgressListener'.

Un peu de folie

Cette petite expérience illustre la base de l'utilisation de XulRunner ainsi que des composants XPCom. XPCom va bien plus loin. Vous pouvez créer votre propre composant et vos interfaces. Vous pouvez implémenter votre protocole (foobar://), implémenter votre système d'autocomplétion dans les barres d'URL, etc. En implémentant les API existantes vous rendez vos composants compatibles avec les composants de Mozilla et les intégrez de manière plus pertinente dans Firefox ou XulRunner.

La complexité (relative) de l'utilisation des composants XPCom vient d'un point que nous n'avons pas abordé : on peut piloter dans un code chrome (donc en Javascript) n'importe quel composant XPCom, qu'il soit codé lui-même codé en Javascript, en C++ ou même en Python. Si vous codez aussi en C++ ou en Python, vous pouvez rajouter des compétences complètement exotiques au framework Mozilla en encapsulant des bibliothèques tierces dans des composants XPCom. Par exemple une bibliothèque permettant de vous connecter à des périphériques extérieurs (contrôler votre robot en Javascript :D), ou encore une bibliothèque de traitement audio pour contrôler vos entrées/sorties et vos différents effets sonores en Javascript, créer des boîtes cliquables en SVG à travers un XBL et créer ainsi un langage XML pour gérer vos compositions sonores ! Allez ... soyons fou, vous pouvez aussi envisager de coder un composant en Javascript qui transforme un WebService de gestion de fichier (un GED) en un nsIFile, et ainsi vous pourriez exploiter vos fichiers distants avec toute la mécanique liée aux nsIFile dans Mozilla, et peut être créer votre propre protocole, vos propres streams que vous passerez dans les tuyaux existants de Gecko.... quel programme !

Site du projet : http://xulfr.org/wiki/TinyBrowser

Paul Rouget
Vice-président *de l'association Xulfr. Travaille depuis plus de 3 ans autour des technologies Mozilla.*
paul.rouget@gmail.com - http://blog.mozbox.org/

Recevez la Newsletter www.programmez.com/inscription.php

Struts, JSF : lequel choisir ?

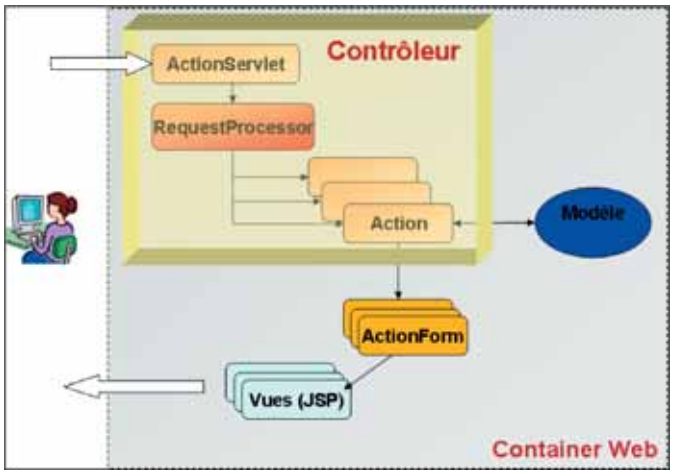
Struts est un framework open source. Acteur majeur de la sphère Java depuis 2001, c'est aujourd'hui encore le framework web java le plus utilisé. Il répond aux besoins de beaucoup d'applications, mais trouve ses limites dans la gestion d'écrans complexes ou le développement d'applications fortement événementielles. JSF (Java Server Faces) est une norme JEE (1) depuis la version 1.5. Elle arrive directement en concurrence avec Struts comme technologie Web pour Java. Comme JSF fait partie de la norme, les développeurs Java sont encouragés à l'utiliser.

Il y a 2 ans, tout portait à croire que Struts serait remplacé par JSF sur bon nombre de projets. Qu'en est-il aujourd'hui? JSF a-t-il détrôné Struts ? L'approche JSF est-elle vraiment révolutionnaire par rapport à Struts ? Et comment ce dernier a-t-il évolué ?

Architectures comparées

Vous avez dit Struts ?

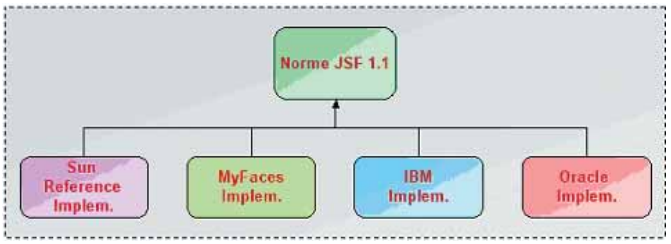
Struts est basé sur une structuration de la couche présentation en Servlet-JSP tout en respectant le modèle MVC 2 (Model-Vue-Contrôleur 2). Il a été initialement développé par Craig R. McClanahan (également à l'origine de Tomcat 4) puis offert à la fondation Apache en 2000. En décembre 2005, Struts a été scindé en deux projets distincts, le projet

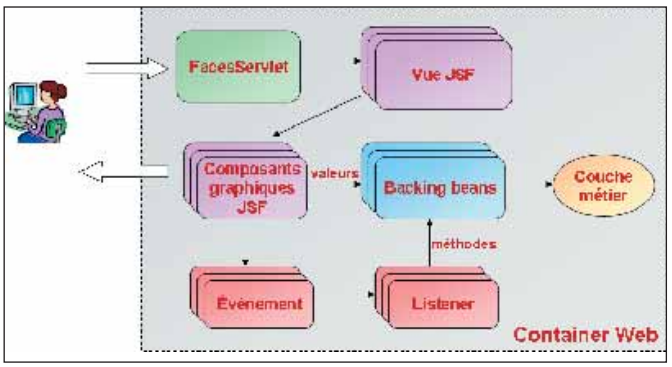


Architecture de Struts

initial de Struts est resté sous le nom de Struts 1.x. Un second projet, Struts 2, correspond initialement à WebWork 2 et émane de la volonté de rapprochement des communautés Struts et Webwork. Nous nous focaliserons dans cet article sur la première version, celle la plus communément utilisée aujourd'hui.

Son approche peut-être qualifiée de " contrôleur dur " et " request-based ", en ce sens que la partie contrôleur du modèle MVC2 est au cœur du système de traitement de la requête. Un contrôleur principal, implémenté par la servlet Struts ActionServlet, est ainsi le point d'entrée de toute requête HTTP constituée d'url se terminant par une extension spécifique (par exemple *.do, à paramétrer dans le fichier web.xml). La vue correspond aux JSP et le modèle est représenté par les données et règles métier à implémenter.





Architecture de JSF

Le contrôleur frontal Struts est facilement configurable (la description de la configuration étant déclarative décrite en XML : fichiers struts-config.xml et chain-config.xml). Il réalise des traitements communs, par exemple la gestion des exceptions ou la gestion des messages, et délègue la partie variable à des sous-contrôleurs, les Actions.

Les formulaires web sont mappés vers des beans dédiés (les ActionsForms) manipulables dans les Actions. Ce modèle est à l'opposé d'une approche orientée composants, comme celle de la spécification JSF où les composants graphiques gèrent les données propres au formulaire. Avec Struts, ces données contenues dans l'ActionForm sont manipulées par l'Action, et affichées dans la vue. Si bien qu'avec Struts, pour une fonctionnalité donnée, au moins quatre éléments sont à développer et à maintenir : l'Action, l'ActionForm, la Jsp et un fichier de configuration XML (<action-mappings> du struts-config.xml). Et ce n'est que le minimum requis, en effet si vous voulez gérer la validation, il vous faudra alimenter un autre fichier de configuration.

Java Server Faces

A la différence de Struts, JSF est une norme disposant de plusieurs implémentations possibles (2).

(1) Anciennement appelé J2EE - (2) A partir de JEE 1.5, chaque serveur d'applications devra fournir son implémentation JSF. En J2EE 1.4, chaque projet peut choisir son implémentation JSF (Sun Reference Implementation, MyFaces…).

JSF émane d'une Java Specification Request 127 dont les premiers travaux débutèrent en 2001 sous le lead de Craig Mc Clanahan (cité plus haut puisque à l'origine de Struts). Aujourd'hui, la spécification JSF 1.2 fait partie intégrante de JEE et il existe un bon nombre d'implémentations JSF (Myfaces de la fondation Apache, ADF d'Oracle, implémentation de référence de Sun, etc.). L'objectif de JSF est de fournir un modèle événementiel indépendant du protocole et en privilégiant une approche par composants graphiques.

Comme Struts, JSF s'appuie sur une architecture de type MVC2. Le contrôleur frontal JSF, la Faces Servlet permet également de gérer un certain nombre de services communs comme la navigation et la gestion des messages configurables de manière déclarative via un fichier XML (faces-config.xml). Les vues peuvent être développées en utilisant des taglibs dans des JSP mais ce n'est pas une obligation, elles peuvent également être décrites beaucoup plus simplement en XHTML grâce aux " facelets " (voir section détaillée dans la suite de l'article). La singularité de JSF vis-à-vis des vues est qu'elles sont représentées par un arbre de composants graphiques (les UIComponents).

En JSF, certaines classes sont responsables du rendu des composants graphiques : ce sont les " renderers ". Différents renderKit sont disponibles pour chaque composant graphique (RenderKits HTML, WML, etc.)

Les "mini-contrôleurs" JSF, appelés backing-beans, jouent un rôle particulier entre le contrôleur frontal et les vues. Ce sont des classes Java qui disposent de références sur les composants graphiques représentant la page. A partir d'un backing-bean, il est par exemple possible de désactiver l'affichage d'un champ de formulaire, de griser certains boutons...

Les backing-beans JSF sont de simples classes Java (pas de contraintes d'héritage ou d'implémentation d'interface).

Critères de comparaison

Montée en compétence

Pour une application simple, Struts est plus facile à appréhender. Le framework est bien documenté, qui propose une place pour chaque chose (contrôles de validation, méthodes actions etc.). A l'inverse, JSF n'est pas directif. Par exemple, en JSF les contrôles de validation peuvent se situer dans la page JSP, dans une classe Java dédiée à la validation, dans une classe Java dédiée au formulaire en cours etc. JSF propose beaucoup de possibilités, c'est à chacun de les étudier et de choisir la meilleure. De plus, JSF reste moins documenté que Struts. Pour une application d'ergonomie simple, il est donc plus facile de démarrer sur Struts.

Pour une application d'ergonomie complexe nécessitant beaucoup de JavaScript, JSF est bien plus facile à appréhender. Le développeur n'aura pas à écrire lui-même le JavaScript, mais à piocher parmi la pléthore de composants existants (Apache Tomahawk, Oracle ADF Faces, Jenia...(3). En comparaison, il est très fastidieux d'utiliser Struts pour gérer les interfaces riches.

Gestion des messages et Gestion de l'Internationalisation

La gestion des messages et de l'internationalisation est très semblable entre Struts et JSF. Il est possible de configurer des Resource bundles dans les fichiers de configuration des deux frameworks (balise <message-resources... /> pour le fichier struts-config.xml ou balise <message-bundle.../ > pour le fichier faces-config.xml)

Des Taglibs peuvent être utilisées pour afficher des messages du bundle. Avec Struts :

```
<bean:message key="label_login " />
```

Avec Myfaces:

```
<h:outputText value="#{messages.label_login}"  />
```

Validation des données

En Struts, il existe 2 façons de gérer la validation des formulaires : côté Java dans les ActionForms (en définissant une méthode validate()), ou par l'intégration du projet Apache commons Validator (configuration Xml de la validation, qui permet également de générer des contrôles de validation JavaScript).

```
<formset>
  <form name="clientForm">
    <field property="firstName" depends="maxlength">
    <var>
          <var-name>maxlength</var-name>
          <var-value>7</var-value>
    </var>
            <arg0 key="client.form.firstName"/>
    </field>
</formset>
```

exemple de validation Struts avec le Validator

JSF propose plusieurs moyens d'effectuer la validation côté Java (dans le code Java directement ou dans les pages JSP pour une validation plus basique). La norme JSF ne propose pas de génération de JavaScript pour les contrôles de validation côté client. Cette fonctionnalité peut être ajoutée par l'utilisation du projet Apache Shale (projet fournissant de nombreuses extensions pour JSF).

Gestion de la sécurité (autorisations)

La déclaration des utilisateurs et de leurs profils est effectuée dans la configuration du serveur d'application. En revanche, l'association entre un profil et une ressource donnée est effectuée au sein de l'application (ex : il faut avoir le profil "admin" pour accéder à "/bankonet/modifierCompte").

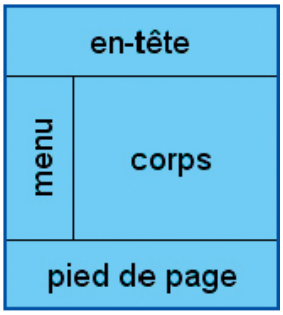
Le tableau ci-dessous résume les différences de gestion des habilitations entre Struts et JSF :

	Restriction d'accès à certaines URLs	Affichage conditionnel d'une portion de page JSP, selon les habilitations de l'utilisateur en cours.	Restriction d'accès à une classe Java ou à certaines de ses méthodes, selon les habilitations de l'utilisateur en cours.
JSF	Oui (norme JEE)	Non	Non
Struts	Oui (norme JEE)	Oui (tag logic:present)	Oui (attribut "role" dans les balises <action> du struts-config)

En résumé, Struts propose une gestion intégrée de la sécurité. JSF ne propose aucune facilité pour la gestion des autorisations. En JSF, il est

(3) Toutes les bibliothèques de composants citées sont gratuites et open-source.

nécessaire de mettre en place un mécanisme "maison" pour la gestion des autorisations (4).



Gestion des pages composites (Tiles vs Facelets)

Les pages composites permettent d'externaliser la structure des pages JSP (en-tête, pied-de-page...). Elles facilitent grandement les changements de charte graphique.

Avec Struts, il est possible d'utiliser la librairie Tiles pour définir des modèles de pages composites. Il s'agit ainsi d'une solution évitant une inclusion manuelle des jsp. Néanmoins elle ne dédouane pas le développeur d'utiliser des taglibs JSP spécifiques.

Avec JSF, il est possible d'utiliser les Facelets. Les Facelets offrent un système de templating beaucoup plus puissant, basé sur des définitions XHTML. Il suffit de déclarer le namespace xmlns:ui=http://java.sun.com/jsf/facelet ou de définir ses propres espaces de nommage, pour qu'ainsi aucune taglib JSP ne soit nécessaire. L'utilisation de taglib est toujours problématique lorsque l'on veut utiliser des environnements de développement avec des fonctionnalités WYSIWYG. Facelets tire son inspiration dans le modèle de template de Tapestry, l'objectif étant de permettre une intégration complète avec JSF en évitant le développement fastideux de JSP.

Voici un exemple de template layout.xhtml :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:ui="http://java.
sun.com/jsf/facelets">
<head>
   <title><ui:insert name="title"></ui:insert></title>
</head>

<body>
   <div>
        <ui:insert name="login">
        </ui:insert>
   </div>
</body>
</html>
```

La définition de page se construit comme suit :

```
<html …>
<ui:composition template="/layout.xhtml">
<ui:define name="title">
   Exemple de page de login avec facelets
 </ui:define>

<ui:define name="login">
<f:loadBundle basename="com.sqli.login" var="message" />
<h:form id="loginform">
<h:panelGrid>
<h:outputText value="#{ message.username}" />
<h:inputText value="#{LoginBean.username}" />
<h:outputText value="#{ message.password}" />
<h:inputSecret value="#{LoginBean.password}" />
<h:commandButton action="#{LoginBean.login}" value="Login" />
</h:panelGrid>
</h:form>
</ui:define>
</ui:composition>
</html>
```

Parmi les caractéristiques de templating avancées, Facelets permet d'utiliser un attribut particulier au sein d'un élément HTML standard, il s'agit de l'attribut jsfc (JSF Compilation), ressemblant au tag jwcid utilisés dans Tapestry. Voici un exemple où jsfc est ainsi utilisé pour informer le compilateur de créer un composant de type h:inputText dans la vue :

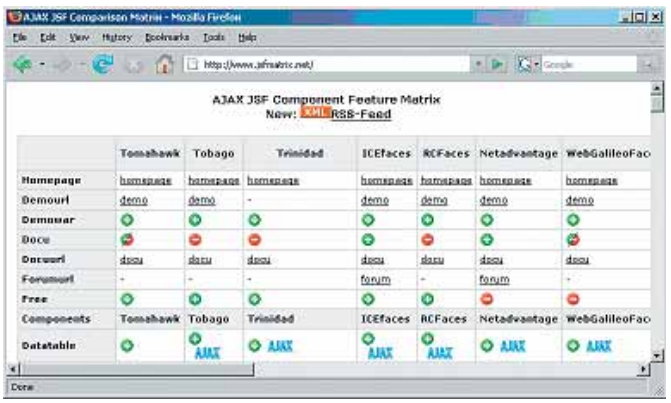
```
<input id="username" type="text" jsfc="h:inputText" value="#{Login
Bean.username}" />.
```

Cette fonctionnalité permet ainsi le développement de pages web par n'importe quel outil de design web comme DreamWeaver.

Gestion d'interfaces riches

L'approche par composants graphiques est le point de démarcation de JSF par rapport à Struts. Les librairies open source disponibles sur le site de Myfaces, comme Tomahawk et Tobago, sont très complètes. Les premières versions présentaient quelques problèmes de stabilité et d'intégration, mais les dernières sont plus robustes. Vous pouvez ainsi intégrer facilement un calendrier, des menus, des onglets sans effort à l'aide de ces composants existants. Dans les librairies les plus complètes, citons Trinidad et IceFaces. La librairie de composants Trinidad est un don d'Oracle (ADF Faces) à la fondation Apache. Cette librairie propose plusieurs dizaines de composants, certains intégrant notamment des fonctionnalités Ajax.

ICEsoft a sorti la version 1.5 de son implémentation JSF sous la Mozilla Public License. Le support de plusieurs effets script.acoulo.us a été



Extrait du site jsfmatrix.net

(4) Un exemple est présenté à l'url suivante : http://forum.java.sun.com/thread.jspa?threadID=590008&messageID=3065356

ajouté. Ainsi le glisser-déposer pour des composants devient possible. Icefaces offre également le Partial submit permettant au développeur de définir quels composants dans la page seront automatiquement soumis. Classiquement, le mécanisme de soumission normale valide et transmet tout le formulaire au cycle JSF. Nous vous conseillons de consulter l'excellent site http://www.jsfmatrix.net/.
Ce site vous permet, pour un composant donné, par exemple la barre de progression, d'avoir une évaluation de chaque solution de composants graphiques.

Impacts / contraintes sur le code html et javascript généré (urls…).
L'exemple ci-dessous montre un exemple de code HTML généré avec JSF pour un formulaire "simple".

```
<form id="loginForm" method="post" action="/bankonet/faces/web/login.jsp" enctype="application/x-www-form-urlencoded">
    Identifiant : <input id="loginForm:userLogin" type="text" name="loginForm:userLogin" />
    <br />
    Mot de passe : <input id="loginForm:userPassword" type="password" name="loginForm:userPassword" value="" />
    <input type="submit" name="loginForm:_id9" value="Valider" />

    <input type="hidden" name="loginForm" value="loginForm" />
</form>
```

De part sa philosophie "composants graphiques", JSF impose quelques contraintes pour le code généré :
- Les formulaires HTML contiennent au moins un champ caché dédié au fonctionnement de JSF (champ "loginForm" dans l'exemple ci-dessus).
- JSF trouve sa valeur ajoutée dans la gestion d'interfaces graphiques "riches". A ce titre JSF repose fortement sur l'utilisation de JavaScript (5).
- Les liens hypertextes doivent toujours se trouver dans un formulaire HTML.

A l'opposé, Struts est peu contraignant pour le code HTML généré et les URLs utilisées.

Intégration de Struts et JSF ensemble ?
A l'ère des interfaces riches, il va bientôt être nécessaire d'adapter une grande partie des applications Struts existantes. A ce titre, il aurait été intéressant de garder les contrôleurs Struts, en migrant les pages JSP vers la technologie JSF (et de continuer de bénéficier des avantages de Struts par rapport à JSF sur certains points). Le projet Apache Struts Faces vise à intégrer les deux technologies ensemble. Cependant, ce projet, toujours en développement, n'est pas très actif. Aucune date n'a été fournie pour une éventuelle version stable. Il faut donc choisir entre Struts et les JSF.

Perspectives

Le futur de Struts est irrémédiablement celui de sa version 2, correspondant au rapprochement des communautés Struts et WebWork. Cette version répond à beaucoup de critiques apportées sur Struts 1 : les sous-contrôleurs peuvent maintenant être de simples objets java (POJO) n'héritant pas d'une classe du Framework Struts, les beans ActionForm ne sont plus obligatoires, la configuration devient plus simple, l'utilisation du moteur de template FreeMarker permet de personnaliser le rendu des tags, etc. Néanmoins, Struts 2, déclaré G.A. (Generally Available) fin février 2007, est très récent, il est donc difficile de gager sur sa pérennité. Quant à JSF, initialement, ce projet devait permettre de composer les pages JSP à l'aide d'assistants graphiques. A l'heure actuelle, les environnements de développement JSF ne sont pas encore satisfaisants, ils sont souvent lourds, peu stables, et incitent les développeurs à utiliser des composants spécifiques à une implémentation donnée (IBM, Sun…), ce qui a pour conséquence de casser la portabilité des applications. Le plug-in open source JSFTools pour Eclipse sera disponible en version stable fin juin 2007. Nous espérons que ce plug-in permettra de rattraper le retard de JSF en matière de composition graphique. Un autre espoir quant à l'évolution de JSF vient du framework JBoss Seam (couture en anglais). Son postulat est de rapprocher simplement la couche WEB (JSF) des couches services métier et persistance (EJB3). JBoss Seam propose d'adresser cette problématique d'intégration, en préférant l'utilisation d'annotations aux configurations XML. Vous pouvez par exemple utiliser vos EJBs directement comme managed bean JSF. Seam répond également aux problèmes classiques du web (gestion du bouton " précédent " du navigateur, intégration aisée d'Ajax, etc.) et propose de nouveaux scopes pour la gestion du contexte. Un cas d'utilisation peut se dérouler sur plusieurs écrans, nécessitant ainsi de définir et maintenir un " état conversationnel " correspondant aux différents échanges (request/response) entre le navigateur et le serveur de ce scénario. Seam propose ainsi un contexte dédié à la gestion aisée d'une conversation. JBoss Seam est " la couture " facile entre les différentes briques EJB3, JSF, jBPM. La simplicité qu'apporte JBoss Seam au développement et la maintenance applicative en font un sérieux challenger dans le monde des architectures JEE et remonte la spécification JSF à la hauteur des espérances que l'on pouvait avoir à ses débuts.

■ **Michael Isvy**
(Consultant SQLI Paris)

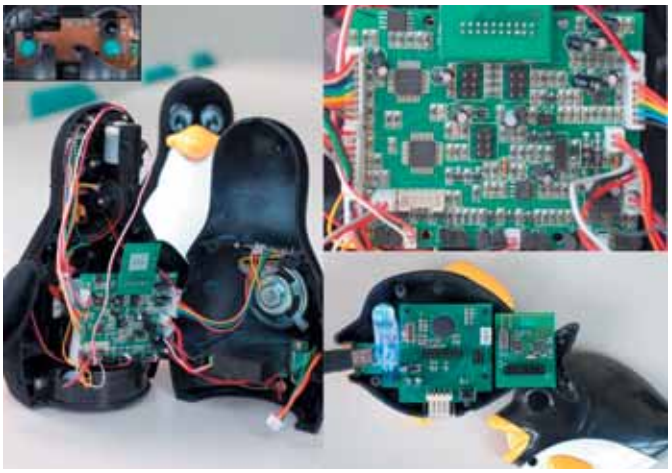
■ **Agnès Crepet**
(Architecte SQLI Lyon)

(5) Utiliser JSF sans le JavaScript nécessite la réécriture d'une bonne partie des mécanismes utilisés par JSF.

EMPLOI :
offres & interviews de recruteurs
www.programmez.com

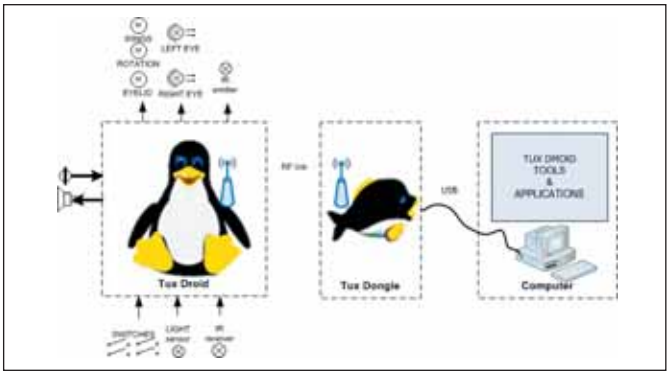
Programmez votre Tux Droid !

Révélé à OSCON aux Etats-Unis et à Solutions Linux en France, Tux Droid est un compagnon de PC intelligent, un périphérique déluré, un avatar roublard inventé tout spécialement pour la communauté Linux et Open Source. En bref, Tux Droid est un robot sans-fil que vous connectez à votre PC. Equipé d'une série de capteurs, d'actuateurs ainsi que d'un micro et d'un haut-parleur, il est capable d'interagir avec son environnement. Que vous soyez débutant ou expert, voici de quoi rendre la programmation amusante.



Un descriptif de ses fonctionnalités et de son fonctionnement interne vous permettra d'appréhender les innombrables possibilités d'application offertes par Tux Droid. N'hésitez donc pas à suivre le tutoriel pour vous familiariser avec sa programmation. Enfin, laissez votre imagination vous guider et développer de nouvelles applications. Distribué sous une licence GPL, tous les codes, les applications, les firmwares et les schémas sont disponibles sur le site communautaire http://www.tuxisalive.com.

1 Tux Droid, un robot



Actuateurs	Capteurs	Audio	Communication
Moteur bouche et yeux	Capteur lumineux	micro	Dongle USB
Moteur ailes	Switch aile droite	haut-parleur	Lien RF 2.4 Ghz avec Tux Droid
Moteur rotation	Switch aile gauche	mémoire flash	I2C pour les processeurs
2 LED bleues	Switch tête		I2C pour capteurs externes
Emetteur infrarouge	Récepteur infrarouge		

Il intègre 3 moteurs qui permettent de bouger les yeux, la bouche, les ailes et de le faire tourner sur lui-même. Tux est aussi équipé de LED dans les yeux commandables séparément. Tux Droid est accompagné d'une télécommande vous permettant de le contrôler à distance. En accédant aux codes, vous pouvez développer de nouveaux programmes et ainsi améliorer vos connaissances robotiques. Il serait cependant dommage de le réduire à cette seule dimension ... les logiciels associés à Tux Droid en font un compagnon d'ordinateur performant et surtout très aisément programmable !

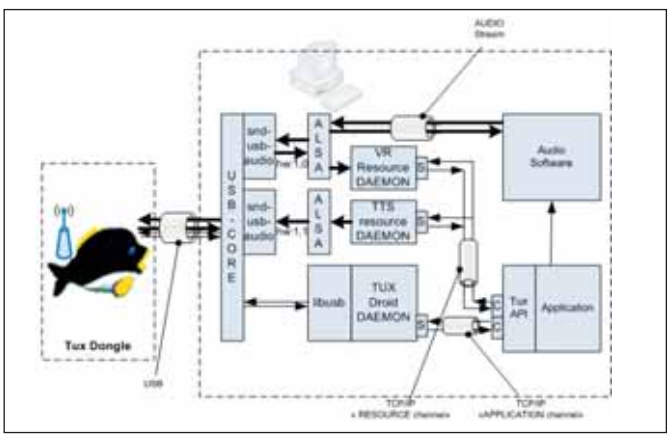
2 Tux Droid, un compagnon d'ordinateur

Grâce à sa communication sans fil, les commandes d'actuateurs, les statuts de capteurs et les flux audio de Tux Droid sont accessibles directement depuis votre ordinateur. Il devient possible d'associer des actions à tout évènement logiciel et, réciproquement, de faire correspondre l'exécution de programmes à des modifications de l'environnement de Tux Droid. Si vous le souhaitez, il bougera les ailes ou clignera les yeux quand vous recevrez un nouvel E-Mail; en appuyant sur son bouton de tête, il vous lira le sujet ou le contenu du message grâce au logiciel de synthèse vocale fourni. Vous pourrez en faire un réveil matin qui vous lira l'heure et vous fera entendre votre radio internet préférée. Il est également possible de contrôler son Tux Droid à distance via ses soquets TCP/IP et de l'associer à votre logiciel de chat ou de VoIP.

3 L'architecture logicielle

L'illustration suivante décrit les principaux composants mis en oeuvre par le logiciel associé à Tux Droid.

Pas de panique ! Il n'est pas nécessaire de connaître dans le détail l'ar-



chitecture logicielle pour programmer son Tux Droid. Toutefois, il est utile de retenir qu'il existe un démon de comportement du robot (Tux Droid Daemon) et un démon de génération de synthèse vocale (TTS Resource Daemon). Ces démons sont accessibles via des soquets TCP/IP en mode serveur (S) auxquels les applications se connectent pour envoyer et recevoir des requêtes. Si on accepte également que Tux Droid soit considéré par le système d'exploitation comme une simple carte son, on dispose de tous les éléments pour en comprendre les mécanismes fondamentaux.

Pour simplifier l'écriture d'applications, une interface de programmation, est proposée. Cette API est écrite en langage Python, un langage orienté objet reconnu pour sa puissance, la richesse de ses librairies mais aussi et peut-être surtout pour la simplicité de sa prise en main.

4 L'API

Le rôle de l'API Python est de fournir des fonctions de communication entre l'application d'une part, les démons et le système d'exploitation d'autre part. L'API se connecte aux démons à l'aide de 2 soquets TCP/IP en mode client. Elle gère l'ensemble du protocole de communication et rend transparents les démons aux yeux du programmeur.

L'API est organisée autour d'un classe principale appelée "tux", structurée en 8 sous-classes gérant chacune un des aspects de Tux Droid. Les deux premières classes s'occupent des échanges de données avec le "daemon" de comportement

- La classe "cmd": permet l'envoi de commandes à Tux Droid pour contrôler les actuateurs. Par exemple, elle permet d'ouvrir les yeux en appelant la méthode suivante:

```
tux.cmd.mouth_open()
tux.cmd.mouth_close()
```

- La classe "status": permet d'obtenir les statuts des capteurs et l'état du robot. Par exemple, la méthode suivante retourne en paramètre le dernier état du capteur de lumière.

```
tux.status.light_level()
```

- La troisième classe gère la communication avec le "daemon" de synthèse vocale.

La classe "tts": met à la disposition du programmeur les méthodes pour lire une phrase, choisir un locuteur, etc.

```
tux.tts.speak('Mon nom est Tux. Tux Droid !')
```

- La quatrième classe facilite la manipulation des statuts venant de Tux Droid en les faisant apparaître comme des évènements logiciels très similaires aux évènements venant d'une fenêtre graphique !
- La classe "event" permet d'associer une fonction à un évènement apparaissant au niveau de Tux Droid. Il suffit pour cela d'assigner à une variable "évènement" de la classe, le nom de la fonction qui doit être exécutée: l'API l'ajoute alors dans son gestionnaire et l'active lorsque l'évènement apparaît. Par exemple, lors de l'appui sur le bouton de tête on pourra demander d'exécuter une méthode de la classe "tts" pour lire l'heure courante.

Comme il est nécessaire au développement d'une application de disposer d'information sur son environnement, les classes suivantes fournissent des données nécessaires.

- La classe "daemon" gère l'état du pilote de comportement (Tux Droid Daemon). Elle permet notamment de connaître le nombre de connexions courantes sur celui-ci, de s'y connecter ou de s'en déconnecter.

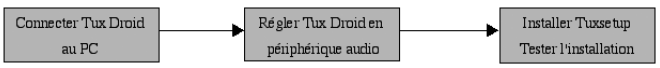
- La classe "hw" gère l'aspect hardware de Tux Droid. Elle permet ainsi de récupérer le nom symbolique de Tux Droid en tant que périphérique audio (par exemple hw:1,0) ou de connaître les numéros de versions des programmes internes de Tux Droid.
- La classe "sys" gère des actions liées au système opératoire (OS). Elle permet d'avoir accès à l'heure, de lancer l'exécution d'un programme externe ou de créer des évènements temporels liés au système. La méthode suivante retourne un entier donnant l'heure en secondes

```
tux.sys.time()
```

- La classe "misc" regroupe des fonctions diverses ne relevant pas des classes citées. On y trouve en particulier les fonctions de documentation des méthodes.

5 Installation

Tux Droid peut être installé sur les distributions les plus courantes de Linux comme Fedora Core, Ubuntu, Mandriva, Xandros, ... Les librairies utilisées par le Tux Droid sont déjà pré-installées dans ces distributions. Les distributions comme Gentoo, Debian ou FreeBSD peuvent nécessiter l'installation de librairies telles que ALSA, libusb et Python. La procédure d'installation se divise en 3 étapes :



Le guide complet d'installation se trouve sur le site www.tuxisalive.com ainsi qu'une vidéo qui vous explique pas à pas la démarche à suivre.

6 Tutoriel

Il est utile de rappeler quelques caractéristiques de Python avant de commencer ce tutoriel. Python est un langage de script. Pour exécuter un programme, vous pouvez l'écrire dans votre éditeur favori, le sauver sous forme d'un fichier ".py" et appeler l'interpréteur Python, en donnant le fichier en paramètre. Une alternative est de lancer l'interpréteur sans paramètre. Celui-ci renvoie une invite de commande et vous pouvez introduire votre code ligne après ligne de façon interactive. Ce

Donnez votre avis sur ce numéro www.programmez.com/magazine_satisfaction.php

tutoriel retient cette façon de faire avec une petite variante: plutôt que de lancer directement l'interpréteur Python nous vous proposons d'appeler le programme "tuxsh" qui, en plus de lancer l'interpréteur Python, charge les différentes classes de l'API. Votre shell Python est dès lors "Tux Droid Ready". Dès que votre installation est effectuée et que Tux Droid est connecté au PC, tapez la commande

```
$ tuxsh
```

Votre écran affichera ce qui suit

```
----------------------------------------------------------
TUXDROID PYTHON API 0.2.0
----------------------------------------------------------
>>> CONNECTED to tuxdaemon
>>> CONNECTED to tuxttsdaemon
>>>
```

Etape 1: Comment fonctionnent les status ?

Les " status " sont les états des différents boutons et capteurs. Toutes les 50 ms, le démon de comportement interroge le robot pour connaître l'ensemble des status et stocke ces valeurs. Du côté logiciel, le démon permet d'y accéder de deux façons: lorsqu'un nouveau status est différent de la valeur connue par le démon, celui-ci envoie un message à l'API pour notifier ce changement; on parle alors de réception "asynchrone". Ce mode d'obtention des status dégage le programmeur de la nécessité de boucler pour vérifier la valeur d'un bouton ou d'un capteur, cette opération étant gérée par le gestionnaire d'évènements de l'API. Le démon peut également fournir la valeur d'un status sur une requête explicite de l'API: on parle alors de réception "synchrone".

Réception asynchrone des status

Le programmeur doit associer une méthode à la notification d'un changement de status. Cela s'effectue simplement en définissant une fonction puis en assignant la fonction à la variable "évènement" correspondante à la notification voulue.
Pour lier l'ouverture puis la fermeture du bec à l'appui sur le bouton de tête on peut écrire :

```
>>> def move_the_mouth() :
...  tux.cmd.mouth_open()
...  tux.cmd.mouth_close()
...
>>> tux.event.on_head_bt_pushed = move_the_mouth
```

Réception synchrone des status

Pour obtenir et imprimer l'état de la connexion sans-fil entre le dongle et le robot on écrira :

```
>>> rf_status = tux.status.rf_state()
>>> print rf_status
```

Etape 2 : Le bouton de tête

La commande suivante bloque le script dans lequel elle est intégrée, jusqu'à ce que l'on appuie sur le bouton de tête ou que le " time-out " fixé à 10 secondes soit dépassé.

```
>>> tux.event.wait_head_bt_pushed(10)
```

Etape 3 : Le son hello

Pour prononcer le son "hello" préstocké dans la mémoire flash de Tux Droid à l'emplacement 4 on écrira :

```
>>> tux.cmd.sound_play(4)
```

Etape 4 : Utilisation du démon de synthèse vocale

Pour sélectionner une voix d'homme française et lui faire dire un texte on écrira :

```
>>> tux.tts.select_voice(SPK_FR_MALE,100)
>>> tux.tts.speak('salut les terriens !')
```

Etape 5 : Séquence

Le code suivant permet de faire exécuter une petite séquence d'actions sur Tux Droid. Celui-ci va lire une phrase, cligner les yeux, bouger la bouche, tourner sur lui-même puis clignoter les leds bleues.

```
# déclaration de la fonction
>>>def seq():
# initialisation du locuteur
... tux.tts.select_voice(SPK_FR_MALE,166)
# ouverture du bec avec l'évènement de début de lecture
...     tux.tts.on_sound_on = tux.cmd.mouth_open
# fermeture du bec avec l'évènement de fin de lecture
...     tux.tts.on_sound_off = tux.cmd.mouth_close
# le locuteur dit "salut tout le monde !"
...     tux.tts.speak('Salut les terriens !')
# fermeture et ouverture des yeux 2 fois
...     tux.cmd.eyes_on_free(4)
# en meme temps, les ailes battent 2 fois
...     tux.cmd.wings_on_free(4)
# extinction des leds si elles étaient allumées
...     tux.cmd.leds_off()
# attente de fin des mouvements des yeux et des ailes (2 sec)
...     tux.sys.wait(2)
# clignotement des leds (20 fois sur 2 sec)
...     tux.cmd.leds_blink(20,25)
# Exécution de la fonction
>>> seq()
```

Par défaut, les commandes du robot bloquent le reste du script tant qu'elles ne sont pas terminées. Le suffixe " _free " ajouté permet d'exécuter ces commandes dans un "thread" concurrent permettant à la suite du script de s'exécuter librement. Toutes les fonctions ne disposent pas de cette option et il faudra se reporter à la documentation de l'API pour connaître celles qui la possèdent. Pour consulter la documentation directement dans "tuxsh ", il suffit de taper

```
>>> tux.misc.doc(tux)
>>> tux.misc.doc(tux.cmd)
```

On obtient alors la liste de toutes les méthodes, variables et constantes de la classe spécifiée (ici la classe tux.cmd).

Etape 6 : Utilisation de la télécommande

Nous allons ici faire cligner les yeux, et battre les ailes de Tux Droid lors

de l'appui sur le bouton " OK "

```
>>> def test() :
... tux.cmd.eyes on free(4)
... tux.cmd.wings on(4)
...
>>> tux.event.on_remote_bt[K_OK] = test
```

Etape 7 : Lancement d'une application externe

Pour lancer une application externe, une fonction"shell" est disponible dans la classe " sys " de l'API. Voici un exemple qui démarre le programme " xmms "

```
>>> tux.sys.shell('xmms')
```

Telle quelle, cette commande bloque le script tant que le programme lancé n'est pas fermé. En ajoutant le suffixe " _free ", cette fonction est exécutée dans un thread concurrent qui se terminera lorsque le programme "xmms" sera clôturé.

```
>>> tux.sys.shell_free('xmms')
```

A partir de là, il est très facile de lier le lancement d'un programme à l'appui d'une touche de la télécommande (Etape 6).

Etape 8 : Le nom symbolique alsa attribué à la carte son tux

Pour se servir d'une carte son en utilisant ALSA, nous devons connaître le nom symbolique qui lui est attribué. Ce nom dépend du nombre de cartes sons présentes sur votre ordinateur et il doit être construit par une fonction. A l'instanciation d'une classe "tux", cette fonction s'exécute, et son résultat est mis dans une variable accessible par le programmeur.

```
>>> print tux.hw.alsa_device
hw:1,0
```

Vous pouvez ensuite vous servir de ce nom pour jouer un son dans votre script.

```
>>> tux.sys.shell('aplay –D %s monwav_8bit8KMono.wav'%tux.hw.alsa_device)
```

Etape 9 : Alarme

Le programme d'exemple "alarm_clock " est proposé sur le site communautaire (http://www.tuxisalive.com/developer-corner/software/alarm-clock). Il utilise une fonctionnalité particulière de la classe "sys" de l'API: la gestion d'évènements temporels. Vous pouvez ainsi exécuter un script ou une commande shell à une heure précise de la journée et définir le comportement qu'aura Tux Droid à ce moment. Tux Droid pourra ainsi lire une phrase, exécuter un script de mouvements, ou bien encore jouer un son. Ce gestionnaire peut accueillir autant d'évènements que vous le souhaitez.
Nous reprenons ici le coeur du script "alarm_clock.py" pour illustration. Ce petit programme vous réveillera à 8h00 en vous lisant un message et en jouant dans "xmms" le fichier "music.wav" que vous aurez stocké au préalable dans le même répertoire que votre script !

```
#!/usr/bin/python
# -*- coding: latin-1 -*-
import sys
sys.path.append('/opt/tuxdroid/api/python')
from tux import *

def on_alarm():
    tux.tts.select_voice(SPK_FR_MALE,120)
    tux.tts.speak("Il est l'heure d'écouter de la musique !")
    tux.sys.shell_free("xmms music.wav")

h = 8   # Heure
m = 0   # Minute
s = 0   # Seconde
tux.sys.add_time_event(2,on_alarm,9999,99,99,h,m,s)
```

Pour l'exécuter après l'avoir sauvé dans votre éditeur, tapez la commande suivante :

```
$ python -i alarm_clock.py
```

Bien que d'apparence élémentaire, les codes proposés dans ce tutoriel permettent des constructions très élaborées: les gestionnaires d'évènements et l'exécution de commandes concurrentes (suffixes "_free") fournissent des mécanismes puissants de contrôle d'un robot à partir d'un ordinateur. Plus encore, l'accès à ces fonctions depuis un langage de haut niveau tel que Python fait de Tux Droid un compagnon d'ordinateur aux applications surprenantes seulement limitées par l'imagination du programmeur.

7 Là où l'amusement commence

A partir du tutoriel, n'hésitez pas à imaginer les applications qui vous seront utiles.

En voici une liste non exhaustive :

- Un lecteur audio portable contrôlé par la télécommande
- Lire certaines news et les e-mails avec la synthèse vocale de Tux Droid (15 langues sont disponibles)
- Alerter lors de la réception d'un e-mail ou quand quelqu'un est prêt pour " chatter "
- Crier quand votre compilation a échoué et hurler quand vous avez fait une vente sur votre site de vente en ligne.
- Téléphoner sur internet
- Être votre baby-phone, jouer des berceuses, vérifier le bruit dans une pièce
- Être un réveil avec vos mp3 favoris ou votre radio internet
- Contrôler votre HIFI ou votre TV de votre ordinateur
- Créer des quizz interactifs ou des petits jeux pour vos enfants

Vous trouverez l'ensemble des codes sources de Tux Droid ainsi que des schémas et toute l'information nécessaire pour s'amuser avec Tux Droid sur le site communautaire www.tuxisalive.com.
Tutoriel vidéo : http://www.tuxisalive.com/media

David, Remi & Co de Kysoh

LANGAGE : DIVERS
DIFFICULTÉ : ●●○○○
APPLICATION : OUTILS
SOURCE : non

(Re)Découvrir Emacs

Avez vous besoin d'un IDE d'une puissance incroyable, d'une souplesse inégalable et d'une polyvalence sans égale ? Cet IDE existe. On ne peut pas vraiment dire qu'il vient de sortir, il s'appelle Emacs, et surtout... c'est plus qu'un IDE :)

C'est dit-on, dans les vieux pots que l'on fait les meilleures soupes. Appliquons ceci à l'informatique, savante cuisine. La soupe serait un projet tout rempli de fichiers soigneusement mitonnés, le pot serait l'outil de développement. L'IDE comme on dit maintenant. Lorsque je cuisine en C++ ou .Net sous Windows, j'utilise volontiers Visual Studio 2005. Pas forcément avec le compilateur C++ de Microsoft d'ailleurs. Mais toujours est-il que VC 2005 est un outil remarquable... sous Windows et pour C++ ou .Net seulement. Je ne peux pas y écrire du code Haskell et je ne peux pas m'en servir sous Linux/Unix. Lorsque je cuisine en Java j'utilise volontiers Netbeans ou Eclipse. Les deux outils sont incontestablement efficaces pour coder en Java. Ils ont aussi, Eclipse surtout, la prétention d'être polyvalents. Qu'en est-il ? Je me suis laissé dire récemment que nombre de développeurs de la JVM à Santa Clara n'utilisent pas Netbeans, mais l'outil, disons de plus vieille facture, qui nous intéresse aujourd'hui, ce qui en dit long. J'ai récemment vu un Eclipse et son plug-in CDT faire preuve d'une excessive lenteur sur une machine bi-xeon 3 Ghz avec 3 Go de Ram en travaillant avec un projet C++ d'environ 10 000 fichiers. Selon moi, la polyvalence, c'est plus qu'écrire du code Java ou C++. Un outil est polyvalent lorsqu'il me permet en même temps de: lancer une compilation en tâche de fond, déboguer à distance, jouer à Tetris ou à Gomoku, écrire un script bash, accéder à un référentiel CVS, surveiller les processus de mon système, écrire du code Java, C++, Python, Tcl, Perl, Haskell, Lisp, SQL, éditer du HTML ou du XML, éditer un fichier distant via FTP, charger du code dans un noeud Erlang et bien d'autres choses encore :) Pour cuire cette soupe compliquée, il n'y a que le vieux pot Emacs. Emacs ne laisse pas indifférent. Il a ses inconditionnels et ses allergiques. Je fais partie des premiers, et par cet article je souhaite vous faire partager mon enthousiasme pour ce qui est finalement beaucoup moins un éditeur qu'un environnement Lisp. D'où sa puissance.

1 Emacs est un environnement Lisp

L'histoire EMACS débute il y a plus de trente ans. Au départ EMACS ou Editeur de MACroS, pour l'outil TECO, devait simplifier un travail ardu dans un monde où ce que nous appelons aujourd'hui un éditeur n'existait tout simplement pas. Rapidement EMACS est devenu Emacs, soit un vrai éditeur. Le tout premier Emacs nous le devons à ... James Gosling, le créateur de Java. Celui-ci a développé en 1981 le Gosling Emacs, doté d'un langage interne présentant quelques analogies avec Lisp. En 1984 Richard Stallman a repris EMACS et le Gosling Emacs, a doté le tout d'un vrai Lisp, créant ainsi un des tout premiers projets GNU, le GNU Emacs, c'est à dire Emacs tel que nous le connaissons aujourd'hui. Emacs est avant tout un environnement de développement Lisp ce qui fait sa puissance, un peu comme Eclipse est un environnement de développement Java. On voit que l'idée de l'IDE autoprogrammable ne date pas d'hier :) Aborder Emacs comme un environnement Lisp est l'aborder de la bonne façon. Cela permet de s'affranchir d'un apprentissage laborieux, voire rebutant, à base de combinaisons de touches ésotériques. Aborder d'emblée Emacs comme un environnement Lisp, c'est comprendre ce que l'on fait et c'est donc profiter réellement de la puissance de cet outil. Lisp souvent effraie, pourtant contrairement à l'idée reçue, c'est un des langages les plus simples et faciles qui soit. Et pour le montrer, plutôt qu'énumérer des combinaisons de touches, je vais vous montrer comment il est facile de modifier un mode d'édition Emacs avec quelques lignes de Lisp. Alors qu'en comparaison, modifier un plug-in Java, à supposer que nous disposions des sources, est plutôt compliqué.

2 Anatomie d'Emacs

Emacs est un environnement Lisp orienté texte, ce qui ne surprend pas. Le texte est contenu dans des buffers qui sont des objets Lisp. Il est visualisé dans des fenêtres (window) et les fenêtres sont contenues dans un frame. Un buffer existe et vit, qu'il soit ou non

Edition de code C++ sous Emacs et ECB.

visible, à travers une fenêtre. L'illustration ci-dessus, digne d'une perspective Eclipse, montre un frame Emacs contenant 6 fenêtres sous le mode ECB (Emacs Code Browser). La sixième fenêtre est la ligne tout en bas. Elle visualise le minibuffer, dont le rôle est d'afficher des messages ou de saisir une commande ou des paramètres à l'intention d'une commande. Ce minibuffer est l'équivalent des listeners des environnements Lisp purs, par exemple Lispworks. Une commande est un nom de fonction Lisp, puisque sous Emacs tout est Lisp. Par exemple find-file pour ouvrir un fichier. Une commande peut être saisie directement dans le minibuffer. D'abord saisir M-x, c'est-à-dire Alt-x ou Esc-x, pour basculer dans le minibuffer, puis saisir le nom de la commande soit find-file, puis [return]. Le minibuffer vous demande alors le nom du fichier et ainsi de suite. Nous n'allons pas passer les fonctions

Lisp/Emacs en revue, un lecteur de Programmez! est capable de le faire seul en consultant une documentation. Et surtout un lecteur de Programmez! se doit de reprogrammer ces combinaison de touches parfois si rebutantes. Car Emacs est un environnement Lisp (vous l'ai-je déjà dit? ;) et comme tel il est entièrement programmable. Regardons plutôt Lisp :)

3 Notion de Lisp

A la base, un type de données règne en maître sur Lisp, la liste. Voici la représentation d'une liste:

```
(1, "deux", trois)
```

Qui contient l'entier 1, la chaîne deux, et un objet quelconque lié au nom trois. Par cet exemple on comprend que Lisp est typé dynamiquement. Une liste est constituée d'une tête (le car) et d'une queue, le (cdr). Ici le car est (1), le cdr est ("deux", trois). Nous avons bien dit que nous avions une représentation d'une liste. Il ne s'agit pas d'une ligne de code. En Lisp une ligne de code, qui commence et se termine par une parenthèse, est une forme, que l'interpréteur Lisp... interprète. En simplifiant, on peut dire que cette interprétation, ou évaluation en Lisp, consiste à appliquer le car sur le cdr d'une liste. Exemple:

```
(+ 1 2 3 4)
```

Va appliquer l'opérateur d'addition sur la liste de nombre 1, 2, 3, 4 et le résultat de l'évaluation sera donc 10. C'est donc tout simple, et le croirez vous, c'est à peu près tout ce qu'il faut savoir pour se lancer dans l'aventure avec Lisp. Bien sûr, il existe des formes spéciales mais le principe de base reste valable. Exemples:

```
(list 1 2 3)
```

construit la liste dont la représentation est (1 2 3) en appliquant l'opérateur/car list sur le reste de la liste, le cdr autrement dit

```
(defun carre (x)
  (* x x))
```

Cette forme spéciale définit une fonction. Le car defun s'applique à la liste de ce qui le suit à savoir
- le nom de la fonction: carre
- la liste (forcément...) des paramètres qu'elle reçoit, ici (x)
- la liste des formes qu'elle devra exécuter lors de son invocation. Ici, une seule forme (* x x) qui calcule donc le carre de x.

Notre fonction ainsi définie s'invoquerait non pas carre(2), nous ne sommes pas en C, mais:

```
(carre 2)
```

Lisp rebute souvent à cause des nombreuses parenthèses de sa syntaxe. Il s'agit en fait d'un faux problème. Les éditeurs Lisp, dont Emacs ;) indentent le code correctement et la lecture de celui-ci n'est qu'une question d' habitude, très facile à prendre. Quant au nombre de parenthèses fermantes qu'il faut écrire, essayez et vous verrez qu'il n'y a même pas besoin d'y faire très attention.

3 Petite personnalisation d'Emacs

Nous en savons plus qu'il n'en faut pour écrire des programmes. Mais avant tout, soyons confortables. Pour cela, définissons quelques touches bien pratiques. Définition de touches veut dire ici que chaque touche définie provoquera l'invocation d'une fonction Lisp. Au démarrage, Emacs lit un fichier de configuration, .emacs qui n'est rien d'autre qu'un programme Lisp, et conformément à la logique d'Emacs où tout est Lisp. Ce fichier se trouve sous ~/.emacs sur système Unix et C:/.emacs sous Windows. Lancez emacs ainsi :

```
emacs ~/.emacs ou bien emacs c:\.emacs
```

Et au début du fichier, ajoutez quelques formes en vérifiant quand même qu'elles ne sont pas en conflit avec d'autres un peu plus loin. En effet, les distributions Linux fournissent généralement un .emacs assez compliqué. Pour vous faire la main, le mieux est probablement de faire une copie de ce dernier et de travailler avec un .emacs vierge. Donc ajoutez:

```
(global-set-key [f2] 'find-file)
(global-set-key [f3] 'save-buffer)=
(global-set-key [f9] 'eval-current-buffer)
(global-set-key [f11] 'list-buffers)
(global-set-key [f12] 'other-window)
```

Pour définir quelques touches de fonctions. (Rien ne vous empêche de choisir d'autres touches :)
Depuis le menu, sauvegardez le buffer et quittez Emacs. Relancez Emacs normalement, sans lui donner .emacs en argument car ce dernier sera lu et évalué automatiquement, et nos définitions de touches seront actives. Voici ce que nous avons fait :

f2 : charger un fichier dont le nom sera donné dans le minibuffer. Commande qui double la commande classique C-x C-f. Qui a dit que les combinaisons de touches sont pénibles sous Emacs ? ;)
f3 : sauvegarder le buffer courant dans son fichier associé, en doublon de C-x C-s.
f9 : Evaluer le buffer courant. On se servira de cette touche si le buffer courant contient du code Lisp. En Lisp on ne dit pas exécuter, mais évaluer :)
f11 : ouvre une fenêtre montrant un buffer listant les buffers existants. Placer le curseur sur le nom d'un buffer provoque son affichage dans une fenêtre.
f12 : Fait naviguer le curseur d'une fenêtre à l'autre. Cette touche s'utilise de concert avec la précédente.

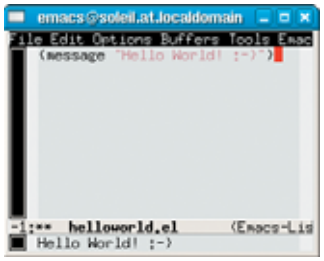
Sachez enfin que si une commande ne se comporte pas comme vous voulez, vous pouvez toujours la faire avorter avec la combinaison de touche C-g

4 Un helloworld sous Emacs

Comme avec tous les langages, la tradition veut que l'on écrive un helloworld. Créons donc un fichier lisp, que nous baptiserons helloworld.el
- presser f2
- saisir helloworld.el dans le minibuffer. Valider avec [return]

Dans la fenêtre du buffer, écrivez:

```
(message "Hello World! :-)")
```



Un Hello World dans Emacs

C'est tout. Pressez F9 pour faire exécuter le programme, autrement dit évaluer le buffer, et vous voyez le message Hello World :-) après dans le minibuffer.

5 Customiser le mode html-mode

Comme pour tout le reste, j'utilise Emacs pour éditer mes fichiers HTML. J'utilise le mode html-mode.el que vous pouvez télécharger à http://fsinfo.cs.uni-sb.de/~abe/htmlmode/download.html. Et bien sûr j'ai adapté ce mode à ma convenance et je veux vous montrer comme c'est facile. (Remarque: cet exemple est volontairement naïf dans le cadre de cette initiation, la méthode ad-hoc pour customiser un mode est de lui adjoindre un hook, sans toucher au mode lui-même). D'origine ce html-mode ne sait pas insérer une balise div. Très bien, commençons par définir un raccourci clavier (en gras ci-dessous):

```
(define-key html-mode-map "\C-c\C-c" 'html-add-citation)
(define-key html-mode-map "\C-c\C-d" 'html-add-div)
(define-key html-mode-map "\C-c\C-e" 'html-add-emphasized)
```

Nous devons donc écrire une fonction Lisp html-add-div. C'est un jeu d'enfant, la voici:

```
(defun html-add-div ()
  "Add a div"
  (interactive)
  (insert "<div class= ></div>")
  (forward-char -6))
```

Le code peut être placé n'importe où dans le fichier html-mode.el. La chaîne "Add a div" est la documentation interne de la fonction. La forme "interactive" fait qu'il sera possible d'appeler la fonction dans le minibuffer via M-x. La forme "insert" se passe de commentaires. La dernière forme positionne après coup le curseur entre les balises div. Autant faire les choses proprement :) Le mode html permet d'insérer les caractères spéciaux, comme é pour é. Mais je n'aime pas m'en servir, car alors le texte est peu lisible. Je préfère écrire le texte normalement puis invoquer une fonction qui transforme d'un coup les caractères accentués, après écriture complète du texte. Il reste à écrire cette fonction, ce qui est également un jeu d'enfant. D'abord définissons une liste d'associations caractère accentué/caractère HTML spécial. La liste n'est pas limitative et n'est là que pour l'exemple :

```
(defvar html-char-dict
  (list
    '("é" . "&eacute;")
    '("è" . "&egrave;")
    '("ê" . "&ecirc;")
    '("à" . "&agrave;")
    '("ç" . "&ccedil;")
    '("ù" . "&ugrave;")))
```

Ensuite la fonction. Elle consiste à parcourir la liste. Pour chaque élément la clé (par exemple "é") est cherchée dans le buffer et si trouvée, remplacé par le caractère HTML spécial correspondant. La fonction parcourt la liste récursivement. En voici l'algorithme :

```
text2html1 dict
```

à moins que la liste dict ne soit vide :
- se placer au début du buffer
- prendre le premier élément de la liste
- rechercher dans le buffer chaque occurence de la clé de cet élément et la remplacer par la valeur associée de l'élément.
- appeler text2html1 avec la queue (alias le cdr) de dict.

Enfin, nous écrivons un frontal à text2html1 qui permettra d'invoquer la fonction depuis le minibuffer et d'afficher un message dans le minibuffer après le travail. Voici le code complet :

```
(defun text2html1 (dict)
  (unless (eq dict nil)
  (beginning-of-buffer)
  (replace-string (car (car dict)) (cdr (car dict)))
  (text2html1 (cdr dict))))
```

```
(defun text2html ()
  (interactive)
  (text2html1 html-char-dict)
  (message "text2html done :-)"))
```

Simple et concis. A l'utilisation, après écriture du texte HTML faire simplement :

```
M-x text2html
```

dans le minibuffer et le tour est joué :) Le mode d'édition HTML est maintenant beaucoup plus agréable. Par cet article et cette approche peu conventionnelle j'espère vous avoir fait (re)découvrir et apprécier Emacs. Car il est bien autre chose qu'un éditeur et pour un programmeur, il est formidable, tout simplement. Ne lisez plus les didacticiels Emacs à base de combinaisons de touches, lisez la documentation de l'API Emacs et un monde infini de possibilités va s'ouvrir à vous :)

par Frédéric Mazué - *fmazue@programmez.com*

L'INFORMATION PERMANENTE
✔ Abonnez-vous
au format NUMERIQUE (PDF)
Tarif unique pour
LE MONDE ENTIER 35 €
(abonnement exclusivement en ligne www.programmez.com)
COUPON D'ABONNEMENT page 53
de ce numéro ou www.programmez.com
Programmez!
LE MAGAZINE DU DÉVELOPPEMENT
✔ 3 NUMEROS GRATUITS
Faites des économies, abonnez-vous !
Exemple : en France, vous gagnez 20 euros, soit plus de 3 numéros !
✔ ETUDIANTS
Tarif spécial 39€ (France métropolitaine exclusivement)

Manipulations avancées avec PowerShell

Programmez! 93 vous a présenté PowerShell, le nouvel environnement en ligne de commandes avec langage de script intégré de Microsoft. Nous allons aujourd'hui plus loin que les manipulations de base pour exploiter quelques fonctionnalités puissantes de cet outil.

Résumons nous. Avec PowerShell, Windows est enfin doté d'un environnement en ligne de commandes qui n'a pas à pâlir devant les Shell Unix. En découvrant les bases du maniement de PowerShell, nous avons constaté qu'il était possible d'en faire un usage "à la Unix". Mais nous avons constaté également en quoi PowerShell se distingue des Shell Unix. Alors que ceux-ci manipulent essentiellement du texte à travers des commandes système, PowerShell travaille avec des objets. PowerShell a pour socle la plate-forme .Net et les objets qu'il manipule sont tout naturellement des instances de classes .Net, parfois adaptées pour la cause. Cette plate-forme simplifie beaucoup la programmation Windows. Considérons par exemple la WMI, ou Windows Management Interface, de Windows. Cette API surpuissante permet de surveiller un système Windows local ou distant dans sa totalité. Le revers de la médaille étant que WMI est basée sur COM et que sa programmation en C++ peut vite tourner aux travaux d'Hercule. Sous .Net, le travail avec la WMI est considérablement simplifié grâce aux fonctionnalités d'interopérabilité de la plate-forme avec COM. Considérant que PowerShell est avant tout un langage de script en .Net, nous avons normalement avec lui un outil qui permet d'écrire une application d'administration Windows complète, ce que cet article va tenter d'illustrer. Vous trouverez tous nos scripts d'exemple sur le Cd-Rom accompagnant le magazine, ou sur notre site.

1 Pour se faire la main

Reprenons quelques manipulations de base, non pas en pensant "Shell", mais en pensant objets .Net. Ouvrez un interpréteur PowerShell et saisissez:

```
$hello = "Hello World"
```

Ensuite saisir simplement

```
$hello
```

provoque l'affichage du message "Hello World" comme on s'y attend légitimement. Qu'est-ce donc que la variable $hello ? C'est un objet, une instance de classe .Net puisque sous PowerShell tout est ainsi. Nous pouvons savoir de quelle instance il s'agit en interrogeant l'objet, puisque .Net est réflexif.

```
$hello.gettype()
```

nous informera que $hello est de type String, c'est-à-dire une instance de System.String, classe qui a pour base System.Object. Nous avons instancié notre objet de la façon naturelle et immédiate pour un langage script. Mais nous aurions pu procéder différemment en instanciant sciemment une classe .Net. PowerShell vient pour cela avec une Cmdlet qui gagne vraiment à être connue: new-object qui porte très bien son nom. Utilisons-là :

```
$hello = new-object System.String "Hello World"
```

Le résultat est clairement le même que précédemment, mais nous avons exprimé explicitement ce que nous faisons. Notons à ce stade que PowerShell nous offre un sucre syntaxique consistant à placer entre crochets [] le type de la classe à instancier. Ainsi nous aurions pu écrire:

```
$hello = [System.String]"Hello World"
```

ou plus simplement, puisque System est un assembly toujours référencé

```
$hello = [String]"Hello World"
```

Ou encore, puisque PowerShell n'est pas sensible à la casse

```
$hello = [system.string]"Hello World"
```

2 Des fenêtres sous PowerShell

PowerShell nous offre des Cmdlet pour le formatage des sorties en ligne de commande. Mais puisque PowerShell c'est avant tout .Net, pourquoi ne pas ouvrir des fenêtres de type Windows.Forms.Form avec l'intention d'écrire dedans ? Essayons:

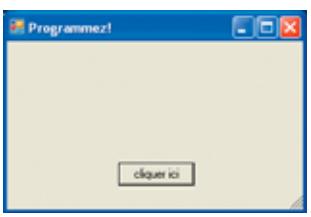
```
$hello = new-object Windows.Forms.Form
```

Nous recevons un message d'erreur nous informant que le type Windows.Forms.Form est inconnu, peut-être parce que l'assemblage le contenant n'est pas chargé. De même qu'en programmation .Net sous Visual Studio, les assemblies .Net ne sont pas chargés automatiquement. Nous devons le faire manuellement :

```
[System.Reflection.Assembly]::LoadWithPartialName("System.windows.forms")
```

Remarquez la variante dans l'emploi du sucre syntaxique expliqué plus haut. En effet, cette fois il s'agit d'accéder à une méthode statique de la classe System.reflection.Assembly, donc sans instance de classe. La méthode LoadWithPartialName pourrait être remplacée par Load à condition de donner le nom complet de l'assembly. Exemple :

```
[Reflection.Assembly]::Load( `
"System.Windows.Forms, Version=2.0.0.0, Culture=neutral, `
PublicKeyToken=b77a5c561934e089")
```



Un Hello World fenêtré avec PowerShell.

Remarque: les quote inverses ou "backtick" servent à écrire la commande sur plusieurs lignes. Nous pouvons maintenant écrire un helloworld fenêtré en PowerShell, voici le code (helloworld.ps1)

```
[System.Reflection.Assembly]::LoadWithPartialName( `
   "System.windows.forms") | out-null
```

```
$hello = new-object Windows.Forms.Form
$hello.text = "Programmez!"
#$hello.size = new-object Drawing.Size (300,200)
# ou plus simplement
$hello.size = "300, 200"


$button = new-object System.Windows.Forms.Button
$button.text = "cliquer ici"
#$button.location = new-object Drawing.Point (110, 120)
# ou plus simplement
$button.location = "110, 120"
$button.add_Click(
  {
      $hello.Close()
  }
)

$hello.Controls.Add($button)
# Faire du bouton un bouton par défaut
$hello.AcceptButton =  $button

$hello.showdialog() | out-null
"Abonnez vous :-)"
```

Ce code parle de lui-même. A l'exception d'une chose. Nous créons un bouton, mais comment avons nous fait pour l'associer à un délégué dont le code ferme la fenêtre ? En PowerShell nous ne pouvons pas travailler avec les délégués comme on le fait habituellement avec un langage .Net. Et, pour nuancer ce que nous avons dit au début de cet article, les instances d'objet .Net sont des instances d'adaptateurs d'objets. C'est le cas de notre bouton. Si l'on écrivait

```
$button | get-member
```

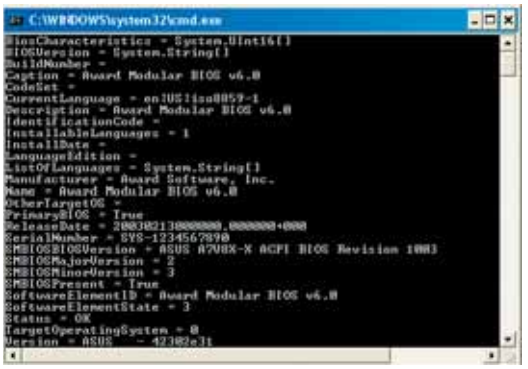
Nous verrions listées toute une série de méthodes préfixées add_xxxx. C'est ainsi que nous avons découvert la méthode add_Click que nous utilisons. Le reste est simple et illustre au passage que PowerShell supporte les méthodes anonymes.

3 Manipuler la WMI avec PowerShell

Les numéros 78 et 79 de Programmez! présentaient la WMI et sa programmation sous .Net. Nous ne reviendrons pas dans le détail ici.Voici un morceau de code en VB .Net qui s'enquiert du BIOS de notre machine.

```
Imports System.Management

Module Module1
  Sub Main()
      Dim bios_management As ManagementClass
      bios_management = New ManagementClass("Win32_Bios")
      Dim all_bios As ManagementObjectCollection =
          bios_management.GetInstances()
      Dim bios As ManagementObject
      For Each bios In all_bios
          Dim bios_prop_all As PropertyDataCollection =
              bios.Properties
          Dim bios_prop As PropertyData
          For Each bios_prop In bios_prop_all
              Console.Write(bios_prop.Name & " = ")
              Console.WriteLine(bios_prop.Value)
          Next bios_prop
      Next bios
      End Sub
```



Les propriétés du BIOS de ma machine, obtenues avec VB.Net.

```
End Module
```

Ce code obtient toutes les instances de la classe Win32_BIOS, une instance par BIOS, et affiche ensuite les propriétés. Chercher tous les BIOS peut sembler stupide. C'est toutefois naturel de chercher toutes les instances d'une classe en programmation WMI .Net. Pour la classe Win32_Bios, on ne trouve qu'une instance, voilà tout. En revanche si vous prenez ce code, remplacez Win32_Bios par Win32_Process et recompilez sans rien changer d'autre, vous obtenez la liste de tous les processus tournant dans le système ainsi que toutes leurs propriétés. Ce serait intéressant de pouvoir faire la même chose en PowerShell dans des scripts d'administration. De même qu'il existe la Cmdlet new-object, nous avons la Cmdlet get-wmiobject. Essayons-la sans attendre :

```
$bios = get-wmiobject -computer "soleil" `
  -namespace "root\cimv2" "Win32_Bios"
```

Cette commande complète permettrait d'accéder à la WMI d'une machine distante si besoin. Toutefois soleil est le nom de ma machine de travail donc une machine locale. root\cimv2 est l'espace de nom par défaut des classes WMI. Donc nous pouvons réduire la commande à

```
$bios = get-wmiobject "Win32_Bios"
```

Qu'avons nous dans la variable $bios ?

```
PS C:\Scripts\PowerShell> $bios

SMBIOSBIOSVersion : ASUS A7V8X-X ACPI BIOS Revision 1003
Manufacturer      : Award Software, Inc.
Name              : Award Modular BIOS v6.0
SerialNumber      : SYS-1234567890
Version           : ASUS   - 42302e31
```

Nous sommes très loin de l'affichage exhaustif montré par notre code VB.Net. Nous pouvons cependant afficher une propriété particulière:

```
PS C:\Scripts\PowerShell> $bios.SoftwareElementState
3
```

Si nous saisissons $bios | get-member nous obtenons bien la liste de toutes les propriétés, mais nous voyons aussi des propriétés et des méthodes parasites par rapport à la sortie émise par le code en VB.Net. La raison est que nous avons affaire à nouveau à un adaptateur d'objet.

4 Passer outre les adaptateurs d'objets

Les adaptateurs d'objets PowerShell ont leurs avantages et aussi leurs inconvénients. Ici nous ne pouvons pas obtenir une collection de toutes les propriétés par $bios.properties. Si nous le pouvions, nous obtiendrions facilement une sortie analogue à celle du code VB .Net et même nous pourrions présenter les résultats dans une fenêtre. Essayons d'atteindre ce but, le dernier pour cet article. Pour cela, imitons avec PowerShell le code VB.Net. Nous commençons par obtenir non pas une instance de Win32_Bios, mais une instance sur sa classe de gestion.

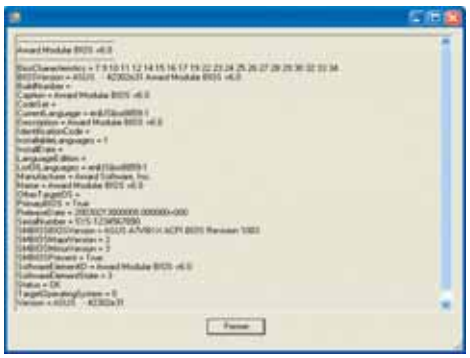
```
$management_class = new-object `
   system.management.managementclass "Win32_Bios"
```

Remarquons bien l'emploi de new-object et non de get-wmiobject. Jusqu'ici tout va bien. Nous essayons ensuite d'obtenir les instances de classes, à l'instar du code VB.Net

```
$all_instances = $management_class.GetInstances()
```

PowerShell ne dit rien. Cela semble fonctionner. Pourtant il apparaît que $all_instances reste désespérément à null. Le problème se situe en amont, car en fait $management_class ne dispose pas de méthode GetInstances comme on s'y attendrait, ce que l'on peut vérifier par $management_class | get-member. Voilà un mauvais tour que nous jouent PowerShell, qui pourrait émettre un message d'erreur à ce niveau, et l'adaptateur d'objets de notre classe. Décidément, le besoin se fait sentir de passer outre les adaptateurs d'objet. C'est possible. La méthode est très bien cachée dans la MSDN. En fouillant, on finit par apprendre que les adaptateurs d'objet nous cachent la propriété psbase qui permet de passer outre et d'accéder aux vrais membres de la vraie classe .Net. Comme ceci:

```
$all_instances = $management_class.psbase.GetInstances()
```



Accès complet aux classes WMI/.Net et affichage de leurs propriétés.

Moyennant quoi nous pouvons écrire un script complet, rassemblant tout ce que nous avons dit dans cet article, pour présenter les instances des classes WMI et leur membres dans une fenêtre. Accessoirement, le script montre comme écrire une fonction en PowerShell et comment exploiter les paramètres en ligne de commande. Vous pouvez donc invoquer le script par wmi-graphic.ps Win32_Bios ou wmi-graphic.ps Win32_Process ou en donnant n'importe quelle autre classe en paramètre, et votre système n'aura plus de secret pour vous. PowerShell lui, en conserve quelques uns... pour une prochaine fois ;)

```
[System.Reflection.Assembly]::LoadWithPartialName( `
        "System.windows.forms") | out-null

function run ([string]$wmicl)
{
   $wmiform = new-object Windows.Forms.Form
   $wmiform.size = "600, 500"
   $button = new-object System.Windows.Forms.Button
   $button.text = "Fermer"
   $button.location = "260, 420"
   $button.add_Click(
        {
             $wmiform.Close()
        })

   $wmiform.Controls.Add($button)
   # Faire du bouton un bouton par défaut
   $wmiform.AcceptButton =  $button

   $tb = new-object Windows.Forms.TextBox
   $tb.location = "10, 10"
   $tb.multiline = 1
   $tb.size = "570, 400"
   $tb.scrollbars = [System.windows.forms.Scrollbars]::Vertical
   $wmiform.Controls.Add($tb)

   $management_class = new-object `
         system.management.managementclass $wmicl
   $all_instances = $management_class.psbase.GetInstances()
   foreach($inst in $all_instances)
   {
        $tb.appendtext("----------")
        $tb.appendtext([Environment]::newline)
        $tb.appendtext($inst.name)
        $tb.appendtext([Environment]::newline)
        $tb.appendtext("----------")
        $tb.appendtext([Environment]::newline)


        $inst_prop_all = $inst.psbase.Properties
        foreach($inst_prop in $inst_prop_all)
        {
             $prop = $inst_prop.Name
             $prop += " = "
             $prop += $inst_prop.Value
             $tb.appendtext($prop)
             $tb.appendtext([Environment]::newline)
        }
   }

   $wmiform.showdialog() | out-null
}

run $args[0]
```

Frédéric Mazué - *fmazue@programmez.com*

Recevez la Newsletter www.programmez.com/inscription.php

Développeur de jeux vidéo : une aventure possible

Les statistiques du Cnam montrent une constante augmentation des formations aux métiers du jeu vidéo depuis leur " vraie " mise en place en 2005. C'est que le marché, initialement formé d'autodidactes, exige maintenant une chaîne de production construite.

La production d'un jeu vidéo fait appel à une succession ou une coordination de rôles bien spécifiques. En même temps, la multiplication des tout petits studios, s'appuyant le plus souvent sur une seule idée de middleware, montre qu'une forme de développement à la mode " garage " continue à exister en parallèle. Cependant, dans un cas comme dans l'autre, l'amateurisme n'est plus de mise.

Profils en indépendant

Pour se lancer en indépendant, par exemple en fondant sa micro-entreprise, il faut avant tout définir le rôle qu'on veut jouer, prestataire, éditeur, ou inventeur. Le prestataire effectue des tâches ponctuelles dans sa spécialité pour un éditeur ou un studio, il apporte un service en matière de gestion réseau, de transfert de plate-forme, d'intelligence artificielle, etc. Le nombre des prestataires va croissant en France, de nombreux studios et éditeurs confiant en externe certaines tâches pour ne pas alourdir leur structure. L'éditeur est le pari le plus risqué. Sauf héritage d'un oncle milliardaire, le problème principal est le réseau de distribution du ou des jeux, qui sera initialement confidentiel. Heureusement, le téléchargement web et le paiement en ligne, par Paypal par exemple, peuvent assurer des revenus suffisants à un petit projet. Il faut cependant compter avec le piratage peer-to-peer qui, même sur un petit jeu, entame les ventes de plus de moitié. L'inventeur semble enfin la voie la plus prometteuse : globalement, il vend une compétence innovante s'appuyant sur une avancée du hardware ou sur une invention purement logicielle de technique ou d'environnement ludique, un " moteur ". C'est le secteur de prédilection des outils et du middleware. Très durable et pérenne si l'invention prouve sa valeur, par la déclinaison des versions successives sur plusieurs années.

Votre " Futurn " ?

Que vous vouliez devenir développeur de jeux vidéo en indépendant ou comme employé d'une structure, studio ou éditeur, nous vous recommandons d'aller consulter sans tarder le site de l'association FUTURN (www.futurn.net) dont les dossiers, le forum, les aides et formations viennent remarquablement soutenir la création francophone en matière de jeux vidéo. Indispensable.

Rôles dans une équipe

La programmation d'un jeu vidéo s'articule aujourd'hui autour d'une vingtaine de rôles aux tâches bien définies. Ce qui ne veut pas toujours dire 20 programmeurs distincts : seuls les grands projets répercutent complètement la division des rôles sur le nombre des emplois. Cependant, tout poste de développeur de jeu a nécessairement un profil de départ, qui peut, et sans doute doit, s'élargir au fil du développement et de la collaboration avec l'équipe ! Voici les profils qu'on peut ébaucher :

Programmeur moteur 2D

Il gère l'affichage, les collisions, la gestion de la mémoire, les algorithmes bitmap à partir d'une programmation orientée objet, avant tout C++ et Visual C++.

Programmeur moteur 3D

Au-delà du code, il doit avoir une forte connaissance des données graphiques de rendu et d'animation, et de leurs impératifs d'optimisation sur plusieurs plates-formes. Les middlewares graphiques doivent lui être familiers. Un rôle qui déborde largement le simple " codeur ".

Programmeur moteur physique

Il gère la restitution réaliste des effets phy-

siques dans le jeu, gravité, trajectoires, répercussions des collisions, etc. C'est une véritable spécialité, très recherchée, exigeant d'être pointu en mathématique, physique, mécanique. Ce programmeur est souvent chargé " d'améliorer " un moteur existant pour l'adapter au jeu.

Programmeur Intelligence Artificielle

Un rôle devenu indispensable aujourd'hui (à l'exception des Casual Games) et une spécialité reconnue. Compétences en algorithmes de trajets, d'expérimentation et d'apprentissage, probabilités, anticipation de décisions. Très recherché aussi.

Programmeur architecture

Proche du chef de projet, c'est le constructeur et l'organisateur du flux global du développement – définition des objets, directions d'optimisation, conformité des développements parallèles, coïncidences des technologies mises en œuvre.

Programmeur optimisation

C'est le plus souvent un spécialiste du hardware, au fait des impératifs techniques des plates-formes du jeu. Il gère l'optimisation du graphisme, du son, de la mémoire et de l'utilisation du processeur pour chacune.

Programmeur outils

A lui de réaliser, presque à la demande, les outils qui peuvent augmenter et optimiser la production des diverses parties du développement. Le plus souvent, des éditeurs de niveaux, des outils de gestion des données, des testeurs automatisés de gameplay, etc. Un généraliste avec de fortes capacités d'adaptation. Essentiel.

Programmeur interface

S'il ne définit pas toujours l'interface ou son ergonomie, il code l'intégration des menus, écrans de commande, icônes, messages d'information, etc. dans le cours même du jeu. Il peut aussi gérer la répercussion à l'affichage des données ludiques (points de vie, nombre d'unités, objets possédés, etc.). Réclame beaucoup d'organisation, et rôle assez ingrat.

Programmeur système / data

Il a en charge la programmation des interfaces, qu'il s'agisse de commandes (clavier, manette…) ou du système de restauration et de sauvegarde des données en début, cours, ou fin de jeu. Programmation de bas niveau.

Programmeur debug

Le plus souvent, il traque simultanément les bugs purement techniques et ceux qui relèvent du gameplay – c'est d'autant plus normal qu'ils sont souvent liés. Un rôle transversal au sein de l'équipe, très exigeant en compétences ludiques, software et hardware, d'autant qu'on exige de lui plus que de simples " rapports de bugs " mais souvent des diagnostics suivis de propositions de solution.

Au-delà de ces profils, on trouve bien entendu tous les développeurs de " mise en scène ", graphistes, animateurs, spécialistes d'effets spéciaux, bruiteurs, musiciens, sound designers, etc. L'appréciation de leur aptitude au poste se fait cependant davantage sur la coïncidence de leur art avec leurs compétences numériques que sur leur pure science du code.

Combien ça gagne ?

Définir une grille de salaires est difficile, vu la diversité de tâches et de responsabilités que couvre l'appellation de " programmeur " dans les jeux vidéo. Il est donc seulement possible de définir une fourchette, qui va de 16 000 à 45 000 Euros par an, la moyenne tournant ainsi autour de 25 000 Euros. Cette estimation ignore cependant les interventions ponctuelles, sortes de piges spécialisées sur un point précis du jeu et du code, intégration de données spécifiques à une plate-forme, gestion d'un seul aspect des effets ou d'un secteur de l'I.A., etc. Elle oublie aussi les postes de développeur de jeu à l'étranger, de plus en plus fréquents. Ainsi, il se raconte que si vous allez travailler à Shanghai pour un grand éditeur international, vous toucherez à peu près 1 200 Euros/ mois pour un poste plutôt élevé. Mais il faut relativiser : ce type de salaire est important par rapport au coût de la vie local, et il s'y ajoutent généralement le logement et deux allers-retours pour la France / an à la charge de l'employeur.

Casual et Serious Gaming

Deux secteurs de développement vidéo ludiques émergent plus spécialement en ce moment. En terme de marché et de chiffre d'affaires (très) porteur, les jeux pour téléphones mobiles : ces petits jeux appartiennent à la catégorie du Casual Gaming (jeu occasionnel). Des principes simples, donc, et une programmation majoritairement 2D. L'embauche de programmeurs y est très active depuis mi-2006. Un mix de compétences C++, Flash, Java est nécessaire.

Second secteur porteur, le jeu éducatif ou promotionnel pour institutions, le Serious Gaming. Des principes simples encore, mais qui doivent se plier à la demande du client. Le support est le plus souvent web, donc Flash et Java sont avant tout requis. Cependant, la déclinaison du Serious Game sous forme d'application téléchargeable ou de CD/DVD-promo est de plus en plus fréquente, revenant vers une programmation plus " classique ".

Le contexte économique

Pour 2007, le chiffre d'affaires du jeu vidéo est estimé à 20 milliards d'Euros pour le monde, dont 2 milliards environ pour la France. Les dernières statistiques indiquent 330 entreprises françaises directement impliquées dans le secteur, offrant entre 2000 et 2500 emplois à temps plein, toutes spécialités confondues. Une " moyenne " de 8 emplois par acteur : il s'agit donc en grande majorité de petites ou de très petites entreprises, cette moyenne devant en plus être relativisée par l'apport des géants, style Ubisoft… Selon l'AFJV (Association Française du Jeu Vidéo), la répartition de ces entreprises, en janvier 2007, serait de :

- 114 studios de développement
- 98 prestataires
- 64 éditeurs
- 28 producteurs de middlewares
- 23 distributeurs et spécialistes hardware

On note aussi une forte mobilité du secteur, pour les entreprises (le nombre de start-up créées sur une innovation technologique particulière a augmenté de 55% depuis 2005), comme pour les salariés.

amen
IN WEB WE TRUST

serveurs dédiés DUO

Vous n'avez pas à nous prier pour vous offrir deux fois plus de performance !

NOUVEAU
Serveurs dédiés DUO

AMD 64 Opteron™

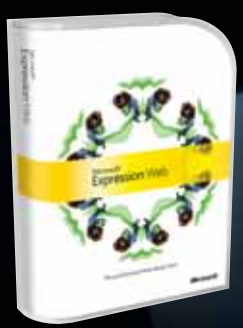
Pour les professionnels les plus exigeants, AMEN lance la nouvelle gamme de serveurs dédiés DUO basée sur des processeurs double coeur, disques durs en RAID, pour vous offrir 2 fois plus de puissance.

DUO 1000 ► 99 € ht/mois*
(118,40 € ttc/mois*)
AMD Opteron 1210 - 2x1,8GHz - RAM 1GB
Disque dur 2x160GB - Raid Soft
2 adresses IP - Interface Plesk 8 jusqu'à 100 domaines - Trafic illimité

DUO 2000 ► 149 € ht/mois*
(178,20 € ttc/mois*)
AMD Opteron 1212 - 2x2,0GHz - RAM 2GB
Disque dur 2x200GB - Raid 1 matériel
4 adresses IP - Interface Plesk 8 jusqu'à 300 domaines - Trafic illimité

DUO 4000 ► 199 € ht/mois*
(238,00 € ttc/mois*)
AMD Opteron 1214 - 2x2,2GHz - RAM 4GB
Disque dur 2x250GB - Raid 1 matériel
6 adresses IP - Interface Plesk 8 jusqu'à 300 domaines - Trafic illimité

Compatibles &



OFFERT ! **

Microsoft Expression Web vous offre tous les outils dont vous avez besoin pour créer des sites web dynamiques de qualité professionnelle : création CSS, prise en charge XML, ASP.NET 2.0, DHTML, XHTML, CSS, Javascript…

Nous avons foi en un idéal de services, surtout lorsqu'il vous permet de bénéficier des dernières avancées techniques : architecture réseau redondée, bande passante dédiée 2GB, haute disponibilité (99,9%), assistance technique par mail et téléphone 6j/7[1]. Quant à notre 'Garantie satisfait ou remboursé'[2], elle vous permettra d'atteindre la sérénité absolue. **Si vous croyez au web, vous croirez en nous.**

► Pour plus de renseignements **0 892 55 66 77** (0,34€ / min) ou **www.amen.fr**

AMEN RCS PARIS : B 421 527 797. IN WEB WE TRUST : Nous croyons au web. Voir conditions Générales de Vente sur www.amen.fr. *Prix au 01/01/2007. Tous ces tarifs sont concédés pour un engagement annuel.
[1] Du lundi au samedi de 9h à 18h au 0899 70 9001 (1,34 € l'appel puis 0,34 €/mn). [2] Garantie satisfait ou remboursé sous 10 jours. AMD, le logo AMD opteron et ses déclinaisons sont des marques déposées de Advanced Micro Devices Inc.
** Offre d'une valeur de 412,62 € TTC, réservée aux 300 premiers clients ayant souscrit sur une période de 6 mois minimum à un serveur dédié DUO avec système d'exploitation Windows.

" Infragistics prend soin de l'interface utilisateur, donc je peux me concentrer sur **l'aspect métier.** "

-Application Developer

Office 2007 UI for Windows Forms

NetAdvantage® for ASP.NET Charts and Gauges

Une experience utilisateur multi-plateformes consistante

NetAdvantage for .NET 2007 Vol. 2

La boite à outils exceptionnelle pour la conception et le développement de vos interfaces utilisateur.

Windows Forms

Une expérience utilisateur sans failles - Développez au-delà de vos applications avec les caractéristiques d'intégration, de collaboration et de notification. Offrez la même expérience utilisateur se trouvant dans Office 2007 et Windows Windows Vista™

ASP.NET

Améliorez la visibilité de votre site internet - Améliorez le score du moteur de recherche de vos sites, grâce aux contrôles ASP.NET, avec manipulation profilée des requêtes du web crawler

Charting

Tableaux de bord professionnels - Nouveau! Gauges pour ASP.NET et Windows Forms en un instant fournissant une compréhension claire des Key Performance Indicators

Application Styling

Design Once, Style Everywhere - pour vos applications Windows Forms et ASP.NET grâce à NetAdvantage. Créez rapidement et appliquez vos normes à travers les contrôles ou applications

Pour de plus amples informations:
infragistics.com/dotnet

sales-europe@infragistics.com

N° Vert 0800 667 307

Infragistics®
Powering The Presentation Layer

WINDOWS FORMS ASP.NET WPF JSF

grids scheduling charting toolbars navigation menus listbars trees tabs explorer bars editors

Copyright 1996-2007 Infragistics, Inc. All rights reserved. Infragistics, the Infragistics logo and NetAdvantage are registered trademarks of Infragistics, Inc. Application Styling is a trademark of Infragistics, Inc. All other trademarks or registered trademarks are the respective property of their owners.