

Sécurité

Prévenir les failles de sécurité
dans votre code

Blinder votre site web

C++

Langage d'avenir
et ultra performant !

Le développement
multiplateforme
a-t-il un avenir ?

*Framework REST, Mono,
Flash Platform : les nouveaux
outils du multiplateforme !*

© istockphoto / mura



Emploi

Recruteurs
recherchent
désespérément
experts **.Net**

Smartphone



Maîtriser l'ergonomie
sous Windows Phone 7

Linux

L'instrumentation sous Linux

.Net

Programmation asynchrone :
la nouvelle bombe Microsoft !

Java

Utiliser la programmation par
contraintes

Maps

Utiliser Google Maps v2 et v3

M 04319 - 140 - F: 5,95 €



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

N°1 EN FRANCE

- Crée des applications Windows, Linux, Mac, ... compatibles Internet, Intranet, Android, Windows Phone 7
- Gestion complète du cycle de vie
- Éditeur d'analyses (UML, Merise...)
- RAD (avec patterns)
- Lien avec toutes les bases de données (ODBC, OLE DB, Lien natif), Oracle, SQL Server, AS/400, Informix, DB2, MySQL, PostgreSQL, SQLite, ...
- HyperFileSQL: puissante base de données Client/Serveur gratuite incluse
- Générateur d'états PDF, Codesbarres
- Accès natif SAP R/3, Lotus Notes, ...
- Gestion de planning, des Exigences
- Audit automatique vos applications
- Langage de 5^e génération (LSG)
- SNMP, Bluetooth, TAPI, OPC, FTP, HTTP, Socket, Twain, API, DLL, XML...
- Domotique • Liaisons série et USB
- Débogage à distance, Profiler
- Multilingue (64 langues)
- Installateur I-clic & Push
- Gestionnaire de versions
- Gestion de l'Infrastructure
- Tout en français
- etc...

support technique gratuit

WINDEV®

16

CRÉEZ VOS APPLICATIONS POUR PC, MAC, LINUX, INTERNET, SMARTPHONES, TABLETTES, ...

VOTRE CODE EST MULTI-PLATEFORMES

Windows, .Net, Java, PHP, Linux, Mac, J2EE, XML, Internet, SaaS, Pocket PC, Windows Phone 7, Android, ...

WINDEV GÈRE LE CYCLE DE VIE COMPLET DE VOS APPLICATIONS.

DÉVELOPPEZ

CONCENTREZ-VOUS SUR LES BESOINS MÉTIER

Elu «Langage le plus productif du marché»

VERSION EXPRESS GRATUITE
Téléchargez-la !

100 TÉMOIGNAGES SUR SIMPLE DEMANDE



► **DEMANDEZ LE DOSSIER GRATUIT**

Dossier gratuit 260 pages sur simple demande. Tél: 04.67.032.032 info@pcsoft.fr

Fournisseur Officiel de la Préparation Olympique



www.pcsoft.fr

sommaire

\\ actus

En bref	6
Agenda	8
Hardware	12

\\ sgbd

Réplication de fusion avec SQL Server Compact	13
---	----

\\ webmaster

Comment sécuriser son site web ?

Faibles et offuscation	19
La sécurité par les bonnes pratiques	22
Protéger son site web avec PHP	24
Sécuriser un site ASP.Net.....	27

\\ gros plan

C++ plus que jamais d'actualité

Le C++ est-il redevenu le langage incontournable ?	31
Le C++ au quotidien	32
Comparaisons d'applications équivalentes C# et C++	34
C++ est-il en train de redevenir LE langage	38

\\ dossier

Le multiplateforme : nouvelle génération ou illusion ?

Pourquoi et comment le multiplateforme ?	41
Restlet et le multiplateforme.....	42
ASP.Net multiplateforme : Mono prêt pour la production !	45
Adobe Flash Platform et les applications multiscreen.....	49

\\ carrière

Retour sur les TechDays 2011 : les recrutements repartent très fort ! ..	52
--	----

\\ code

Analyse de l'ergonomie sur Windows Phone 7	55
IIS Express : le nouveau serveur web pour le développeur.....	59
S'inspirer d'Atom et de Google pour créer des services Web Restful	61
Créer une carte avec des marqueurs, grâce à Google Maps API version 2 et 3 et Dojo Toolkit	64
Introduction à la programmation par contraintes en Java	68
Instrumentalisez vos Linux	72
Async : une nouvelle approche de la programmation Asynchrone sous .NET	76

\\ temps libre

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



9



18



30



52



76

L'info continue sur www.programmez.com

CODE

Les sources des articles

NOUVEAU

Livres blancs : langages, outils...

TÉLÉCHARGEMENT

Les dernières versions de vos outils préférés + les mises à jour

QUOTIDIEN

Actualité, Forum Tutoriels, etc.

DESIGN

Concevez des applications qui vous aident à diriger votre business



Notre contrôle™ xamMap dans Silverlight et WPF vous permet de tracer toutes les données géospatiales dont vous avez besoin, comme ce plan de cabine d'avion vous permettant de gérer votre business ! Venez l'essayer dès aujourd'hui sur infragistics.com



NetAdvantage® **ULTIMATE**

for ASP.NET, Windows Forms, WPF, Silverlight, WPF Data Visualization, Silverlight Data Visualization

Infragistics®

Infragistics Ventes France N° Vert
0800 667 307 • Infragistics Europe Ventes
+44 (0) 800 298 9055 • [@infragistics](http://t@infragistics)



Aidons nos amis les virus

Début mars, une affaire éclate, plusieurs dizaines (centaines ?) d'ordinateurs du ministère des finances de Bercy sont attaqués, hackés. L'attaque est désarmante de simplicité : un simple mail portant un cheval de Troie. Et une fois présent sur la machine, le code malicieux fait le reste. Il ne faut pas croire que les attaques informatiques soient toujours d'un perfectionnement, d'une perversion technique élevés. Mais elles démontrent les carences des procédures, du manque de sensibilisation des personnels. Comment se fait-il que les Troyens n'aient pas été détectés et bloqués ? Pourquoi des utilisateurs n'ont-ils pas été plus prudents par rapport à des mails suspects ? Une en-tête de mail aguichante suffit souvent à convaincre l'utilisateur.

N'oublions jamais qu'un large pourcentage des ordinateurs infectés, des usurpations d'identités, des vols de données... est dû à la simple négligence, au manque de rigueur dans l'observation des règles de sécurité de la part des utilisateurs. Soit par manque d'informations du service informatique et de mise en place d'outils adaptés, soit par simple déni des règles internes. Encore trop souvent, les entreprises oublient les réflexes basiques de la sécurité informatique. Les artisans, TPE, PME n'ont pas de service informatique (ou peu), ni de responsable sécurité (notre ami le RSSI), ne parlons même pas de la situation à la maison...

Mais la sécurité informatique passe aussi par les applications, donc par les développeurs. Sur ce point, je suis parfois découragé devant le manque de rigueur, voire l'absence quasi totale des mesures élémentaires à prendre dans une application web ou desktop. Le développeur doit absolument et impérativement blinder son code ! La surface potentielle d'attaque sera alors réduite, rendant (un peu) plus difficile le hack. Pourtant aujourd'hui, nous disposons de bonnes pratiques, de métriques, d'outils, des méthodologies de développement sécurisé. Et combien d'applications, de sites 'tombent' par un vulgaire dépassement de capacité ou une injection de code ? C'est le b-a-ba de la sécurité et de la programmation. Même si j'avoue, j'ai parfois eu des manquements sur ce point quand je codais. Manque de temps, ou tout simplement, pas envie de faire quelques lignes de code supplémentaires...

Plus aucune application ne devrait être déployée sans un niveau acceptable de sécurité. Même les éditeurs ont du mal à combler les trous : PDF, Flash, Windows, Linux, MacOS X, tout le monde est vulnérable. Evitons de donner le bâton pour se faire battre.

Le développeur doit être sensibilisé, formé à la sécurité. Il doit comprendre comment fonctionnent les principaux risques et de quelle manière mettre en place les contre-mesures. Il doit pouvoir établir une matrice de surface d'attaque pour comprendre les règles de sécurité à implémenter. Certains codes nécessitent une rigueur de travail. Le parallélisme, l'asynchronisme, l'embarqué exigent une structure de programmation propre car ces modèles de développement n'aiment pas l'approximation.

Facilitons la vie des gentils pirates, donnons-leur les codes sources... Merci pour eux !

■ François Tonic

Editeur : Go-02 sarl, 21 rue de Fécamp 75012 Paris - diff@programmez.com.

Rédaction : redaction@programmez.com

Directeur de la Rédaction : Jean Kaminsky.

Rédacteur en Chef : François Tonic - ftonic@programmez.com. **Ont collaboré à ce numéro :** F. Mazué, S. Saurel, S. Nabih. **Experts :** P. Cambier, F. Queudret, C. Villeneuve, T. Laurent, E. Vernié, M. Foucault, C. Pichaud, T. Templier, J. Louvel, T. Boileau, A. Labarre, J. Ph. Gouigoux, M. Chaize, J. Dollon, T. Vidal, N. Esprit, T. Sorgnard, V. Lecomte,

Illustrations couverture : ©istockphoto/hjroy et istockphoto/mura

Publicité : Régie publicitaire, K-Now sarl. Pour la publicité uniquement : Tél. : 01 41 77 16 03 - diff@programmez.com.

Dépôt légal : à parution - Commission paritaire : 0712K78366 ISSN : 1627-0908. Imprimeur : S.A. Corelio Nevada Printing, 30 allée de la recherche, 1070 Bruxelles Belgique. **Directeur de la publication :** J-C Vaudecrane

Abonnement : Programmez 22, rue René Boulanger, 75472 Paris Cedex 10
Tél. : 01 55 56 70 55

abonnements.programmez@groupe-gli.com
Fax : 01 55 56 70 91 - du lundi au jeudi de 9h30 à 12h30 et de 13h30 à 17h00, le vendredi de 9h00 à 12h00 et de 14h00 à 16h30. **Tarifs**
abonnement (magazine seul) : 1 an - 11 numéros France métropolitaine : 49 € - Etudiant : 39 € - CEE et Suisse : 55,82 € - Algérie, Maroc, Tunisie : 59,89 € - Canada : 68,36 € - Tom : 83,65 € - Dom : 66,82 € - Autres pays : nous consulter. **PDF :** 30 € (Monde Entier) souscription exclusivement sur www.programmez.com

L'INFO PERMANENTE
WWW.PROGRAMMEZ.COM



PROCHAIN NUMÉRO
N°141 mai 2011
parution 30 avril

✓ Dossier open source

Open Innovation, Open Data, Open Cloud : les nouveaux défis du monde du libre !

✓ Sharepoint

Pourquoi déployer Sharepoint ?
Les outils pour les développeurs.
Comprendre l'architecture Sharepoint

✓ Carrière

Les profils open source.
Enquête : les formations

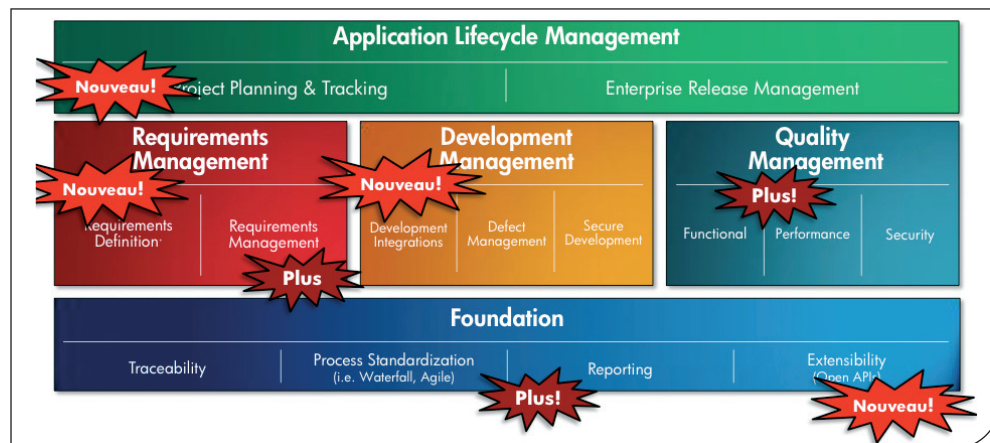
■ **Modeliosoft** annonce la sortie de Modelio SysML Designer. SysML est un standard reconnu dans l'ingénierie des grands systèmes. Il est utilisé pour spécifier, analyser, vérifier et concevoir les systèmes complexes, tels que des avions, des centres de contrôles aériens, des systèmes d'armes, de grands systèmes techniques, etc. L'outil est disponible en open source.

■ **Micro Focus** met à jour sa gamme d'outils de tests : Silk-Central Test Manager R2 inclut un nouveau principe de définition et de suivi des tests pouvant être menés sur des configurations multiples. La nouvelle version de l'automate SilkPerformer inclut des fonctions avancées de test de montée en charge pour les applications AJAX et supporte dorénavant les environnements Microsoft Visual Studio 2010, Citrix Online plug-in 12, Citrix Web Interface 5.3., Flex 4 et OpenSSL 1.0.0.a.

■ **Windows Azure SDK 1.4** est disponible. Cette version corrige plusieurs bugs dans le support ISS, améliore la stabilité sur les plateformes x86, permet une meilleure gestion des erreurs dans le web.config. La fonction Remote Desktop dans les Roles Azure a été améliorée et devient plus stable.

■ **Adobe** rend disponible une pré-version de son outil de conversion Flash vers HTML 5 : Wallaby. Wallaby est une application AIR qui permet aux designers et développeurs de convertir les fichiers Adobe Flash Professionnel (FLA) en HTML5 grâce à un simple glisser-déposer – augmentant rapidement et facilement la distribution de contenu créatif sur les plates-formes, y compris les terminaux iOS comme l'iPad et l'iPhone.

HP booste son ALM avec ALM 11



Dans un marché de l'ALM en pleine ébullition, notamment du côté Microsoft avec TFS 2010, HP veut reprendre position avec ALM 11 et Quality Center 11. Il s'agit d'une version majeure pour l'éditeur qui cherche à consolider ses positions sur les outils de qualité et de tests. ALM 11 repose sur trois idées clés : communication et collaboration, accélération des tests et de leur exécution, planification / suivi / rapports. Pour accomplir tout cela, ALM 11 introduit de nombreuses nouveautés et améliorations. Les nouveautés se concentrent sur la définition des exigences, la gestion du développement, la planification et le tracking ALM, sans oublier la couche bas niveau, une architecture plus extensible. La partie qualité a été améliorée ainsi que les rapports. Une des nécessités dans un ALM est la méthode de planification et de son suivi. ALM 11 veut donner la visibilité aux différentes strates projet, identifier les points de blocage, faire une mise à jour constante entre le planning réel et le plan initial du projet, localiser les dérives éventuelles. C'est pour cela que ALM 11 surveille de près les KPI (KPI = indice de

performance) du projet, de bout en bout. On pourra maintenant définir la release, les jalons projet, visualiser les seuils d'alerte, définir les KPI sur chaque jalon, etc. Et l'outil de planification suivra en temps réel les évolutions. C'est un tableau de bord indispensable pour les chefs de projet et la direction.

La partie exigence concerne ALM et Quality Center Enterprise 11. Une bonne gestion des exigences repose sur 4 piliers : capture, collaboration, traçabilité et standardisation (pour que les exigences aient toutes les mêmes caractéristiques). Un des défis est de pouvoir modéliser les processus métiers. HP supporte BPMN et est donc capable de récupérer, d'échanger avec des outils tiers utilisant le même format. Pour saisir les exigences, même si Word et Excel demeurent largement utilisés, HP introduit un éditeur de texte riche s'intégrant parfaitement à la plateforme. Le métier peut ainsi travailler aussi bien dans Quality Center que dans Word. L'interface facilite la saisie et l'intégration des exigences (dans un format uniforme pour éviter des problèmes d'interprétation). Et différents types

d'exigences peuvent être créés. Ces templates proposent par défaut des modèles de saisie, un peu comme les Work Items de TFS 2010. A cela se rajoute la traçabilité de l'exigence : qui fait quoi, est-ce fait, les tests ont-ils été effectués, par qui, etc. La possibilité de tracer les éléments projet permet de remonter aux origines d'un problème.

Tester avec Sprinter

Les tests constituent un autre gros morceau de ALM 11 et les utilisateurs voulaient du nouveau ! Désormais, un script est possible pour toutes les configurations des tests. Sur-tout, HP introduit Sprinter dédié aux tests manuels. L'outil se connecte à Quality Center et est inclus par défaut dans ALM 11. Le but est de réaliser les tests manuels plus rapidement et sans intrusion par rapport aux codes. On peut annoter directement à l'écran pour aider le développeur, on peut rejouer le test. Il est possible d'injecter des données pour réaliser les tests manuels. Objectif : réduire le temps nécessaire ! Et Sprinter permet des tests sur de multiples environnements.

**Codejock Xtreme Chart Pro ActiveX** à partir de € 167

Incluez des diagrammes dans vos applications ActiveX en quelques lignes de code.

- Affiche un ensemble riche de classes de personnalisation et d'amélioration
- Histogrammes, barres de dispersion, barres empilées, 100% barres empilées et barres horizontales
- Graphiques en secteurs 2D/3D, anneaux 2D/3D et tores 3D, ainsi que les secteurs éclatés
- Inclut les lignes standard, à dispersion, à traçage rapide, en escalier et spline
- Inclut aussi les diagrammes de points, de zones, entonnoir, financiers et de Gantt

**Spread for Windows Forms** à partir de € 731

Feuille de calcul pour les applications Windows Forms, compatible avec Microsoft Excel.

- Accélérez le développement avec les concepteurs de feuilles de calcul, l'Assistant de prise en main et les concepteurs de graphiques
- Renseignement automatique : anticipation de la frappe dans la cellule
- Nouveau - outil intégré de création de diagrammes avec 85 styles
- Nouveau - préserve les .XLS et restaure les fonctions non supportées
- Inclut des apparences prédéfinies ainsi que la possibilité de créer des apparences personnalisées

**QuickBooks Data Provider for ADO.NET** à partir de € 292

Applications liées à QuickBooks avec les Assistants Visual Studio.

- Cache intelligent pour une couche réactive d'accès aux données
- Support lecture/écriture (CRUD) avec une interface standard ADO.NET
- Fonctionnalités de sécurité de classe entreprise dont chiffrement TLS/SSL
- Donne aux développeurs le pouvoir de récupérer facilement les données de QuickBooks et QuickBooks Online, inclut clients, transactions, factures, reçus de ventes, rapports et plus

**FusionCharts** à partir de € 146

Graphiques Flash & JavaScript (HTML5) interactifs pour les applications Web.

- Animez vos applications Web avec des graphiques interactifs et pilotés par les données
- Créez des graphiques AJAX avec des possibilités d'exploration en quelques minutes
- Exportez les graphiques au PDF et les données en CSV directement depuis les graphiques
- Créez des jauges, des tableaux de bord, des graphiques financiers et plus de 550 types de carte
- Adopté par plus de 17 000 clients et 330 000 utilisateurs dans 110 pays

■ L'éditeur français **eXo** annonce une série de services dans le cloud : cloud service, Cloud IDE. Ce dernier est un service multi-tenant, la création de nouveaux réseaux est instantanée. Les entreprises choisissent un domaine, y invitent leurs développeurs et ils peuvent commencer à coder tout de suite avec leur propre IDE.

■ **Ingres** dévoile une stratégie cloud avec notamment SkySafe. SkySafe est un service dédié aux entreprises et basé sur le cloud, qui leur permet de développer et d'exécuter des applications en toute sécurité sans supporter les coûts administratifs traditionnels. Il propose un stockage flexible, et une sécurité de haut niveau.

■ **Xen.org** dévoile Xen Cloud Platform. Il s'agit d'une infrastructure de type IaaS avec l'hyperviseur Xen et des piles logicielles adaptées au monde cloud : prise en charge de la gestion réseau, du stockage, implémentation d'outils provenant d'OpenStack, interaction avec OpenStack Bexar. Mécanisme d'orchestration et d'automatisation. Le réseau virtuel se gère via Open vSwitch. Le projet dérive de XenServer et est disponible en licence open source GPL 2.

■ **Mageia**, fork de Mandriva, se prépare activement au lancement de sa première version stable. La v1 devrait sortir début juin après 6 mois de développement interne. Les premières alpha ont été dévoilées aux développeurs en début 2011. La version alpha 2 doit intégrer le logo, l'écran de démarrage, l'ensemble des packages, le cadre défini de l'ensemble du système. La distribution sera disponible en 32 et 64 bits. Pour en savoir plus : <http://blog.mageia.org/fr/>

Des applications mobiles illégales à 71 % ?

OpenLogic, éditeur oeuvrant dans l'univers open source et spécialiste de la conformité aux licences ouvertes, a annoncé 635 applications mobiles sur Android et iOS. Pour OpenLogic, plus de 70 % des applications utilisent du code, des composants open source. Mais là, où le bât blesse c'est que ces applications violent les clauses contractuelles des licences ouvertes : c'est-à-dire la disponibilité du code source, la présence des fichiers de ladite licence.

Les principales licences sont Apache, GPL et LGPL. Et chacune possède des clauses spécifiques. Là où la situation se



complique, c'est que selon la Free Software Foundation, la GPL et la licence Apple iTunes sont incompatibles. Et Apple

aurait déjà rejeté ou retiré plusieurs applications mobiles sous GPL à cause de cela. Un des derniers exemples (même si les causes exactes restent opaques) est le lecteur multi-média VLC.

Mais Android ne serait pas mieux loti. Cependant, OpenLogic note tout de même que 29 % des applications ayant des éléments GPL / LGPL respectent les obligations.

Pour OpenLogic, ces problèmes viennent surtout d'un manque de connaissance des développeurs sur les licences et les obligations qu'elles induisent.

Oracle s'intéresse (aussi) au mobile

Oracle Application Development Framework Mobile Client est désormais disponible. Oracle ADF Mobile Client est un composant d'Oracle Fusion Middleware qui permet aux développeurs d'étendre leurs compétences existantes au domaine des terminaux mobiles. Il simplifie la création et le déploiement des applications en offrant un environnement Java unique et standardisé pour le développement des interfaces utilisateur, ainsi que des outils adaptés à tous les terminaux supportés. Les développeurs réalisent une seule fois leur application puis la déploient sur plusieurs terminaux. Et lorsque le support de nouveaux terminaux est ajouté à Oracle ADF

Mobile Client, ces applications peuvent être déployées sur ces nouvelles plates-formes sans nouveau développement. D'autre part, Oracle annonce Oracle Application Development Framework (ADF) Mobile Client, une extension de l'environnement de développement applicatif Oracle Application Development Framework. Cette solution s'appuie pleinement sur Java et JSF, gère le connecté / déconnecté pour les données. Enfin, Oracle ADF Mobile Client fournit des fonctionnalités mobiles natives, et permet ainsi de déployer les applications directement sur les terminaux mobiles, les premiers environnements supportés étant BlackBerry et Windows Mobile.

agenda \

AVRIL

- Le 05 avril 2011, Paris, Quai de Grenelle, **Journée française du test logiciel 2011**. <http://www.cftl.net>
- Le 05 avril 2011, Lyon, locaux de l'E.P.I.T.E.C.H., **Conférence Mix-it 2011** : Java/JEE, Agilité, et Tendances novatrices <http://www.mix-it.fr/sessions>
- Le 07 avril 2011, Webinar : **Msdn et le cycle de vie de l'application** : Développer une application moderne est complexe. http://comsoft.origin.com/visual_avril.html

- Du 7 au 9 avril 2011, Paris, ESIEA (Ecole Supérieure d'Informatique) 9, rue Vésale, 75005. Conférence HES 2011. Hackito ergo sum (Attention, il n'y a que 215 places !). <http://hackitoergosum.org>

MAI

- Du 10 au 12 mai 2011, CNIT de Paris - La Défense **Solutions Linux/Open Source 2011** <http://www.solutionslinux.fr>
- Le 10 mai 2011, Paris 3e Espace St Martin, 3e édition des **GS Days, Journées Francophones de la Sécurité**. <http://www.globalsecuritymag.fr>

- Du 26 au 27 Mai 2011, Grand Rex : Paris, **What's Next : quel avenir pour Java ?** Conférences organisées par Zenica. <http://whatsnextparis.com>

ETRANGER

- Le 08 avril Belgique, Bruxelles, **DotNetHub : Programmation Orientée Aspect avec PostSharp** www.dotnethub.be/DnhEventManager/EventDetail.aspx?eventId=18&paged=1&mid=1
- Du 11 au 14 mai, Allemagne, Berlin : **Linux Tag 2011** <http://www.linuxtag.org/2011/>

Web

SensioLabs : sortir au plus vite la version 2 de Symfony

Les 3 et 4 mars se déroulait à Paris, la grand' messe annuelle de la communauté Symfony, le framework PHP développé par SensioLabs : Symfony Live. En réalité, l'événement se déroula sur quatre jours : une première journée uniquement d'ateliers techniques, deux journées de conférences et enfin une dernière intitulée « hacking day » pour discuter du cœur même du framework, trouver des solutions à des problèmes, etc.

Et le succès a été au rendez-vous : plus de 500 personnes, contre 400 pour l'édition 2010. Beaucoup de développeurs s'étaient déplacés pour rencontrer les développeurs du projet et discuter avec d'autres utilisateurs. Le Symfony Live est aussi un immense réseau social.

Pour SensioLabs, l'événement est crucial pour rencontrer la communauté qui utilise tous les jours Symfony, mieux comprendre

les attentes, faire remonter des idées, des problèmes. Et les participants ont été comblés avec un grand nombre de sessions techniques pour pousser les fonctions du framework. Côté inattendu du Symfony Live, le marché de l'emploi. Car aujourd'hui, trouver des compétences Symfony n'est guère facile. Ce type d'événement est un bon vecteur pour dénicher un travail ou trouver les bons candidats.

Un des enseignements de ce Live est la volonté de SensioLabs de rapidement sortir la version finale de Symfony 2 qui tarde à venir. La communauté a envoyé de nombreuses contributions qui ont participé de ce retard car plusieurs propositions ont été intégrées, parfois dans le cœur même de l'outil. Il s'agit aussi de diffuser l'usage du framework, d'acquiescer de nouvelles belles



références. Symfony est clairement en concurrence frontale avec Zend. La v2 est actuellement disponible en pré-version et se veut la version de la maturité : plus stable, moins consommatrice de mémoire vive, plus flexible, réécriture (ou presque) du framework. L'approche MVC a été renforcée et améliorée.

Symfony 2 promet d'être plus pragmatique que la v1, par exemple, la présence d'injection de dépendances. L'architecture plus flexible permet d'introduire de multiples bundles, bref des plug-ins.

Nous reviendrons en profondeur sur le framework lors de sa disponibilité.



WHAT'S NEXT ?

LA conférence technique sur l'avenir de Java
26-27 Mai 2011, Le Grand Rex, Paris

Un événement exceptionnel à Paris !

Les meilleurs experts rassemblés pour vous

Découvrez les technologies & acteurs de demain...

www.whatsnextparis.com



1&1 HÉBERGEMENT

J'ai choisi 1&1 car ils bénéficient de serveurs d'une grande stabilité, de coûts abordables, d'une gamme de services très complète et d'une bonne réactivité en hotline.

Sylvain Jacquet
www.lemagasinbio.com



VOTRE SITE TOUT COMPRIS GRATUIT PENDANT UN AN !*



*Le Pack Initial est gratuit pendant 1 an sous réserve d'un engagement de 24 mois. Frais de mise en service de 5,97 € TTC. À l'issue de la première année, le Pack Initial est au prix habituel de 2,38 € TTC/mois. Offre domaine applicable uniquement la première année au lieu du prix habituel de 6,99 € HT/an (8,36 € TTC). Conditions détaillées sur www.1and1.fr. Offres sans engagement également disponibles.



1&1 PACK INITIAL
1 AN
GRATUIT*
OFFRE PROLONGÉE !

1&1, des packs hébergement tout compris au meilleur prix !

.com
.fr
.org
.net

Nom de domaine inclus

Profitez d'un nom de domaine inclus pendant toute la durée de votre pack, sans coût supplémentaire.



Trafic illimité

Restez à l'abri des mauvaises surprises sur votre facture grâce au trafic illimité.



Logiciels avancés de web design

Concevez votre site comme un pro et optimisez-le pour la consultation sur terminaux mobiles.



Fonctions pour développeurs

Disposez de toutes les fonctions nécessaires pour développer votre site et vos applications dans votre environnement favori.



Outils de marketing en ligne

Propulsez votre site en tête des moteurs de recherche à l'aide des outils de référencement 1&1.



Centres de données verts

Réduisez votre impact sur l'environnement en optant pour un hébergement 1&1.

1&1 PACK INITIAL :

- Nom de domaine inclus
- 2,5 Go d'espace disque
- Trafic ILLIMITÉ
- 10 comptes email (POP3 et IMAP)
- Accès FTP
- Base de données MySQL
- e-Boutique Start
- Outils de référencement
- Assistance technique par email et téléphone
- Et bien plus encore...

0€* ~~1,99€~~
 pendant un an (2,38 € TTC/an)

Retrouvez toutes nos offres du moment sur **1and1.fr**, comme par exemple le nom de domaine en **.fr à seulement 3,99 € HT/an (4,77 € TTC/an)* !**

1&1

Appelez le **0970 808 911** (appel non surtaxé) ou consultez notre site Web

www.1and1.fr

■ Un clavier à ranger dans sa poche et résistant aux liquides et taches ? **Mobility Lab** propose un clavier flexible en connexion USB. Il ne craint ni l'eau, le café, les miettes et se range facilement dans un sac ! Prix public : 19,99 €.

■ **Intel** dévoilait début mars ses nouveaux disques SSD : série 510 avec interface SATA 6 Gb/ seconde. Ils embarquent une puce Core. Les débits peuvent atteindre 500 Mo/s. Outre une meilleure réactivité globale de la configuration, toujours par rapport à un disque dur mécanique, ce type de stockage est dépourvu de toute pièce en mouvement et est donc plus résistant, consomme moins d'énergie et son fonctionnement est plus silencieux.

■ **H&B** propose une nouvelle mini-chaîne numérique capable de supporter clé USB, iPod, MP4, iPhone, etc. cette mini chaîne HF-430i possède une montre digitale et quartz qui fait office d'alarme ! Ainsi, il est possible de s'endormir bercé par de jolies mélodies et de se réveiller en mode pop rock pour un réveil qui booste la journée ! Cerise sur la partition, un tuner radio est disponible ! 10 stations AM et 30 stations FM peuvent être présélectionnées.

■ Les tablettes tactiles sous Android se multiplient même si leur disponibilité sur le marché est souvent aléatoire. **TazTag** a annoncé la sienne : TazPad. Elle utilise la technologie NFC. Le NFC est une technologie de communication en zone proche, sans fil et sans contact, pour échanger facilement des données. La tablette possède les critères suivants : écran 7 pouces couleur, capacitif multitouch, sortie HDMI full HD et se connecte en WiFi, Bluetooth, NFC et Zigbee, une caméra, GPS, et accéléromètre. Disponibilité annoncée : avril.

PC

ChipPC : un PC léger dans une prise de courant !

Spécialiste dans la sécurité, ChipPC propose des PC ultra légers d'un nouveau genre. Le plus spectaculaire est le JackPC qui a la forme d'une prise de courant murale ! Il embarque un système de type Linux ou Windows fonctionnant avec un processeur RISC avec peu de mémoire. Par contre, la connectique est très complète : 4 ports USB, DVI,

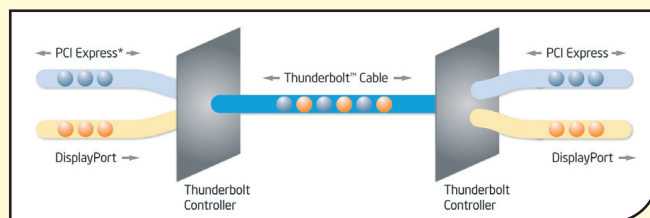
audio, lecteur vidéo. Cet ultra PC est idéal dans une configuration de réseau local, pour faire du poste virtualisé. La prise peut embarquer des applications sécurisées et identifiées par le système. Une belle technologie ! site : chippc.com



Connectique

Thunderbolt : un nouveau tuyau sur les PC et périphériques

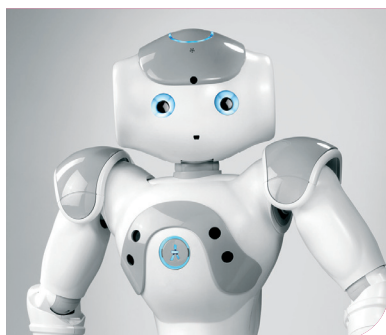
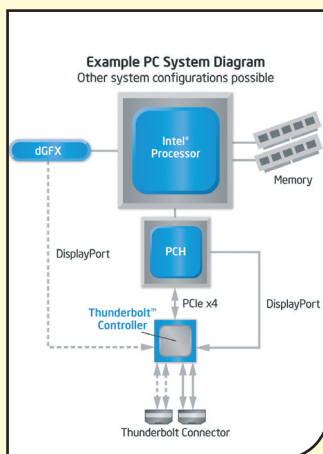
Dans le monde de la connectique, les choses bougent régulièrement. Alors que USB 3.0 a vocation à remplacer la v2, voire peut être le Firewire, Intel a dévoilé (enfin !) la première implémentation de sa nouvelle interface de connexion : Thunderbolt. Il s'agit de pouvoir combiner dans un seul câble (et donc un seul connecteur) différents



flux d'informations pour les Entrées / Sorties (vidéo, son, PCI Express), avec une approche bidirectionnelle (= les données pouvant aller dans les deux sens, simultanément), avec une bande passante théorique de 10 Gb par seconde. Et chaque port possède deux canaux Thunderbolt. D'autre part, il est possible de chaîner plusieurs périphériques ensemble comme en USB, Firewire, avec une latence la plus réduite possible.

Cette technologie semble très prometteuse et le premier constructeur à avoir annoncé son utilisation est Apple depuis

quelques semaines. Et les constructeurs de matériels commencent à dévoiler les produits Thunderbolt. Par exemple, LaCie a annoncé le Little Big Disk, un système Raid 0 de 500 Go SSD ou 1 To en disque normal. Il sera alors facile de créer une ferme de stockage Thunderbolt avec d'autres boîtiers. A terme, Thunderbolt va pouvoir remplacer plusieurs connectiques : port DVI, firewire, pourquoi pas USB même si cela va obliger à racheter le matériel. Par contre, aujourd'hui, Intel fournit le contrôleur nécessaire et aucune licence comme pour l'USB n'est disponible.



Robot

Aldebaran : un programme développeur pour NAO

Le surdoué de la robotique française, Aldebaran, propose depuis quelques semaines un programme développeur pour son robot NAO. Le but est de créer, tester des usages, des fonctions, des applications à NAO. Et les challenges sont nombreux : intelligence artificielle, fonctions cognitives, R&D, interaction

homme - robot. Ce programme développeur comprend la toute dernière version du robot, la suite complète de développement avec les API, IDE, SDK, l'accès à la communauté en ligne pour permettre de contacter les meilleurs experts NAO et surtout un accès au code source du robot. Prix du programme : 3 600 €.

Réplication de fusion avec SQL Server Compact

Cet article explique l'implémentation d'un système de synchronisation de données entre des terminaux mobiles et un serveur, en utilisant la réplication de fusion SQL (merge replication) entre une base centrale SQL Server et des bases mobiles SQL Server Compact.

La réplication SQL est un produit fiable, intéressant, mais qui peut parfois se révéler complexe en termes de configuration et de maintenance, surtout dans le cas de grandes volumétries de données. C'est pourquoi cet article aborde cette problématique et propose de nombreuses préconisations afin d'optimiser au mieux les performances du système.

Cet article est non seulement destiné aux personnes qui ne connaissent pas la réplication de fusion, mais également aux informaticiens plus expérimentés.

Pré-requis nécessaires :

- Notions en développement d'applications .NET avec Visual Studio.
- Connaissances minimales de SQL Server (2005 ou 2008 conseillé).
- Maîtrise de l'interface « Microsoft SQL Server Management Studio » conseillée.

PRINCIPES DE LA RÉPLICATION DE FUSION

Généralités

La réplication de fusion (merge replication) est un système puissant et riche en fonctionnalités qui permet la synchronisation de données entre une base centrale et plusieurs bases répliquées. Lors du processus

de réplication, seules les données modifiées (par des ordres UPDATE, INSERT, ou DELETE) sont prises en compte. Le système peut déterminer de manière intelligente quelles sont les données modifiées et quelles sont les données qui doivent être envoyées sur les bases répliquées. La réplication SQL offre la possibilité importante d'implémenter un système de filtres complet afin de distribuer au mieux les données sur les différentes bases. Ce qui fait également la force de la réplication, c'est qu'elle offre de nombreux outils d'aide à la résolution de conflits. Il sera d'ailleurs possible d'éviter toute sorte de conflits en paramétrant convenablement le système de synchronisation, et plus particulièrement les différentes publications.

Vocabulaire

La réplication de fusion est une fonctionnalité de SQL Server à part entière. Elle dispose de son propre vocabulaire. Il est important de connaître ce vocabulaire pour la compréhension de l'article.

Publisher	Il s'agit de la base publiée, c'est-à-dire celle qui va être répliquée.
Distributor	Il s'agit de la base de distribution, celle qui va être chargée de gérer la distribution des données vers les subscribers.

Subscribers (ou Abonnés) Il s'agit des bases qui vont recevoir les données répliquées. Dans notre article, les abonnés sont les terminaux mobiles.

Subscription (ou Abonnement) Demande de remise d'une copie de la publication à l'abonné.

Snapshot Un snapshot est un ensemble de métadonnées qui vont être utilisées lors des synchronisations.

Publication Une publication définit une liste d'éléments qui doivent être répliqués vers les abonnés. Elle définit aussi un ensemble d'options importantes (filtres, partitions, flux, etc....)

Article Un article est une table de la base publisher qui doit être répliquée. La publication est un ensemble d'articles.

Partition Il s'agit d'un ensemble de lignes ou de colonnes pour un abonné spécifique.

Publisher, Distributor et Subscriber

La notion de réplication SQL passe obligatoirement par les notions de « Publisher », « Distributor » et « Subscriber » [Fig.1].

Il s'agit en fait de 3 parties distinctes ayant chacune un rôle précis dans les processus de synchronisations. Ces 3 parties correspondent d'ailleurs à 3 bases de données différentes. Pour rappel, le publisher correspond à la base publiée, c'est-à-dire celle qui contient les données à répliquer. Le distributor correspond à la base qui distribue les données (base système), tandis que le



Je suis bilingue en .NET

subscriber (ou abonné) correspond à la base répliquée, c'est-à-dire celle qui reçoit les données répliquées. A noter que le publisher et le distributor peuvent être configurés sur le même serveur.

C'est toujours aux subscribers que revient la tâche d'initialiser une synchronisation avec le publisher.

Notion de snapshots

Les snapshots sont des métadonnées utilisées lors des synchronisations et sont exposées sur le serveur SQL. Les snapshots permettent de définir le schéma de la base à un instant t, ainsi que l'ensemble des contraintes liées.

Il existe 2 types de snapshots :

- Le snapshot principal qui doit être généré obligatoirement suite à la mise en place d'une publication.
- Un snapshot spécifique par abonné, lié à une partition (système de partitions).

Les synchronisations de données entre le PDA et le serveur SQL se basent sur ces snapshots.

Déroulement des synchronisations

Synchronisation initiale : [Fig.2]

Voici comment se déroule la première synchronisation :

- Le distributor utilise le répertoire de snap-

shots pour préparer le schéma de la base répliquée.

- Le schéma de la base mobile est créé.
- Les contraintes sont créées sur la base répliquée.
- Les données sont téléchargées.

Synchronisations suivantes : [Fig.3]

Lors des synchronisations suivantes, le distributor n'utilise plus les snapshots. Seul le différentiel des données est téléchargé.

SPÉCIFICITÉ DE LA RÉPLICATION AVEC SQL SERVER COMPACT

Réplication entre un serveur et des terminaux mobiles

Dans le cadre de projets mobiles, la réplication de fusion s'établit entre un serveur SQL Server et plusieurs bases mobiles de type SQL Server Compact. Chaque terminal possède une base de données SQL Server Compact, répliquée à partir d'une base de données SQL Server. Les bases SQL Server Compact représentent donc les abonnés aux publications serveur. Ces bases SQL Server Compact peuvent être :

- soit générées directement sur les terminaux par réplication SQL lors de la première utilisation d'une application mobile par exemple,
- soit déployées directement avec une application mobile

Pour les projets mobiles, la notion de filtres de réplication va être encore plus importante puisqu'elle va permettre de filtrer au mieux les données à distribuer sur l'ensemble du parc de terminaux mobiles. De ce fait, chaque terminal possèdera uniquement ses propres données. Nous verrons d'ailleurs plus loin qu'il pourra s'avérer nécessaire de limiter la taille des bases mobiles, et donc de filtrer les données, en raison de contraintes de volumétrie.

Il existe plusieurs types de réplication dans SQL Server, cependant seule la réplication de fusion est prise en charge et permet de synchroniser des données entre un serveur et des terminaux portables. Pour la mobilité, nous parlons de « web synchro » dans le sens où les PDA doivent s'appuyer sur une DLL exposée sur un serveur web IIS pour échanger leurs données (= Agent de réplication). L'URL de cette DLL sera donc précisée dans le code de l'application mobile. Les synchronisations de données avec SQL Server Compact sont soumises à des règles importantes :

L'utilisation d'un serveur IIS est indispensable pour implémenter un système de réplication entre un serveur et des terminaux mobiles (= web synchro).

Les terminaux mobiles peuvent uniquement être des subscribers (jamais des publishers, ni des distributors).

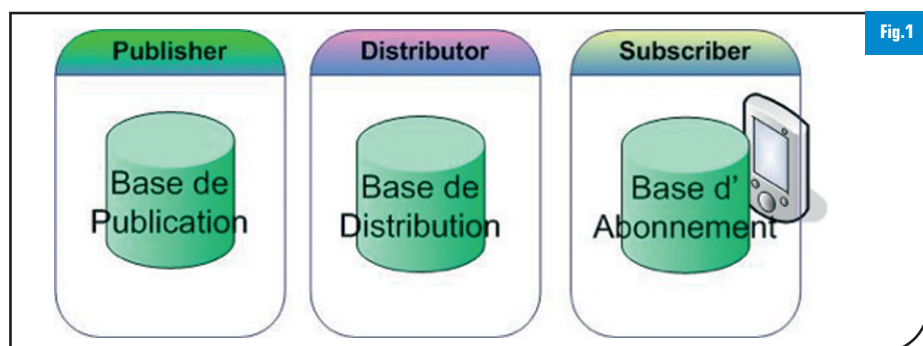


Fig.1

Restrictions et limitations

SQL Server Compact peut être considéré comme un système de gestion de bases de données allégé et conçu pour les terminaux portables. Aussi, il comporte moins de fonctionnalités et présente plusieurs limitations. Par exemple, certaines modifications de schéma sur l'abonné sont interdites et peuvent générer des échecs de synchronisation. Voici la liste complète des actions à ne

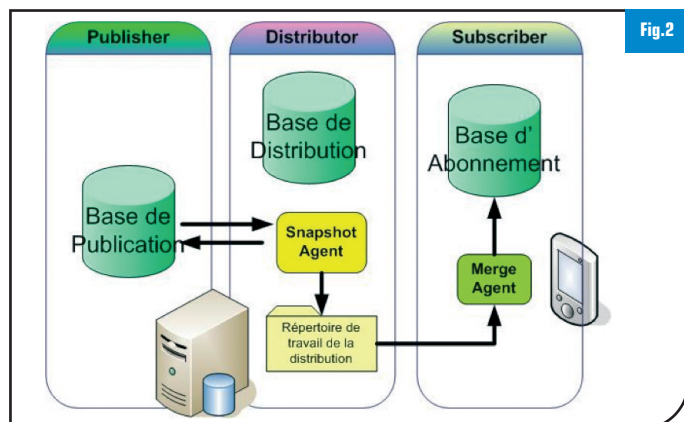


Fig.2

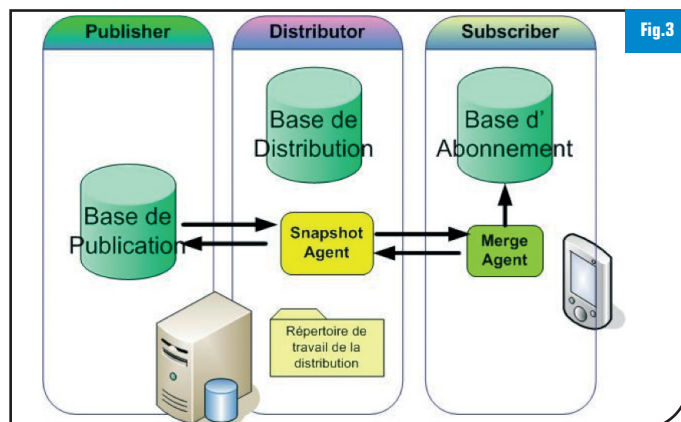


Fig.3

pas entreprendre sur la base de données abonnée :

- Supprimer ou renommer une table
- Ajouter ou supprimer une colonne dans une table
- Ajouter ou supprimer une clef primaire ou une clef étrangère

De la même façon, certaines fonctionnalités ne sont pas prises en charge par SQL Server Compact :

- Procédures stockées
- Propriétés étendues
- Vues
- Contraintes CHECK
- Fonctions utilisateurs
- Triggers

Enfin, Il est à noter que les bases SQL Compact ne peuvent pas excéder la taille de 4Go (en version 3.5).

Pré-requis

Voici la liste de tous les éléments nécessaires pour implémenter convenablement la réplication SQL dans le cadre de projets mobiles :

Environnement de développement	<ul style="list-style-type: none"> - .NET Framework 2.0 (ou versions ultérieures) - Visual Studio 2005 (ou versions ultérieures) - ActiveSync 4.x ou Windows Mobile Device Center (pour Vista et 7)
---------------------------------------	--

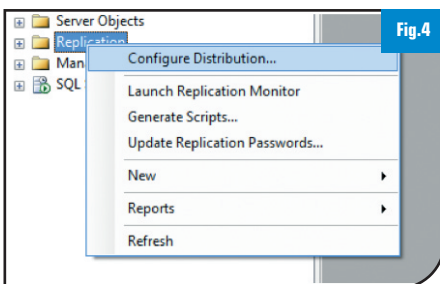


Fig.4

Serveur SQL

- SQL Server 2005 ou 2008
- Composants de réplication SQL Server 2005 ou 2008
- Outils de réplication SQL Server Compact

Serveur IIS

- IIS 5.x (ou versions ultérieures) installé sur le même serveur ou sur un serveur différent
- Pour IIS 7.0, il est nécessaire d'installer les composants de compatibilité IIS 6.0

Pour pouvoir construire un système de réplication entre des terminaux mobiles (SQL Server Compact) et une base centrale (SQL Server), il est nécessaire d'installer les outils de réplication SQL Server Compact.

Téléchargement des « Microsoft SQL Server Compact 3.5 Server Tools » :

<http://www.microsoft.com/downloads/details.aspx?FamilyID=b18327f3-96e1-415d-b037-9e0c46d49956&DisplayLang=en>

Installation et déploiement sur un terminal mobile

Pour que la réplication SQL soit prise en charge par un terminal, il est question d'installer certains composants sur le PDA. Les fichiers d'installation CAB se situent par

défaut dans le répertoire "%Program-Files%\Microsoft SQL Server Compact Edition\3.5\Devices\plateforme\processeur\" :

- sqlce.type_appareil.plateforme.processeur.CAB
- sqlce.dev.langue.type_appareil.plateforme.processeur.CAB
- sqlce.repl.type_appareil.plateforme.processeur.CAB

Ces composants dépendent du type d'appareil, de plateforme et également du processeur de l'appareil.

A noter que les bons fichiers CAB sont poussés et installés automatiquement par Visual Studio lors du lancement d'une phase de déploiement.

CONFIGURATION DU DISTRIBUTOR

Pour implémenter un système de réplication, il est important dans un premier temps de configurer la base de distribution. Cette configuration doit normalement s'effectuer une seule fois.

Procédure :

Il faut dans un premier temps cliquer-droit

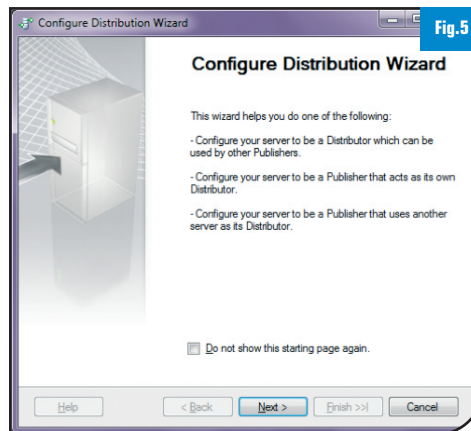


Fig.5

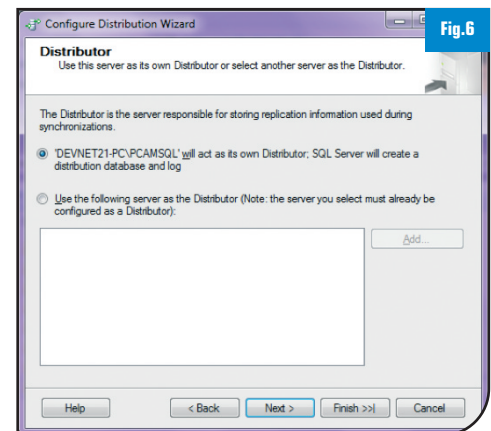


Fig.6

Je parle couramment
le PHP

sur le nœud « Replication » et choisir l'option « Configure Distribution... » : [Fig.4]

Cliquez sur « Next » sur le premier écran du wizard : [Fig.5]

Sur l'écran suivant, vous allez préciser que le distributor utilisera la même instance SQL Server que la base publiée. Cochez la première option et cliquez sur « Next » : [Fig.6]

L'écran suivant vous demande de saisir le chemin vers le répertoire des snapshots : [Fig.7 et 8]

Comme la réplication est de type « web synchro » et utilise un serveur IIS pour assurer l'échange des données, il est nécessaire de stocker les snapshots dans un répertoire accessible à la fois par le serveur IIS et par l'agent de snapshots situé sur le serveur. Il est donc important de créer au préalable un répertoire partagé et de spécifier son chemin réseau (ex : \\NOM_MACHINE\rep).

Créez donc un répertoire nommé « repldata » à la racine de C. Partagez ce répertoire en y attribuant les droits nécessaires pour autoriser l'accès à SQL Server mais également au serveur IIS :

Une fois le répertoire créé et paramétré, notez son chemin d'accès réseau. Inscrivez ensuite ce path dans la nouvelle fenêtre du wizard SQL Server, puis cliquez sur « Next ». L'écran suivant permet de préciser le nom et le path de la base de distribution. Laissez les paramètres par défaut et cliquez sur le bouton « Next ». Cet écran permet d'activer la base de distribution. Ne modifiez rien et cliquez sur le bouton « Next ». Sur l'écran suivant, vous pouvez sélectionner la deuxième option si vous souhaitez générer un script de configuration du Distributor. Cliquez sur « Next ». Enfin, ce dernier écran permet de récapituler les différentes actions qui vont être entreprises par SQL

Server. Pour lancer réellement la configuration du distributor, cliquez sur le bouton « Finish » : [Fig.9] Une fois le processus achevé, vous pouvez remarquer la présence d'une base nommée « distribution » dans le nœud « System Databases » : [Fig.10]

Si vous souhaitez maintenant modifier les propriétés de la distribution, vous pouvez cliquer droit sur le nœud « Replication » et sélectionner « Distributor Properties... ».

CONFIGURATION DES PUBLICATIONS

Notion de publication

Une publication est rattachée à une et une seule base publisher.

La publication permet de définir l'ensemble des éléments suivants :

- Articles : Ce sont les tables et les colonnes répliquées
- Filtres : La publication peut se baser sur un système de filtres pour restreindre le nombre de données sur chacun des abonnés
- Options : De nombreuses options peuvent être liées à une publication particulière et permettent de définir le comportement des synchronisations.

Plusieurs publications peuvent être rattachées à une même base. Si le modèle de données le permet, nous pouvons par exemple distinguer les tables du référentiel de travail de celles qui correspondent aux tables destinées à stocker les données collectées sur le terrain par les différents terminaux. Dans notre exemple, il est astucieux d'implémenter 2 publications différentes. La première permettra de répliquer les tables du référentiel de travail (tables dont les données doivent uniquement descendre sur les PDA), tandis que l'autre publication va être utilisée pour la

remontée des données terrain. Ces 2 publications implémentées ne posséderont pas les mêmes options, et seront configurées selon le sens des flux (descendant ou remontant).

Le fait de séparer les tables du référentiel de celles des données terrain en 2 publications différentes permet d'éviter les conflits.

Autre avantage : dans le code .NET de l'application mobile, il sera possible de lancer la synchro du référentiel ou celle des exports terrain indépendamment et à différents endroits de l'application (selon les besoins métiers).

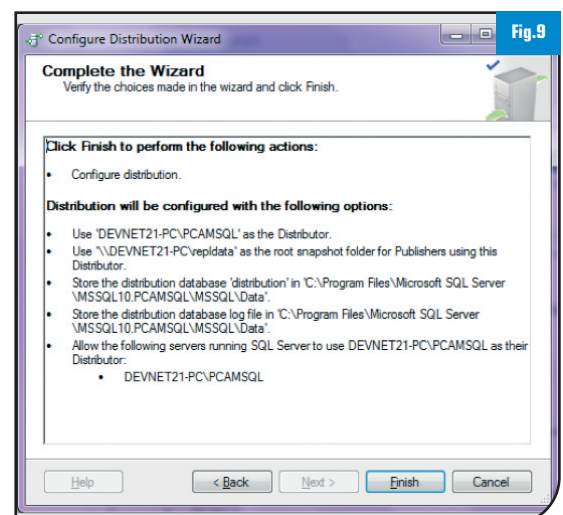
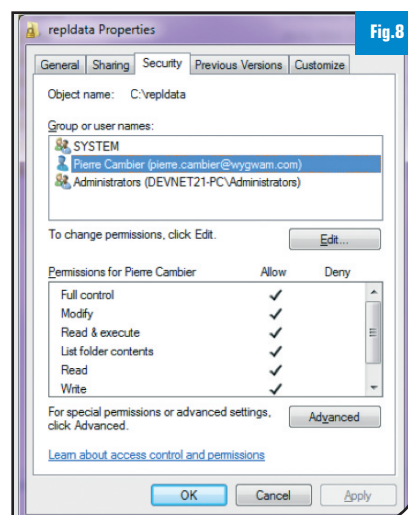
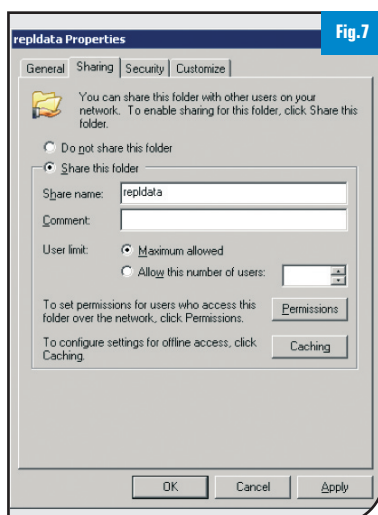
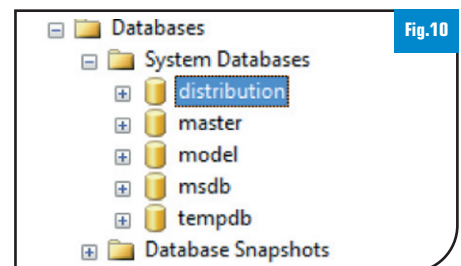
Création d'une nouvelle publication via le wizard

Une publication peut se créer via un wizard ou via un script. Nous utiliserons le wizard pour créer la publication. Pour créer une nouvelle publication, il est nécessaire de déployer le nœud « Replication », de cliquer droit sur « Local Publications » et de choisir l'option « New Publication... ». S'ensuivront alors une série d'écrans permettant de sélectionner les articles, de définir les filtres et autres options de la réplication.

Procédure :

Sur le premier écran du wizard, cliquez directement sur « Next ».

Sur le second écran, vous devez sélectionner la base à répliquer (publisher). Sélectionnez donc « TECHNOMADE » et cliquez



ensuite sur « Next ». Ensuite, sélectionnez « merge replication » et cliquez sur « Next ». Pour rappel, la « merge replication » est le seul type de réplication pris en charge par SQL Server Compact. Sur l'écran suivant, décochez « SQL Server 2008 » et cochez l'option « SQL Server 2005 Mobile, SQL Server Compact 3.1 and higher ». En validant ce dernier écran en cliquant sur « Next », vous arriverez sur l'écran du choix des articles.

Définition des articles

Les articles ne sont autres que l'ensemble des tables et colonnes qui vont être répliquées sous l'utilisation d'une publication. L'écran « Articles » représente la liste des articles définis pour la publication. L'écran permet d'ajouter ou de supprimer facilement des tables à répliquer en les cochant ou décochant dans la liste. Chacune des tables est représentée sous la forme d'un nœud, qui une fois déployé, permet également de cocher ou de décocher les colonnes répliquées. Pour créer la publication liée au référentiel, nous allons donc sélectionner l'ensemble des tables excepté les tables d'export : [Fig.11].

A noter que l'option « Show only checked articles in the list » permet de ne visualiser que les articles sélectionnés dans la liste.

Des propriétés supplémentaires peuvent également être définies sur chacun des articles. Le bouton « Article Properties » permet d'afficher une fenêtre permettant de définir les options des articles. Ces propriétés peuvent être saisies pour des articles particuliers, ou pour l'ensemble des articles sélectionnés (option « Set properties for all

table articles »). Le premier groupe de paramètres permet de définir les objets copiés sur l'abonné par le processus de réplication (index, clefs, contraintes...). Le second groupe, quant à lui, est le plus important car il définit des propriétés concernant le comportement des tables répliquées. La propriété importante, à laquelle il est nécessaire de faire attention, est la « Synchronisation direction ».

En effet, cette option permet de préciser si les données d'une table seront répliquées d'une manière bidirectionnelle ou uniquement dans le sens « Serveur vers Abonné ». Pour la mise en place de notre publication, nous allons donc modifier l'option « Synchronisation direction » pour choisir « Download only to Subscriber, prohibit Subscriber changes » : [Fig.12].

En validant l'option « Download-only » en cliquant sur le bouton « OK », les petites icônes qui symbolisent les articles sélectionnés dans la liste seront grisées. De la même façon, l'option « Highlighted table is download only » sera cochée et l'intitulé « For this table, only changes made at the Publisher or a server subscription are replicated » sera visible.

Nous venons ainsi de définir l'ensemble des articles liés à la publication du référentiel de travail. Pour la seconde publication, c'est-à-dire celle qui permettra de remonter les informations saisies sur le terrain, il sera nécessaire de cocher les tables RAPPORT_INTERVENTION et RAPPORT_INTERVENTION_FOURNITURE et de ne pas modifier les propriétés afin de les laisser en mode « Bidirectionnel ». Le fait de déclarer ces 2 articles en « Download only » rendrait bien évidemment la remontée des saisies terrain

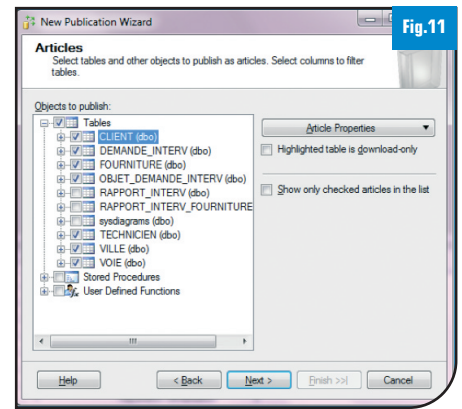


Fig.11

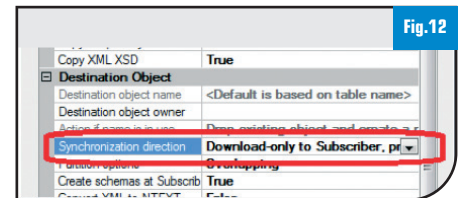


Fig.12

impossible. La sélection et la définition des articles prennent fin en cliquant sur le bouton « Next ». L'écran suivant vous avertira que la réplication SQL va créer une colonne supplémentaire sur chacun des articles sélectionnés.

Cette colonne de type « Unique identifier » est essentielle au bon fonctionnement de la réplication, et permettra au moteur de réplication de tracer l'ensemble des modifications sur les tables à répliquer. Sans cette colonne, les synchronisations différentielles de données seraient donc impossibles. En cliquant sur le bouton « Next », vous passez directement à la définition des filtres...

■ Pierre Cambier

.NET & Windows Mobile Solution Developer
Microsoft Certified Technology Specialist
Wygam



Comment sécuriser son site web ?



Qui pourrait penser qu'en 2011, il faille encore rappeler les fondamentaux de la sécurité informatique aux développeurs et webmasters ? Malgré 20 ans, ou presque, de rappels, de mises en garde, de livres, de dossiers dans Programmez !, il faut croire que la sécurité, le code sécurisé font toujours partie des réflexes oubliés. Cependant, reconnaissons que la situation s'améliore. Même si les attaques basiques restent des classiques : injection de code, injection SQL, dépassement de capacité des champs, porte dérobée. Avec l'utilisation du cloud computing, des plaintes se sont élevées quant au manque de sécurité du cloud. Mais ceci est finalement un faux problème car la sécurité du

cloud dépend tout d'abord de la bonne utilisation des mécanismes de sécurité web, des sites web. Bref, du b.a.-ba !

Le développeur web, le webmaster dispose de tous les outils, de toute l'aide nécessaire pour blinder son code, sécuriser les accès, éviter les trous dans les bases de données, les formulaires, etc. Mais que les choses soient claires : c'est à VOUS de le faire. Personne ne le fera à votre place !

Oui, implémenter la sécurité est parfois long et pas très excitant. Oui cela coûte du temps et de l'argent (certains projets auront donc tendance à aller au plus simple). Non, la sécurité n'a rien de superflu.

Nous allons, dans ce dossier, revenir sur les fondamentaux à implémenter, à vérifier.

■ François Tonic



Faibles et offuscation

Les menaces courantes et les contre-mesures possibles

Catégories de menaces	Contre-mesures possibles
Vol d'identité (Spoofing)	Utilisation d'une authentification forte. Ne pas stocker les secrets (mot de passe...) « en clair ». Ne pas échanger de secrets « en clair ». Protéger les cookies d'authentification avec SSL.
Altération de données (Tampering)	Utiliser la signature et le hash de données. Utiliser des protocoles de communication résistants face à l'altération de données et permettant de contrôler l'intégrité des messages échangés.
Répudiation	Avoir une traçabilité forte. Utiliser des signatures numériques.
Divulgaration d'information (Information disclosure)	Utiliser un chiffrement fort. Utiliser des protocoles de communication permettant d'assurer la confidentialité des messages échangés. Ne pas stocker les secrets « en clair » (comme les mots de passe)
Déni de service (Denial of service)	Utiliser des mécanismes de limitations de bande passante.
Élévation de privilèges	Suivre le principe de n'accorder que les privilèges nécessaires pour effectuer l'action demandée.



La table ci-dessous propose un système de quantification permettant de calculer le risque face à une menace :

	Elevé (3)	Moyen (2)	Faible (1)
Damage potentiel	L'attaquant peut tromper la sécurité, obtenir tous les accès, être administrateur, télécharger du contenu.	Fuite de données sensibles.	Fuite de données banales.
Reproductibilité	L'attaque peut être reproduite autant de fois que voulu.	L'attaque peut être reproduite à certaines conditions.	L'attaque est difficile à reproduire.
Exploitableté	Un développeur novice peut exécuter l'attaque rapidement.	Un programmeur expérimenté pourrait exécuter l'attaque, puis répéter les étapes.	L'attaque ne peut être réalisée que par un développeur chevronné avec des connaissances pointues sur les couches basses.
Utilisateurs Affectés	Tous les utilisateurs.	Quelques utilisateurs.	Une très petite partie des utilisateurs.
Découverte	Des informations publiées expliquent l'attaque.	La vulnérabilité n'est connue que d'un groupe restreint d'utilisateur.	Le problème est obscur et complexe et certainement difficile à découvrir.

Si nous appliquons cette mesure de risque à deux menaces très connues, nous obtenons les quantifications suivantes :

Menace	D	R	E	A	D	Total	Niveau
L'attaquant obtient des informations d'authentification via un outil de supervision du réseau	3	3	2	2	2	12	Elevé
Injection de commandes SQL	3	3	3	3	2	14	Elevé

Offuscation

L'offuscation est l'art de brouiller les pistes ou encore obscurcir le code. Cette méthode ralentit considérablement la rétro-conception en fonction des algorithmes utilisés. Il existe environ une quarantaine d'algorithmes d'offuscation. Un algorithme d'offuscation possède en général 3 mesures permettant de connaître son efficacité : sa puissance, sa robustesse et son coût.

La puissance d'un offuscateur est la mesure selon laquelle le code est difficile à lire pour un humain. Par exemple, un des algorithmes d'offuscation est basé sur la destruction systématique de tout commentaire dans le code qui pourrait aider à comprendre son fonctionnement. Un autre algorithme d'offuscation consiste à modifier tous les noms des variables du code. Le programme fonctionne de la même manière mais est plus difficile à lire pour l'humain.

La robustesse d'un algorithme d'offuscation est la difficulté qu'aurait un dé-offuscateur (un programme) à régénérer une structure de code lisible. L'un des mécanismes les plus efficaces est celui de l'interprétation, mais c'est aussi l'un des plus coûteux : cela consiste à modifier le code MSIL pour le rendre interprétable par un runtime faisant partie de l'offuscateur.

Le coût représente par exemple la perte de performances à l'exécution induite par l'utilisation d'un algorithme d'offuscation. En général, plus l'algorithme est robuste, plus il est coûteux : il est nécessaire de bien connaître les effets des algorithmes d'offuscation sur l'exécution et sur la protection du code. Sans cette connaissance, le développeur pourrait se méprendre sur la qualité de la protection qu'il donne à son code.



■ Frédéric Queudret

MVP Client Application & CTO de la société Mpoware, accélérateur d'innovation. Société française d'édition de logiciels et de prestations de services sur les technologies .NET.

<http://www.mpoware.com/>

JOURNÉES FRANCOPHONES DE LA SÉCURITÉ - 10 MAI 2011 - Espace Saint-Martin
Des conférences sur la sécurité des SI exclusivement en français.

Contact Marc Brami - 01 40 92 05 55 - marc.jacob@globalsecuritamag.fr - www.gsdays.fr



Optez pour un guichet unique :
5 datacentres,
17 filiales dans le Monde,
25 *Points Of Peering* en Europe,
350 personnes à votre service,
800 Gbps de capacité Internet,
90 000 serveurs.

10 sms offerts*

Code promo : **prosms1011**

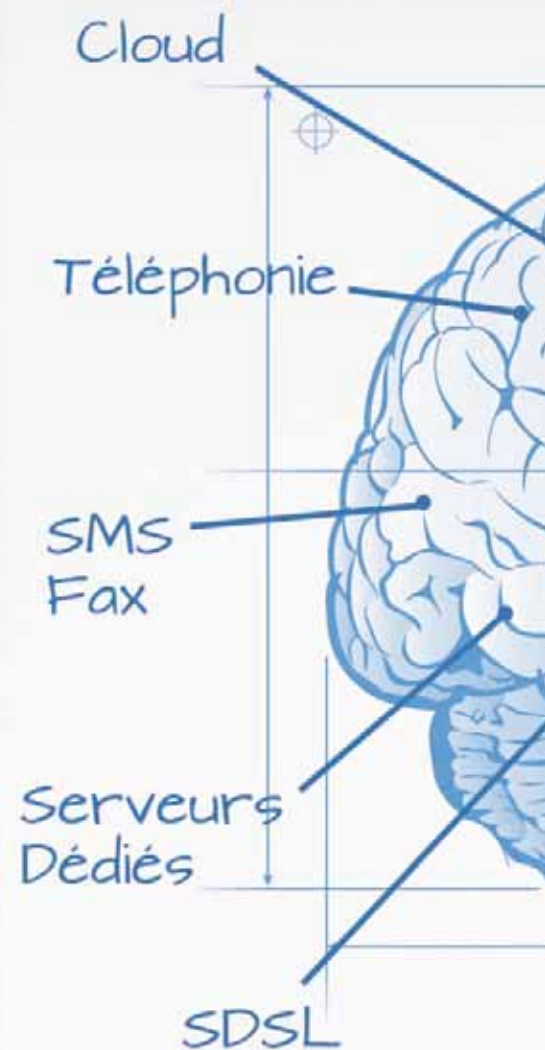
1. Connectez-vous sur www.ovh.com,
2. Commandez le produit de votre choix,
3. Entrez votre code promotionnel.

*Offre non cumulable, valable pour toute nouvelle commande sur le site ovh.com avant le 31/04/2011 minuit. Offre soumise à conditions plus d'information sur ovh.com.

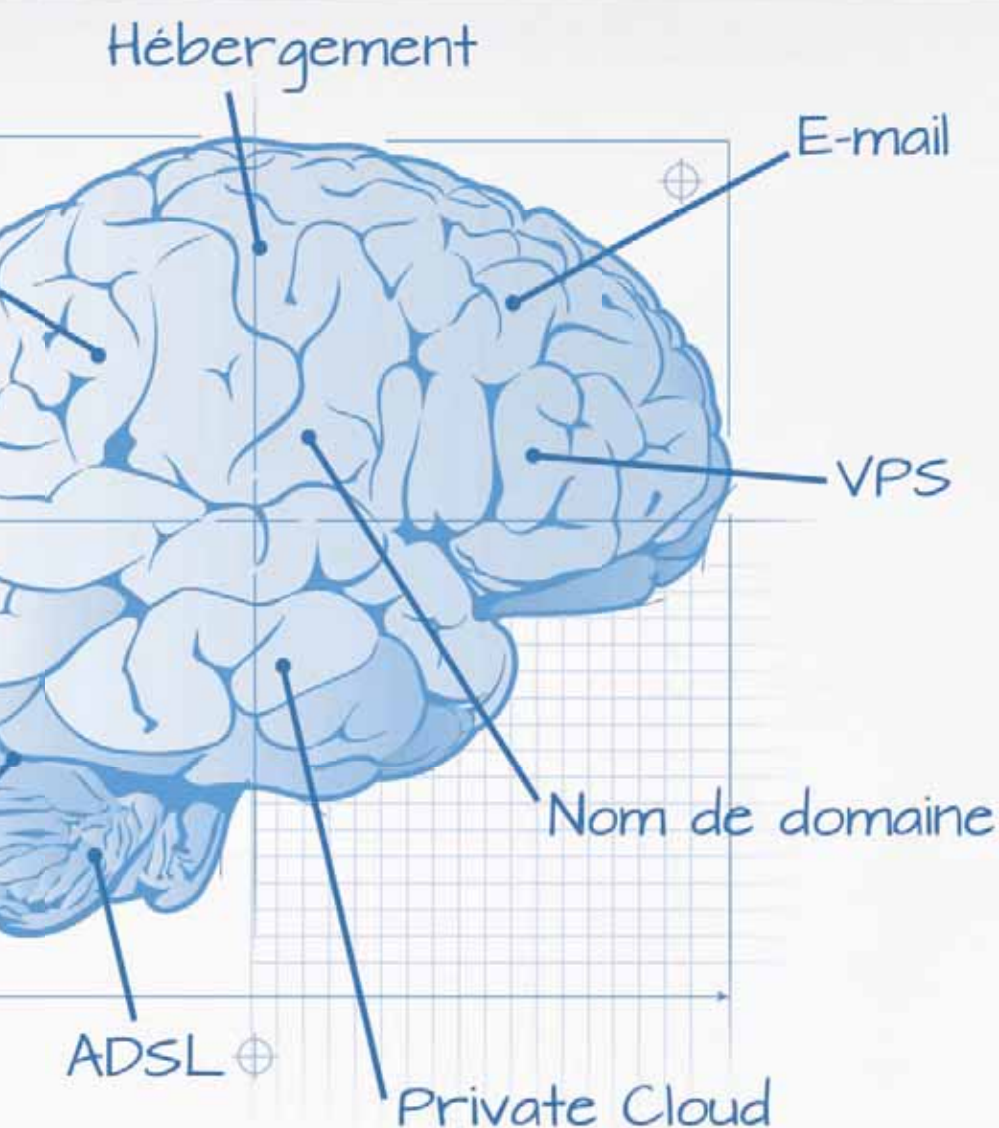


Plus d'informations sur : www.ovh.com

Support commercial : **08 203 203 63** (du lundi au vendredi de 8h00 à 20h00)



Ovh centralisateur d'idées!



Solutions de téléphonie et d'hébergement Internet

Serveurs dédiés | VPS | Cloud | Housing | Téléphonie | SMS | Fax





La sécurité par les bonnes pratiques

Les bonnes pratiques sont toujours appréciées des développeurs, mais souvent, rapidement oubliées. Cette carence peut avoir d'importantes conséquences. C'est pourquoi, quel que soit le langage, la technologie, il est important, voire crucial, de revenir aux fondamentaux ! Cet article se veut une piqûre de rappel...

L'internet est considéré comme une jungle depuis ses débuts, mais le phénomène s'est accentué avec l'élargissement de la bande passante (56ko, ADSL, Fibre...). Cependant, le contenu disponible est toujours aussi abondant et explose littéralement. Qu'ils soient d'information ou de commerce en ligne, ces sites sont très convoités et les risques d'attaques sont nombreux et omniprésents.

Soyons clair, parler de risque zéro sera utopique. Le risque zéro n'existe pas et n'existera jamais, surtout en matière de sécurité informatique. En effet un site web non sécurisé peut se voir très rapidement "imposer" du contenu non souhaité. Le résultat est impitoyable car instantané et les effets peuvent être pire comme la perte de lecteurs et donc de fréquentation, une chute des ventes, etc. Toutefois, il existe des méthodes, des tactiques pour bien mener son projet web et limiter les risques d'attaque, d'insécurité. Des pratiques, simples et efficaces, sont disponibles !

RAPPEL DES RISQUES

Avant d'entrer dans le vif du sujet, vous devez placer la sécurité en tête de liste de vos priorités. Il ne faut jamais la négliger. C'est aussi vital que de respecter les spécifications W3C ou surveiller la qualité de son référencement. La sécurité web s'apparente à une stratégie claire et définie : surface d'attaque, risques potentiels, contre-mesures à appliquer, mesures préventives. Les risques potentiels, les failles sont nombreux et se cachent partout. A vous de créer des checklists où chaque point devra être vérifié, validé. Souvent, le premier but des attaques est de modifier l'apparence visuelle d'un site : contenus parasités, faux contenus, modification de l'interface, fausse redirection, etc. Les techniques ne manquent pas. Un des classiques est l'usurpation de nom de domaines. Le risque

suivant, mais au moins aussi important que le précédent concerne le vol de données. Cette source d'informations provient souvent de la base de données et par conséquent vous perdez complètement la totalité de la confidentialité de votre site. Ce vol de données peut être réutilisé pour du Spam ou par le biais de relais pour effectuer des opérations malhonnêtes, ce qui est plus grave. La sécurité des données et leur confidentialité est primordiale !

LES BONNES PRATIQUES

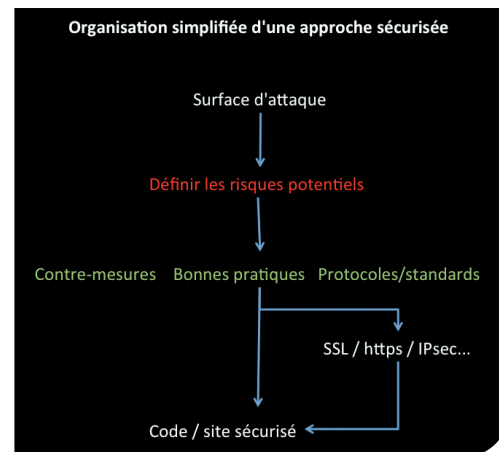
Les bonnes pratiques sont des principes de base que vous devez connaître et maîtriser. Elles sont connues, largement documentées, mais ne sont pas toujours appliquées par défaut par les outils. C'est VOUS le développeur qui devez les mettre en place. Petit rappel.

Utilisation de gestionnaire de formulaire

Le principe d'utilisation de gestionnaire de formulaire doit permettre de générer un écran en semi-automatique. Ce principe d'utilisation vous permet de réaliser une page de saisie commune, sous une même structure, un seul fichier contient donc l'ensemble des fonctionnalités pour générer un formulaire. Lors de la mise en place de celui-ci, les données entrantes passeront par le même endroit, limitant ainsi les risques. Ainsi, ces données entrantes doivent obligatoirement passer à travers un même point de contrôle avant de continuer le processus classique du traitement des données.

Front-end

Le « front end » est la partie visible d'un site web. Il montre à l'internaute un contenu, une interface que vous aurez décidé d'afficher, c'est-à-dire ce qu'il peut voir. Dans un premier temps, il va accéder au contenu,



pour ensuite se diriger dans les différentes pages et formulaires que vous lui aurez proposés. L'ensemble de cette navigation peut laisser apparaître des risques car il s'agit d'un des premiers points d'attaques d'une personne malhonnête.

Les attaques, les plus courantes sont :

- Les attaques XXS
- Les Injections SQL

Bien sûr, si vous souhaitez effectuer des tests pour chaque page et formulaire, il faut prévoir un peu de temps car ils sont souvent effectués manuellement. Cependant, vous pouvez maintenant vous appuyer sur des outils automatiques comme en propose le navigateur Mozilla Firefox :

- XXS Me (<https://addons.mozilla.org/en-US/firefox/addon/xss-me/>)
- SQL inject Me (<https://addons.mozilla.org/en-US/firefox/addon/sql-inject-me/>)

Ces 2 addons proposent un résultat poussé pour vous aider dans l'identification des problèmes. Bien entendu, ce ne sont pas eux qui vont vous corriger les différents risques de sécurité mais vous aurez déjà une idée plus précise des risques que vous pouvez rencontrer.

Utiliser un framework

L'utilisation d'un framework peut être une solution alternative au niveau des risques de sécurité. De nombreux développeurs vont préférer s'en servir car il répond en partie aux failles les plus répandues et souvent c'est un produit maintenu par une société et/ou une communauté. Pour rappel, un framework est un outil terminé et fiable car il contient un noyau et celui-ci se trouve déjà sécurisé. Son utilisation vous soulagera des tests à réaliser manuellement qui ont été

décrits précédemment. Bien sûr, le choix d'un framework est toujours difficile, mais il est préférable d'en choisir un de connu avec beaucoup d'utilisateurs car vous êtes certain si un nouveau risque apparaît, qu'il sera identifié et corrigé plus rapidement.

Classe de base de données

Lors d'un développement, il existe d'autres méthodes pour réduire les failles de sécurité comme l'utilisation d'une classe de gestionnaire de base de données. Celle-ci va vous permettre de regrouper un certain nombre de fonctions pour communiquer entre un formulaire et votre base de données.

Cette classe vous sera utile car quel que soit le nombre de pages dans votre site web, vous passerez automatiquement par celle-ci. Ainsi, vous avez un seul point à maintenir au lieu de l'effectuer dans toutes les parties de votre projet. Cette méthode peut aussi être utilisée pour nettoyer automatiquement les données entrantes avant de communiquer avec votre base de données. Elle vous sera utile comme une classe de préparation.

Tests unitaires

Les tests unitaires sont aussi un bon moyen de vérifier la vulnérabilité de votre site web, en particulier liée aux accès et fonctionnalités restreintes. Il s'agit d'un outil dont le but principal est de vous soulager dans la réalisation de certaines tâches répétitives en automatique, ce qui vous permet de gagner du temps dans cette série de tests.

Tester votre code

Un autre critère souvent oublié consiste à effectuer des tests en continu car il n'est pas rare de voir un développeur réaliser complètement une page web avant de passer à l'étape des tests et bien sûr se rendre compte des problèmes un peu tard !

Cette méthode n'est pas recommandée car souvent le résultat obtenu peut créer des soucis et par conséquent nécessiter de revoir une partie du développement (même pour les développeurs confirmés). Bien sûr, les tests ne sont pas les seuls à poser problème, d'autres bugs courants peuvent apparaître aussi bien au niveau de la présentation, de la saisie ou encore d'une page.

Méthode de programmation

Quelle que soit la méthode que vous allez utiliser, il est primordial de garder une cohérence dans votre code. Cette bonne pratique permet de vous y retrouver plus facilement, même quelques jours, semaines ou mois après. Cette méthode de travail vous sera utile lors du passage des tests de vulnérabilités. Bien entendu, si vous ne respectez pas cette méthode, vous allez être confronté à des risques plus importants au niveau de la sécurité et vous mettrez plus longtemps à effectuer les corrections pour que votre page fonctionne.

CONCLUSION

Ces bonnes pratiques montrent que quel que soit le langage utilisé, des règles existent pour limiter les risques de sécurité. Cependant le nombre d'attaques et la technologie avancent vite, c'est pourquoi il est important de faire prendre conscience aux développeurs des risques auxquels ils peuvent être confrontés

■ Christophe Villeneuve

Consultant pour Alter Way solutions, auteur du livre « PHP & MySQL-MySQLi-PDO, Construisez votre application », aux Éditions ENI. Rédacteur pour nexen.net, Trésorier AFUP et membre du LeMug.fr, PHPTV, PHPteam...

Passionné(e) par le développement et les nouvelles technologies Microsoft ?

Connaissez-vous un des sujets XML, .NET Framework, C#, Web Services, WCF, WF, SQL Server, BizTalk Server, SharePoint, SOA, ASP.NET ?

Spécialisez-vous chez **Codit**, leader en intégration, et devenez **expert** en Microsoft BizTalk, AppFabric et Azure !

En savoir plus? Veuillez visiter la page «jobs» sur www.codit.eu et contactez-nous.

www.codit.eu
jobs@codit.eu

codit
integrating your business



Protéger son site web avec PHP

Un aspect à ne pas négliger concerne les attaques envers les formulaires car c'est un point d'entrée crucial pour accéder à une partie où vous ne possédez pas les droits d'accès... normalement.

Les attaques XSS

Le Cross Site Scripting ou XSS est une attaque qui s'utilise en l'absence de traitements, de contrôles et de validation de données. En effet, elle consiste à écrire un message ou du code sur une page d'un site web afin qu'il soit exécuté normalement par les visiteurs. Ce type d'attaque XSS se fait par le biais d'un formulaire ou la possibilité de laisser un commentaire dans un blog. Elle se présente par l'envoi d'informations dans des balises Javascript :

```
<?php
$title = "Un petit exemple";
$title .= "<script>alert ('Programmez 2011')</script>";
echo $title."<br>";

?>
```

Le résultat montre un message qui ne devrait pas apparaître : [Fig.1](#). Pour résoudre ce problème, vous pouvez utiliser la fonction `htmlentities()` ou `htmlspecialchars()` car les caractères '<' et '>' deviennent '<' et '>'. Par conséquent quand la page sera chargée par votre navigateur, vous verrez bien '<' et '>' mais pas interprété et donc non exécuté.

```
<?php
$title = "Un petit exemple";
$title .= "<script>alert ('Programmez 2011')</script>";
echo htmlentities($title)."<br>";

?>
```

Le résultat est le suivant : [Fig.2](#). L'exemple montre un principe simple d'attaque XSS, mais avec cette même technique, vous pouvez être redirigé vers un autre site sans vous en rendre compte.

Les injections SQL

L'injection SQL est un des principaux problèmes que vous pouvez rencontrer avec votre base de données. Pour rappel, une base de données possède les identifiants et surtout toutes sortes de rensei-

gnements importants. C'est une mine d'informations. Une injection consiste à ajouter du code SQL dans une requête existante. Cela est possible avec l'envoi de données provenant des variables GET ou POST utilisées dans les formulaires d'identification.

Votre requête peut se présenter sous cette forme :

```
<?php

$login = $_POST['login'];
$motpasse = $_POST['password'];

$sql="
SELECT id FROM user WHERE
login = $login
AND
motpasse = $motpasse
";

$resultat=mysql_query ($sql) or die (mysql_error);

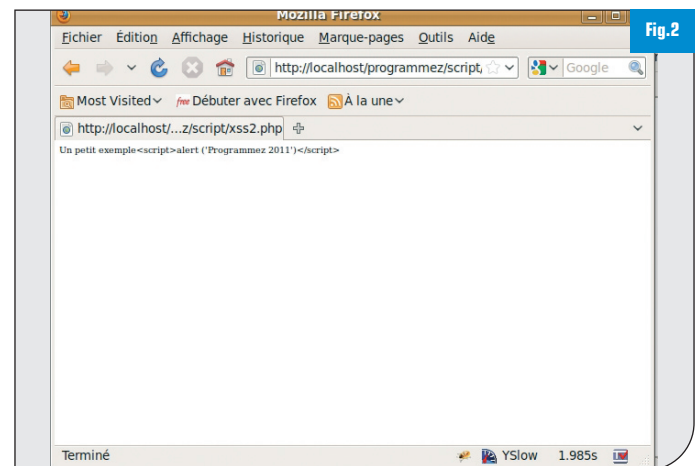
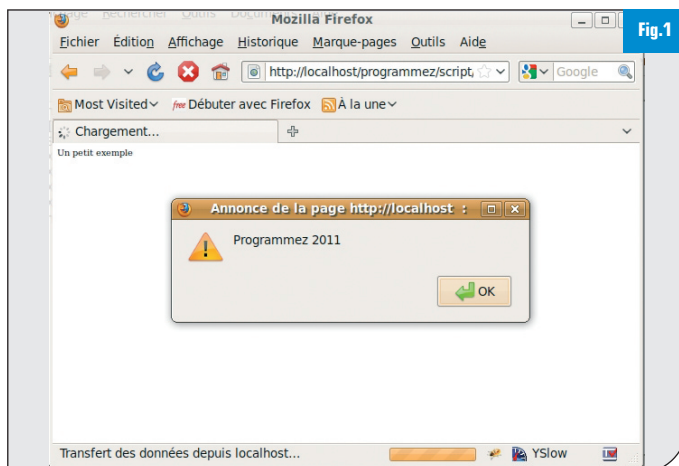
if ($resultat)
    echo "Ligne Trouvée";
else
    echo "Aucune données";

?>
```

Si aucune quote (apostrophe) n'existe dans votre requête SQL, vous risquez l'envoi de code non souhaité. Lors de l'envoi des données du formulaire, vous ne connaissez pas les identifiants, par conséquent vous pouvez envoyer à partir d'un formulaire de saisie des informations de ce type :

```
<?php
$login=" ' OR 1 ";
$motpasse=" ' OR 1 ";

?>
```



vous obtenez alors la requête suivante :

```
sql = "SELECT id FROM user WHERE login = '' OR 1 AND motpasse = '' OR 1 ";
```

Et lorsque la requête est exécutée, elle vous retourne ce message :

Ligne Trouvée

Par conséquent, vous êtes connecté sans posséder de compte. La protection pour résoudre ce type d'injection SQL consiste à utiliser les fonctions PHP qui vous sont proposées comme :

- addslashes()
- mysql_real_escape_string() pour MySQL
- pg_real_escape_string() pour PostgreSQL
- pdo_quote() pour PDO

Pour mettre en place ces filtres, vous devez effectuer l'opération suivante :

```
<?php
$login = addslashes($login);
$motpasse = addslashes($motpasse);
?>
```

Et par conséquent votre requête devient :

```
SELECT id FROM user WHERE login = '\\' OR 1 AND motpasse = '\\' OR 1
```

Avec un retour signalant qu'aucun résultat n'a été trouvé. Donc la personne ne pourra pas se connecter.

Arborescence

L'arborescence d'un site est souvent propre aux développeurs mais il est très facile d'accéder à un dossier sans que vous soyez invité à le visualiser, par exemple un dossier image, configuration... comme le montre l'image ci-contre.

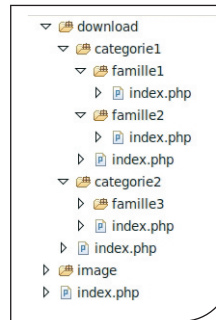
Imaginez que dans votre projet web vous proposez une page avec la possibilité de charger des documents ou des applications.

Pour réaliser l'opération, l'internaute doit remplir un formulaire pour laisser ses coordonnées et indiquer le choix du document. Cette opération vous permet d'obtenir des statistiques comme le nombre de téléchargements d'un document par rapport aux autres, le lieu, etc. Si tout un chacun peut accéder à la structure de votre site, les personnes vont pouvoir récupérer ces documents sans passer par votre formulaire, et vos statistiques seront fausses. C'est souvent le cas des images mais aussi du dossier config qui contient les codes d'accès de votre base de données ou du site internet. Les conséquences peuvent être encore plus dramatiques que vous ne l'imaginez. La solution à ce problème consiste à mettre un fichier « index.php » qui manque dans tous les dossiers. Le script de ce fichier se présente de la façon suivante :

```
<?php
header("Location: ../index.php");
?>
```

La fonction header permet en PHP d'envoyer un entête HTTP, c'est-à-dire de charger une nouvelle page web en perdant le chemin du bouton précédent. Si vous placez ce fichier dans tous les dossiers et sous-dossiers, la personne va essayer d'accéder à ce dossier et se retrouvera directement à la racine de votre site web et par conséquent à la page d'accueil. Pour tester cette protection, voici une URL qu'il est possible de trouver sur :

<http://votreSite/download/categorie/famille1/document.pdf>



Et maintenant essayez d'accéder directement à votre document comme le montre l'image ci-contre.

PHP_SELF

PHP_SELF est une variable qui retourne le script en cours d'exécution. Son but est de renvoyer le nom et le chemin du fichier en cours. La plus part du temps, celle-ci est utilisée dans le champ action d'un formulaire. Ainsi, vous ne vous occupez pas du nom de cette action lorsque vous concevez un formulaire.

La fonction PHP_SELF s'utilise sous cette forme :

```
<?php
echo $_SERVER['PHP_SELF'];
?>
```

Si cette variable se trouve dans la page suivante : <http://votreSite.fr/contact.php> ici vous obtiendrez en utilisant cette variable :

/contact.php

Mais pour utiliser cette variable dans un formulaire, celle-ci se place dans le champ action de votre <form> et se déclenche quand l'internaute aura appuyé sur le bouton envoyer du formulaire.

Bien sûr, l'utilisation de cette variable n'est pas obligatoire, mais si vous proposez la possibilité de laisser un commentaire en bas d'un article, vous ne pouvez pas connaître le nom de la page. Par conséquent cette variable vous sera très utile. Sa mise en place dans un formulaire, peut se présenter ainsi :

```
<?php
if (isset ($_POST) && (sizeof($_POST) > 0))
{
    echo "Votre saisie : ".$_POST['saisie']."<br />";
}
?>
<html>
<body>
<form method="POST" action="<?php echo $_SERVER['PHP_SELF']; ?>">
    Votre texte : <input type="text" name="saisie" />
    <input type="submit" value="Submit" />
</form>
</body>
</html>
```

L'exemple montre que dans une même page, vous avez une partie saisie et une autre traitement. Bien sûr, les données envoyées seront traitées si l'action a été déclenchée. Cependant cette variable est aussi un risque au niveau de la sécurité car elle peut subir des attaques du type XSS (voir plus haut). Mais vous pouvez sécuriser votre formulaire en utilisant la fonction htmlentities() et par conséquent la ligne <form> se présente ainsi.

```
<form method="POST" action="<?php echo htmlentities($_SERVER['PHP_SELF']); ?>">
```

Les sessions

Les sessions sont un des points qu'il ne faut pas négliger, car il y a un aspect classique pouvant être utilisé par tout le monde, et un autre utilisé par PHP.



Classique

Lorsque vous souhaitez utiliser des sessions avec votre navigateur, celui-ci peut stocker de nombreuses informations venant de l'internaute. Pour réduire les risques avec les cookies, vous pouvez stocker des informations concernant :

- Le navigateur
- L'IP
- La durée de validité

Par ailleurs, vous pouvez aussi utiliser différents noms de cookies permettant de définir s'il s'agit d'un internaute, d'une personne identifiée, d'un administrateur, etc. La multitude de cookies en général, et le stockage provisoire de nombreuses informations venant de l'internaute, vous permet de mieux filtrer les risques éventuels auxquels vous pouvez être confronté.

En PHP

Les sessions en PHP sont des sessions classiques et par conséquent une source d'informations non négligeable pour un pirate car elles reposent principalement sur un jeton unique pour reconnaître l'utilisateur au cours de sa navigation.

Les sessions peuvent subir différents types d'attaques :

- Le premier type concerne la prédiction qui est une technique permettant de voir les identifiants de session. Depuis de nombreuses années, l'identifiant de SESSION est généré automatiquement de façon aléatoire. Donc il y a très peu de risque qu'un pirate arrive à deviner un identifiant.
- Le deuxième type touche la capture et consiste à intercepter les identifiants de session lors de l'envoi d'un GET. Ce cas est souvent présent lorsque les utilisateurs n'acceptent pas les cookies. C'est pourquoi il est préférable d'utiliser l'envoi par le POST.
- Enfin, le troisième type va faire appel à une personne tierce en passant par la fixation de SESSION qui consiste à fixer l'identifiant de session avant que le serveur n'en attribue un.

Pour illustrer ce dernier cas, le pirate peut utiliser le scénario suivant. Le pirate va envoyer à une cible le lien de la page qui l'intéresse et ajouter à la fin de l'URL, une valeur de session comme ceci :

<http://votresite/login.php?PHPSESSID=666>

La personne qui reçoit ce lien, va cliquer dessus sans le vouloir. Celle-ci va saisir ses identifiants pour accéder à son compte. Le pirate revient quelque temps après en utilisant le même lien et se retrouve directement connecté sur le compte de sa victime.

La solution à ce problème consiste à modifier une ligne dans le fichier php.ini

```
session.use_trans_sid = 0
session.use_only_cookie = 1
```

De plus, vous pouvez activer un filtre par IP de l'internaute, ce qui limite beaucoup plus le risque de vol des sessions.

Les superglobales

Les superglobales sont des variables définies en PHP, ce qui signifie qu'elles sont disponibles quel que soit le contexte du script. Les variables superglobales sont \$GLOBALS, \$_SERVER, \$_GET, \$_POST, \$_FILES, \$_SESSION, \$_REQUEST, \$_ENV. Certaines ont été vues précédemment, mais les autres superglobales sont aussi importantes et il ne faut pas les négliger. Tout d'abord, vous pouvez sécuriser l'ensemble des superglobales en leur ajoutant la fonction htmlentities(), qui se compose de la façon suivante :

```
<?php
$pseudo = htmlentities($_COOKIE['pseudo'], ENT_QUOTES);
echo 'Votre pseudo est : '.$pseudo;
?>
```

L'exemple vous montre que vous imposez l'ajout de quotes concernant l'utilisation d'un cookie. Cependant si vous décidez d'utiliser la variable GLOBAL, de nombreux risques peuvent apparaître. Celle-ci se trouve souvent utilisée dans la récupération de valeurs dans les fonctions. Par exemple, si vous ne l'utilisez pas, voici quelques lignes de programmation :

```
<?php

$mag = "Programmez 2011";
afficher_titre();

function afficher_titre()
{
    echo 'Votre magazine est : '.$mag.' .';
}

?>
```

Le résultat que vous obtenez est

votre magazine est : .

Si vous utilisez la variable GLOBAL, vous pouvez obtenir ceci :

```
<?PHP
GLOBAL $mag;
echo 'Votre magazine est : '.$mag.' .';
?>
```

Pour obtenir le résultat suivant :

Votre magazine est : Programmez 2011.

Une des solutions existantes pour ne pas utiliser cette variable consiste à effectuer un point d'entrée dans votre fonction, c'est-à-dire :

```
<?php
$mag = "Programmez 2011";
afficher_titre($mag);

function afficher_titre($mag)
{
    echo 'Votre magazine est : '.$mag.' .';
}

?>
```

Le résultat est identique, mais vous êtes certain que cette variable est utilisée pour cette fonction, et non pour l'ensemble du site car il est toujours plus dur de sécuriser ce type de valeur quand elles sont accessibles sur l'ensemble du site web.

Conclusion

Cet article vous a montré les points principaux concernant les risques d'attaques que vous pouvez subir. On ne le dira jamais assez, la sécurité web est un des critères à prendre en compte et à ne pas négliger.

■ **Christophe Villeneuve**

Consultant pour Alter Way solutions, auteur du livre « PHP & MySQL-MySQLi-PDO, Construisez votre application », aux Éditions ENI. Rédacteur pour nexen.net, Trésorier AFUP et membre du LeMug.fr, PHPTV, PHPteam...

Sécuriser un site ASP.NET

Le groupe WhiteHat Security produit chaque année des rapports sur la sécurité des sites internet dont le top 10 des vulnérabilités les plus présentes sur le web (graphique ci-dessous). Dans cet article nous allons nous intéresser aux vulnérabilités de type injection SQL et Cross-Site Scripting ainsi qu'à la configuration générale des projets ASP.NET et des serveurs Windows dans le but de sécuriser au mieux un site web. A vous de jouer pour mieux protéger votre site web !

INJECTION SQL

Explications du risque

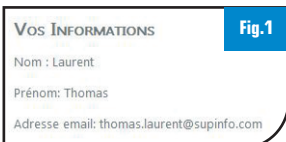
Une attaque de type injection SQL consiste à exploiter une faille dans une application pour envoyer à l'interpréteur SQL, d'une base de données utilisée par cette application, des données hostiles à travers une requête ou une commande. Cette attaque peut permettre d'accéder à des données sensibles ou d'exécuter des commandes inattendues contenues dans la base de données. C'est avec ce type d'attaque qu'en 2006, trois pirates, dont Albert Gonzalez, ont volé au total plus de 130 millions de numéros de cartes bleues à travers plusieurs actions sur différents sites. Une des entreprises visées, Heartland Payment Systems proposant des services de paiements, s'est vue ainsi dérober 100 millions de numéros de cartes de crédit. Sur un site web, les données hostiles peuvent être transmises à travers plusieurs sources d'entrées. Celles-ci incluent les champs de formulaire de données, les en-têtes « header » des requêtes HTTP ainsi que les paramètres passés dans les URL. Les bases de données relationnelles ne sont pas les seules cibles de ce type d'attaque. En effet, tout ce qui passe par un interpréteur est potentiellement vulnérable comme par exemple les technologies LDAP.

Exemple d'exploitation

Prenons un exemple simple de requête SQL :

```
SELECT * FROM users WHERE login = 't_laurent'
```

Un site web affichant les informations du profil d'un utilisateur va effectuer cette requête de manière dynamique en ajoutant « t_laurent » à la fin de celle-ci. La valeur du login provient d'un paramètre dans l'URL de la page et cette valeur n'est pas vérifiée avant d'être insérée dans la requête. Le résultat : [Fig.1].



VOS INFORMATIONS

Nom : Laurent

Prénom: Thomas

Adresse email: thomas.laurent@supinfo.com

Fig.1

Un pirate pourra entrer dans l'URL du code SQL « t_laurent OR 1=1 » au lieu d'entrer juste le login demandé, ce qui donnera comme requête :

```
SELECT * FROM users WHERE login = 't_laurent' OR 1=1
```



VOS INFORMATIONS

Nom : ContosoLeGrandLaurent

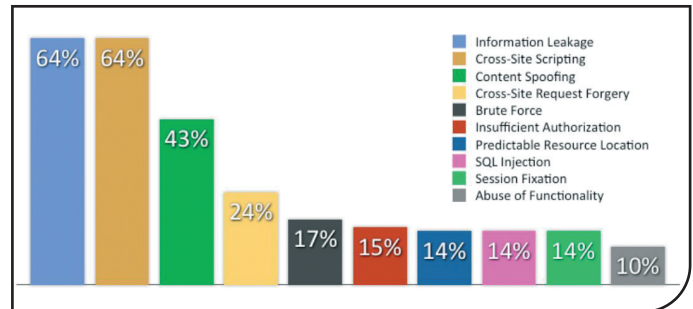
Prénom: GrégorySébastienThomas

Adresse email: greg@free.frsebastien@orange.frthomas.laurent@supinfo.com

Fig.2

Cette requête retournera alors tous les utilisateurs contenus dans la base de

données, peu importe leur login, et pourra donc potentiellement délivrer des informations qui ne devaient pas être accessibles à l'utilisateur. Le résultat : [Fig.2].



On remarque qu'ici le programme se servait des données retournées pour compléter les différents champs du profil de l'utilisateur et que l'injection a permis de récupérer les informations de tous les utilisateurs (3 dans cet exemple).

Solutions

Pour protéger au mieux son site web ASP.NET des injections SQL, différentes solutions sont à mettre en place quel que soit le type de sources d'entrées utilisé :

- Dans le cas de requêtes SQL dynamiques construites dans le code, il faut impérativement utiliser des paramètres nommés pour passer des valeurs à la requête, plutôt que d'utiliser la concaténation.
- Il faut toujours utiliser des valeurs fortement typées, comme par exemple, convertir en type Int la valeur d'un champ de formulaire lorsqu'on sait qu'il doit être un nombre entier au lieu de conserver cette valeur en type String. Si la conversion en type Int échoue, une erreur sera soulevée, ce qui empêchera l'injection.
- Une autre solution pour vérifier que l'utilisateur entre une donnée conforme est d'utiliser les validateurs disponibles en ASP.NET. L'un d'eux, le contrôle CompareValidator permet, entre autres, de vérifier le type de la valeur contenue dans un champ du site, exemple avec un type Int :

```
<asp:CompareValidator ID="ageValidator" runat="server" Error
Message="Age invalide" ControlToCompare="ageTb" Operator="Data
TypeCheck" Type="Integer" />
```

- Pour des entrées plus complexes, tel un champ d'adresses mail, la vérification de la bonne conformité du contenu du champ peut être effectuée via une expression régulière à l'aide du contrôle RegularExpressionValidator ou de la classe Regex, exemple :

```
<asp:RegularExpressionValidator ID="emailValidator" runat="
server" ErrorMessage="Email invalide" ControlToValidate="email
Tb" ValidationExpression="\w+([-+.' ]\w+)*@\w+([-+.' ]\w+)*\.
\w+([-+.' ]\w+)*" />
```

- Une des meilleures protections réside en l'application systématique du principe du moindre privilège au niveau des comptes d'accès à votre base de données. Le principe étant de créer un



compte n'ayant que les privilèges nécessaires pour répondre aux besoins d'un groupe d'utilisateurs et de créer autant de comptes que l'application en requiert.

XSS (CROSS-SITE SCRIPTING)

Explications du risque

Une attaque de type XSS (Cross-Site Scripting) consiste à exploiter une faille dans une application pour exécuter du script malicieux dans le navigateur de la victime. L'attaquant va en fait détourner une partie de l'application censée afficher de simples données pour y entrer du code qui sera exécuté par le navigateur. Cette attaque peut permettre de récupérer des informations sensibles sur le PC de la victime (sessions, cookies, etc.), de rediriger l'utilisateur sur un autre site, d'agir sur le site sous l'identité de l'utilisateur, de modifier les pages du site web pour le rendre inutilisable ou même de voler des informations à travers de faux formulaires. C'est à cause d'une attaque de ce type qu'en Septembre 2010 la timeline de Twitter est devenue inutilisable à cause de tweets devenus entièrement noirs, de lettres formats XXL ou encore de redirections automatiques sur des sites pornographiques. C'est suite à une mise à jour importante que la faille est apparue dans le traitement des tweets. Des pirates l'ont alors exploité pour injecter du code à l'intérieur de ces tweets. De manière générale, un script malicieux peut être injecté sur un site web à travers le dépôt d'un message, d'un commentaire ou bien même à l'intérieur de l'URL de la page web.

Exemple d'exploitation

Prenons l'exemple d'une injection à l'intérieur de l'URL d'une page web qui affiche un message personnalisé suivant le nom de l'utilisateur passé dans l'adresse du site.

```
http://www.contoso.fr/profil.aspx?nom=t_laurent
```

Un pirate pourra remplacer la valeur du paramètre « nom » de l'URL avec du code HTML, par exemple :

```
http://www.contoso.fr/profil.aspx?nom=<script>document.location='http://google.fr'</script>
```

Ce script malicieux va rediriger l'utilisateur vers un autre site, ici « google.fr », dès que la page aura été chargée par le navigateur. Pour être encore plus discret, le pirate peut encoder le script passé en paramètre dans l'URL pour obtenir ceci :

```
http://www.contoso.fr/profil.aspx?nom=%3C%73%63%72%69%70%74%3E%64%6F%63%75%6D%65%6E%74%2E%6C%6F%63%61%74%69%6F%6E%3D%27%68
```

```
%74%74%70%3A%2F%2F%67%6F%6F%67%6C%65%2E%66%72%27%3C%2F%73%63%72%69%70%74%3E
```

Solutions

Aujourd'hui, avec la version 4 d'ASP.NET les sites web sont par défaut protégés face aux attaques XSS grâce à la validation des requêtes qui est activée par défaut sur toutes les pages Aspx. Lorsque cette protection est active, l'application web va automatiquement valider les différentes entrées des pages et soulever une erreur en cas de détections de scripts ou de codes HTML potentiellement dangereux ainsi qu'afficher une page d'erreur 500.

Néanmoins, cette protection est parfois trop contraignante et peut alors être désactivée sur certaines pages :

```
<%@ Page ValidateRequest=false %>
```

Dans ce cas-là, le traitement peut-être fait manuellement en 2 étapes :

- De la même manière qu'avec l'injection SQL on peut vérifier la conformité des valeurs en entrées à l'aide, entre autres, des validateurs.
- Ensuite, il faut encoder les données et pour cela Microsoft fournit gratuitement sur Codeplex la librairie Anti-XSS dans sa version 4.0 : <http://wpl.codeplex.com/>

Utilisation d'Anti-XSS

Cette librairie fournit plusieurs fonctions qui permettent d'appliquer un traitement sur des données et qui vont encoder tous les caractères non autorisés, car potentiellement dangereux, suivant le type d'entrée choisi : HTML, XML, CSS, JavaScript ou LDAP. Pour démontrer l'utilisation de cette librairie, prenons l'exemple d'un pop-up JavaScript qui affiche un message personnalisé avec le code suivant :

```
var nom = Encoder.JavaScriptEncode(nomLb.Text, false);
var script = "<script>alert('« + nom + »');</script>";
ClientScript.RegisterClientScriptBlock(GetType(), «jspopup», script);
```

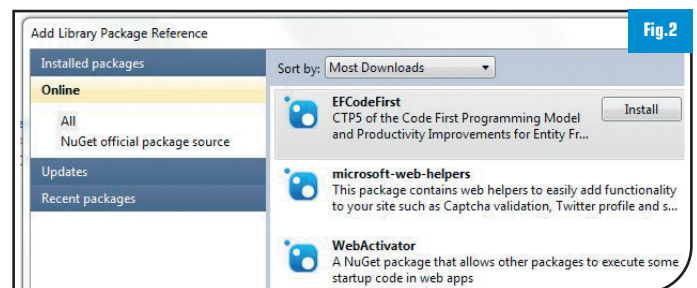


Fig.2

General Application Rules			
Title	Severity	Result	Resolution
Drive "C:\\" is NTFS volume	1	Passed	
Windows Guest account is renamed on the server	2	Indeterminate	
Windows Guest account is disabled	1	Indeterminate	
Root Drive "DeviceID\" does not have Read and Execute permissions for Guest, Everyone and Authenticated Users Group	1	Indeterminate	
Windows shares on the system drive are removed (FAILED)	2	Failed	Remove all shares from the system drive.
Access to required share "C:\Windows\SYSVOL\sysvol\AD2008R2.loc\SCRIPTS" is configured to least privilege (FAILED)	1	Failed	<ul style="list-style-type: none"> • Reduce share membership to user list. • Do not grant permissions to system accounts and gr IIS_Usrs.

Fig.3

Comme vous le voyez, l'entrée potentiellement dangereuse vient ici du champ « nomLb » qui est censé contenir un nom mais qui pourrait contenir du code malicieux JavaScript.

C'est pourquoi les données contenues dans ce champ ne sont pas directement injectées dans le script du pop-up mais sont d'abord encodées à l'aide de la méthode JavaScriptEncode de la classe Encoder fournie par Anti Xss.

SÉCURISER VOS PROJETS ASP.NET

Mettre à jour vos librairies avec NuGet

Garder à jour les différentes librairies externes que l'on utilise dans un projet n'est pas évident et pourtant c'est indispensable pour que notre application reste sécurisée tout au long de sa vie. NuGet est une extension de Visual Studio qui permet de répondre à cette problématique. En effet, cette extension propose d'ajouter, de supprimer mais surtout de mettre à jour les librairies utilisées dans vos projets, vous pouvez télécharger cette extension sur Codeplex : <http://nuget.codeplex.com> [Fig.2].

Messages d'erreurs personnalisés

Plus le fonctionnement d'une application est transparent pour l'utilisateur plus celle-ci est vulnérable. Par défaut, les sites ASP.NET affichent des messages d'erreur divulguant énormément d'informations sur l'application et sur son code. Lors du déploiement d'une application, il est impératif de retirer ces pages d'erreurs pour afficher à la place une page d'erreur plus simple qui ne divulgue pas d'informations sensibles. Pour activer cette fonctionnalité il faut ajouter dans la balise system.web du fichier Web.config :

```
<customErrors mode="On" redirectMode="ResponseRewrite" defaultRedirect="~/Error.aspx" />
```

La page Error.aspx devant être créée auparavant dans votre projet.

Désactiver le Tracing et le Debugging en déploiement

Autre point sensible parfois oublié, les fonctionnalités de Tracing et de Debugging d'ASP.NET doivent être désactivées lors du déploiement. Une fois de plus il s'agit de modifier la balise system.web du fichier Web.config avec :

```
<trace enabled="true" localOnly="true" />
<compilation debug="false" targetFramework="4.0" />
```

SÉCURISER VOTRE SERVEUR AVEC WACA

Enfin, le dernier point à ne surtout pas négliger c'est la sécurité du serveur hébergeant notre site web. Microsoft propose depuis peu un outil permettant justement d'analyser rapidement la sécurité d'un serveur Windows et de générer des rapports indiquant ce qui doit être amélioré. Ce logiciel se nomme Microsoft Web Application Configuration Analyzer (WACA) et est téléchargeable gratuitement sur le site Microsoft Download Center; voici un exemple type de rapport généré par cet outil : [Fig.3].

Maintenir la sécurité d'un site internet est un combat de tous les jours, néanmoins Microsoft apporte des solutions efficaces pour protéger les sites ASP.NET. Nous n'avons vu qu'une partie des vulnérabilités existantes, pour aller plus loin je peux vous conseiller le site owasp.org qui traite des nombreuses autres vulnérabilités.

■ Thomas Laurent - Formateur SUPINFO - @t_laurent

éligibles
au DIF

Formations DÉVELOPPEMENT

Notre catalogue de formation axé sur le développement d'applications entreprises est structuré en 4 catégories : les concepts de développement et modélisation, la technologie Java, les applications Microsoft, et le développement en environnement Open source. **Préparez efficacement vos projets!**

Concepts

- Initiation à la programmation objet 5-6/05 • 16-17/06
- XML prise en main 20-22/04 • 25-27/05
- UML, concepts et mise en oeuvre 16-19/05 • 6-9/06

Java, JEE, C++

- Java pour développeur objet 23-26/05
- Java, Développer un client riche avec Swing 3-5/05
- Programmation Objet en C++ 4-8/04 • 2-6/05
- C++ Programmation efficace et avancée 6-9/06
- JavaScript, Développer en HTML dynamique 6-8/06
- Java et XML 19-20/05
- Hibernate, framework Open Source 23-25/05
- Développer avec Spring 27-29/06
- Développer des applications Web avec Struts 1 20-22/06
- Tests Java: Méthode de plans de tests et jeux d'essais avec Java EE 27-28/06
- Sécurité Java 30-31/05

.Net

- Programmation C# avec Microsoft .Net Framework 4 4-8/04 • 6-10/06
- Développer des applications Web avec les formulaires ASP.Net 11-15/04 • 4-8/07
- Développer des applications Web avec Visual Studio 2010 2-6/05 • 20-24/06
- Développer des solutions d'accès aux données avec VS 2010 23-27/05
- Développer des solutions WCF avec VS 2010 9-11/05 • 16-17/05 • 22-24/06

Open Source

- Développement système sous Linux ou Unix 20-23/06
- Linux temps réel et embarqué, Programmation 20-23/06
- Maîtriser la programmation objet en PHP NC
- PHP 5 : développer un site Web dynamique 14-17/06
- Programmation Shell 15-17/06

Pour nous contacter, composez le
0821 20 25 00 (prix d'un appel local) ou
posez-nous vos questions par email :
info@globalknowledge.fr

www.globalknowledge.fr



Global Knowledge

C++ plus que jamais d'actualité !

Donné pour mort il y a moins de 10 ans, on n'avait alors que Java à la bouche ! Puis C# ou encore PHP, Ruby, Python, Javascript, les frameworks... Le C++ est un peu comme le scorpion. Mettez-le au cœur du feu, dans l'eau, dans le désert, il survivra toujours. En 12 ans de Programmez !, nous avons tout entendu, tout écrit, sur C++ : dépassé, trop vieux, inutile. Mais, inlassablement, il revient. Il se tapit dans l'ombre, parfois.

Pourquoi aujourd'hui, le C++, et même le C, redevient-il LE langage de référence ? C et C++ sont les seuls langages à être natifs et de performances inégalées. Non, Java et C# ne pourront jamais être aussi optimisés et performants qu'un code C++. Pour l'embarqué, l'enfoui, C++ demeure incontournable pour réellement faire du bas niveau. C demeure au cœur des systèmes d'exploitation pour des raisons de performances. Pour les applications

hautement critiques et temps réel dur, C++ ne peut pas être remplacé. Et même en téléphonie, le langage objet reste la référence absolue. Et que dire de la 3D, des interfaces de type MFC, Win32 et surtout dans le massivement parallèle, le HPC ? Là encore, point de salut en dehors des langages natifs !

Tout n'a pas forcément besoin d'être codé en pur natif, vous pouvez parfaitement faire le « core code » en langage managé et des portions très précises, des fonctions en natif. D'autre part, maîtriser le C++, c'est aussi mieux maîtriser la programmation et ses subtilités...

Dans ce dossier, plusieurs experts et développeurs C++ donnent leur avis, le comparent à d'autres langages. On oublie souvent que le développement consiste aussi à choisir le bon langage...

■ François Tonic

Le C++ est-il redevenu le langage incontournable ?

Pour pouvoir répondre à cette question, revenons un peu en arrière dans le monde des compilateurs Microsoft.

Microsoft n'a inventé ni le langage C ni le C++ comme vous vous en doutez, mais depuis 1983 date à laquelle il propose son premier compilateur C, puis son premier compilateur C/C++ 7.0 avec un environnement tournant sous Dos, Microsoft n'a eu de cesse d'améliorer à la fois son compilateur et son environnement de développement. 1993 a vu l'arrivée de Visual C/C++ 1.0 avec un environnement de développement basé sur Windows, et ainsi jusqu'à Visual C++ 6.0 en 1998.

Les années 2000, sont un tournant pour C/C++ chez Microsoft, mais pas pour les raisons qu'on imagine. En effet, depuis l'arrivée de la plateforme .NET et de Visual Studio .NET "le mal nommé" (et les déboires des développeurs Visual Basic 6.0 avec cette nouvelle plateforme, la communauté des développeurs C/C++ a cru à tort que le langage C++ était mort ou plus à la mode, du moins chez Microsoft. En réalité, Microsoft n'a jamais autant investi dans la norme ISO/ANSI C++ qu'à partir de cette époque.

En embauchant d'une part les célèbres **Herb Sutter** secrétaire du comité de normalisation du C++, et **Stanley Lippman**, qui a commencé à travailler sur le langage C++ avec son inventeur **Bjarne Stroustrup**, (Lippman travaille désormais à Emergent Game technologies), non seulement Microsoft s'est rapproché au plus près de cette norme, avec un taux de 98% et continue à investir dans la prochaine version C++OX, mais propose également un langage C++/CLI, le langage de transition, la passerelle entre le monde natif et le monde .NET.

Alors est-ce que le C++ est redevenu le langage incontournable, je répondrai (et ceci n'engage que moi), non bien sûr !

Il n'est pas redevenu incontournable, car il l'a toujours été. Les très grands logiciels, sont majoritairement développés dans ce langage depuis des années et le seront sans doute pour des années à venir encore avec un langage et des environnements de développement qui vont évoluer, pour plus de simplicité, de productivité et de qualité, les nouveautés de la norme ISO/ANSI C++Ox et

l'évolution des bibliothèques en sont la preuve vivante.

Alors c'est vrai qu'aujourd'hui il jouit peut-être d'une visibilité accrue, porté par des périphériques à la mode dans le grand public et que l'on peut apparenter à une renaissance (<http://channel9.msdn.com/Shows/Going+Deep/Craig-Symonds-and-Mohsen-Agsen-C-Renaissance>), mais le C/C++ n'est pas, ou plutôt n'est plus un phénomène de mode. Lorsqu'on décide de partir sur ce langage c'est pour longtemps et pas sur un coup de tête, du moins lorsqu'on a l'idée de développer des applications, qui vont durer longtemps.

D'ailleurs, le développeur C++ est comme dame nature, il prend le temps, il est patient, peaufine, raffine, et optimise son code à l'extrême. Il s'investit dans la compréhension de la plateforme sur laquelle il développe. Il veut en comprendre les rouages, pour mieux l'exploiter. Aujourd'hui, il a peut-être plusieurs millions de lignes de codes dans la nature, testées, éprouvées, peaufinées depuis des années et cela lui paraît impensable d'y mettre un terme parce que de nouvelles plateformes plus simples, plus rentables et plus productives sont disponibles.

Quand je parle de développeurs C++, j'inclus également les développeurs Microsoft. Pensez-vous honnêtement que des logiciels comme Excel ou Word sur le poste client, pourraient-ils être redéveloppés en .NET ? Pas si sûr, pas tant que l'avènement d'un nouveau système d'exploitation basé sur .NET n'ait vu le jour, s'il le voit jamais ! D'autant que nous avons mis plus de 12 ans pour passer tout Office de C à C++.

Je ne suis pas en train de stigmatiser une population par rapport à une autre, à chaque objectif son outil. A chaque type de développeur son outil de développement. Faut-il développer un site Web en C/C++ ? Faut-il développer un driver de carte graphique en C# ?

Attention, je reste persuadé que l'on peut développer de très grosses applications clientes de qualité sur la plateforme .NET en WPF, et qu'un développeur C++ s'y mettrait rapidement. Il est même possible de créer

de la valeur ajoutée autour de son application C++ en construisant des add-in basés sur cette dernière (Visual Studio 2010 avec son éditeur WPF en est une belle illustration), mais de nombreux développeurs natifs, ne souhaitent pas avoir d'adhérence avec cette plateforme.

Microsoft l'a très bien compris et a ajouté dans Windows 7 pléthore de nouvelles API Win32 pour permettre à ses développeurs de moderniser leurs applications et attirer à l'occasion de nouveaux développeurs sur C/C++. Les rubans, les services Web, la gestion simple de l'accélération hardware et des animations, le multitouche et autres nouveautés qui font que développer en C/C++ n'a plus rien à envier à la plateforme .NET.

Alors, avec un titre somme toute accrocheur, C++ n'est pas redevenu incontournable, il l'a toujours été et ceci quelle que soit la plateforme. Tant que les systèmes d'exploitation seront développés en C/C++ et proposeront des API pour y accéder le C/C++ a de beaux jours devant lui et je me félicite qu'aujourd'hui d'autres plateformes proposent un modèle de développement autour de ce langage, je passe moins maintenant pour un vieux rabat-joie!

Néanmoins, car il y a toujours un risque, les développeurs C++, ne sont pas immortels et si personne n'est là pour relever le gant, sous-entendu si nos chères écoles ne l'enseignent plus, alors oui le C/C++ est sans doute voué à mourir et les applications avec, car trop cher à maintenir. Mais qui peut prévoir l'avenir ? Et ne nous voilons pas la face non plus, les environnements managés type Java et .NET possèdent de beaux atouts.

Mais je reste optimiste, car récemment un développeur .NET plutôt orienté Web, m'a demandé conseil sur le passage d'un pointeur de fonction rien de moins. Je lui demande sur quel OS il est ? Il me répond BADA, système sur lequel il semblerait que C, C++ soit incontournable.

■ **Eric Vernié**

Microsoft France

Relations techniques avec les développeurs

Le C++ au quotidien

De nos jours, la plate-forme .Net est devenue incontournable. Il devient de plus en plus difficile de trouver des développeurs maîtrisant le C++. Quelles bonnes raisons pousseraient une entreprise à continuer à maintenir sa plate-forme de développement plutôt que de passer à .Net ?

Tout d'abord, le C++ est un langage de programmation à haut niveau de performance que le C# n'atteint pas aujourd'hui. Ensuite, le savoir-faire accumulé au sein d'une entreprise, représenté par l'ensemble des bibliothèques métier représente une telle quantité de travail, qu'il n'est pas envisageable d'abandonner ou de porter l'intégralité de l'existant dans un autre langage, quelles que soient les performances de ce dernier.

Trouver un équilibre

Ce ne sont pas pour autant des raisons suffisantes pour que l'ensemble d'un système soit implémenté exclusivement en C++. Ce langage n'est pas forcément adapté à toutes les problématiques rencontrées alors que d'autres langages les adresseront de manière plus efficace. La plateforme .Net trouve toute sa force dans l'interopérabilité entre les langages (C#, Ruby, Python, C++ ...). Il permet aux développeurs de programmer dans leur langage de prédilection sans avoir à s'inquiéter des problématiques d'intégration. De plus c'est un framework extrêmement bien structuré, facilitant l'implémentation de systèmes complexes, qui serait sans nul doute plus difficile à réaliser et à maintenir en C++. Tout ceci est vrai tant que l'on reste dans un contexte purement managé. Qu'en est-il lorsqu'il s'agit de coopérer avec du C++ natif ?

Au premier abord, la tâche semble complexe et difficile à mettre en œuvre. Cet article présente différents outils à disposition du développeur pour intégrer du code C++ natif dans un environnement .Net.

Imaginons une entreprise ayant une librairie C++ de gestion de licence, utilisée dans une suite logicielle existante. Ce composant est 100% fonctionnel, et a été éprouvé depuis de nombreuses années. Un nouveau produit arrivant sur le marché doit utiliser ce composant, mais en mode «connecté». Ce produit implémenté en C#, doit se connecter à un serveur pour valider son exécution.

Côté client la tâche est plutôt triviale, et consiste à générer un stub wsdl (ou d'utiliser WCF), d'effectuer l'appel au service et de montrer patte blanche. Côté serveur, une page asp.net pour implémenter le service web (ou un service Window en WCF) paraît une solution pratique et rapide à mettre en place. Comment le service d'activation va-t-il exécuter le code C++ de la librairie historique ?

Techniques d'interopérabilité

Plusieurs approches sont possibles :

- Créer une assembly C++/CLI, embarquant la librairie de licences native en son sein.
- Créer une dll native et effectuer les appels aux points d'entrée via DllImport.
- Utiliser un wrapper automatique tel que SWIG.

Le choix dépend surtout des contraintes de développement (portabilité, temps consacré à l'intégration, état de l'existant...).

	Avantages	Inconvénients
C++/CLI	<ul style="list-style-type: none"> - Une seule Dll à déployer. - Débogage facilité. - Utilisation directe des classes natives. 	<ul style="list-style-type: none"> - Windows uniquement - 32/64 bits dépendant - Marshaling manuel
Dll native	<ul style="list-style-type: none"> - Cycle de vie des objets maîtrisé - Portabilité - Marshaling simplifié. 	<ul style="list-style-type: none"> - Fonction C (Flat API). - Dll supplémentaires. - Débogage complexe.
SWIG	<ul style="list-style-type: none"> - Rapide à mettre en place - Permet de cibler d'autres frameworks - Marshaling automatique 	<ul style="list-style-type: none"> - Sensible à la qualité des entêtes - Types trop complexes - Difficile à maintenir

L'utilisation de Dll natives permet de bien cloisonner le fonctionnel bas niveau des couches supérieures. La conversion des arguments est largement simplifiée grâce à l'utilisation des attributs de fonctions DllImport. Cette technique fonctionne aussi bien sous Windows, MacOS ou Linux. En contrepartie il faut implémenter et maintenir l'interface entre .Net et le code Natif sous la forme d'une flat API. La multiplication du nombre de Dll peut poser des problèmes de déploiement si leur gestion n'est pas maîtrisée.

L'utilisation de SWIG est envisageable dans le cas où le nombre de classes et de fonctions à exporter devient trop important pour qu'il soit géré manuellement. Pour ce faire, l'interface publique exportée doit être la plus standard possible, l'utilisation d'objets de type stl est à proscrire. Des problèmes de performance surviennent lorsque les appels aux fonctions natives deviennent trop nombreux. Le coût du P/Invoke n'est alors plus négligeable, le cycle de vie des objets n'est plus maîtrisé, l'appel à la méthode Dispose des wrappers de classes doit être effectué dans tout le code client et devient invasif.

L'avantage principal des Dll C++/CLI est l'utilisation directe des classes C++ natives sans passer par une flat API. Il est également possible d'accéder à des bibliothèques managées pour appeler des méthodes ou des Delegate directement depuis l'assembly C++. Le risque est de construire des systèmes interdépendants difficiles à maintenir et à déboguer. L'essentiel du marshaling doit être fait manuellement.

L'utilisation des Dll native est pour ma part la meilleure des trois approches. La mise en place est relativement simple et avec un marshaling quasi automatisé. De par leur structure elles imposent une organisation par couche.

Cependant, la construction d'une Flat API est une activité laborieuse, qui ne permet pas d'exporter des classes entières ni même les espaces de nom. Un nommage systématique des fonctions permet d'éviter les collisions de noms et d'organiser l'API de manière logique à la fois du côté natif comme du côté managé et facilite la maintenance.

L'exemple d'une Flat API exposant une classe permet de se faire une idée de la gestion de ressources native depuis une librairie écrite en C#. La partie native ("Image.h/cpp") présente dans une section C les fonctions relatives à une classe Image. On y retrouve l'équivalent

des constructeur/destructeurs, accesseurs et méthodes publiques de la classe. La partie managée ("Image.cs") reprend cette même API dans une classe Image statique cloisonnée dans un espace de nom "Native". Une classe Image dans l'espace de nom principal de la librairie permet de gérer le cycle de vie des instances non managées. Ce wrapper héberge un handle de la classe native et implémente IDisposable pour libérer la ressource non managée. Les accesseurs sont implémentés via des propriétés C#.

Une fois cette classe implémentée son utilisation est standard en .Net, l'utilisation de la portée "using" en C# permet de garantir la libération une fois qu'il n'est plus utilisé. Dans l'exemple, le wrapper fait 4 appels à P/Invoke pour 2 lignes de code.

Il ne s'agit pas non plus d'exporter l'ensemble des classes et méthodes d'une librairie, mais d'exporter les points d'entrées d'une façade (patron de conception), masquant la complexité des sous-systèmes. Cela évite également de limiter le nombre de P/Invoke. L'interface publique s'en trouve simplifiée, les sous-systèmes plus faciles à valider et à maintenir. Il est toujours plus facile de complexifier le problème côté .Net (threads, event, asynchrone...).

Le Marshaling

Quelle que soit la technique employée, le marshaling est une des clés de l'interopérabilité managé/natif. Il est essentiel de comprendre dans le détail ce mécanisme fourni par le framework .Net. Il permet la conversion de types entre les deux mondes. Chaîne de caractère, structures de données, callbacks... ; il est nécessaire de passer des arguments aux fonctions et de récupérer en retour les résultats.

Chaînes de caractères

Pour les chaînes de caractères, il faut savoir sous quel format elles sont utilisées dans le code non managé. ASCII, Utf-8, Utf-16 ou MBCS, il faut convenir de l'encodage à employer. Lors de l'utilisation deDllImport il faut utiliser l'attribut MarshalAs afin de transmettre le bon pointeur à la librairie native. La transmission d'un encodage particulier est plus délicate. Pour être sûr de l'encodage (Utf-8 par exemple), il est préférable de transmettre des IntPtr en argument de la fonction native, la librairie retrouvera le bon type par convention :

```
[DllImport("NativeLib.dll")]
public static extern void SendUtf8String(IntPtr utf8);
```

La conversion d'une chaîne managée en pointeur Utf-8 peut s'effectuer via des portées unsafe et fixed :

```
static void SendUtf8String(string str)
{
    unsafe
    {
        byte[] buffer = System.Text.Encoding.UTF8.GetBytes(str);
        fixed (byte* pBuffer = buffer)
        {
            IntPtr utf8 = new IntPtr(pBuffer);
            SendUtf8String(utf8);
        }
    }
}
```

Concernant le marshaling en C++/CLI, l'en-tête <vcclr.h> permet d'obtenir le buffer interne d'une chaîne managée :

```
#include "vcclr.h"

String^ str;
pin_ptr<const wchar_t> wch = PtrToStringChars(str);
std::wstring wtr = wch;
```

Structures de données

Concernant le marshaling de structure de données, il est essentiel de reproduire les structures natives et managées avec les mêmes alignements. Pour cela, l'attribut StructLayout permet de définir correctement les structures managées. La structure native doit être de type POD (Plain Old Data). La récupération de l'adresse d'une structure managée se fait via GCHandle et la méthode ToIntPtr(). Dans le cas d'échange de tableau de structures, il est nécessaire que la taille des structures soit identique.

```
GCHandle gch = GCHandle.Alloc(myStruct);
IntPtr ptr = GCHandle.ToIntPtr(gch);
gch.Free();
```

Ces contraintes permettent d'éviter de nombreux bugs et la méthode est similaire que ce soit via DllImport ou avec C++/CLI.

Callback

La transmission de callback s'effectue via la récupération d'un pointeur de fonction associé à une delegate managée. Cela s'effectue grâce la méthode Marshal.GetFunctionPointerForDelegate. Ce pointeur est compatible stdcall, il suffit donc de le transmettre au code natif et de le convertir en callback C.

```
public delegate void MouseMove(int x, int y);
static void OnMouseMove(int x, int y) {
    Console.WriteLine("MouseMove: {0} {1}");
}
MouseMove mouseMove = new MouseMove(OnMouseMove);
GCHandle gch = GCHandle.Alloc(mouseMove);
IntPtr mouseMoveHandle = Marshal.GetFunctionPointerForDelegate(
    mouseMove);
gch.Free();
```

L'utilisation d'un GCHandle permet de conserver une référence managée de la delegate. Cet objet ne doit pas être libéré tant que la couche native possède une copie de ce pointeur de fonction.

Conclusion

L'interopérabilité managé/natif est à mon sens indispensable de nos jours. Cela facilite l'intégration et la réutilisation de code C++ par des personnes n'ayant pas a priori la connaissance du langage. On trouve aujourd'hui de nombreuses librairies de qualité (codecs, traitement d'image, compression, moteurs 3D, ...) pouvant ainsi être exploitées directement en .Net. La double compétence C++/C# permet au développeur de s'adapter et de répondre efficacement aux problématiques qu'il rencontre quotidiennement.

■ Michel Foucault - Développeur indépendant
michel.foucault@codhex.fr

Comparaison d'applications équivalentes C# et C++

Je vais prendre comme support pour cet article deux missions que j'ai eu l'opportunité de réaliser en C++ et en C#. La première sous la direction de Xavier C., chef de projet pour Microsoft Services. La seconde sous la direction de Joël D., chef de projet au CACIB. Ces deux missions sont séparées de plus de 10 ans et pourtant, leur contenu est sensiblement le même. Il s'agit, sur une partie de chaque mission, de réaliser une application graphique contenant des modules ou plug-ins.

Après mures réflexions, pour chaque application, il faut une gestion de profils et une sécurité minimum et tracer les actions au maximum pour des raisons d'exploitation. La première application est une application de gestion de prospect pour l'immobilier et la simulation de crédit. La seconde est une application de monitoring [Fig.1 et 2] pour la gestion des processus de réconciliation des opérations FO/BO pour la finance qui permettra de visualiser ce qui s'est passé la nuit en termes de lancement de Workflows fonctionnels qui transforment les données sur les instruments financiers et les opérations de bourse.

Le photomontage [Fig.2] avait pour but de vendre au client la migration de l'application de monitoring de C# vers C++ avec Visual Studio 2008 sp1. Les premiers retours de la visualisation de cette maquette furent excellents. Un futur manager voulait cette application et considérait que la première version de l'application de monitoring était trop pauvre, visuellement. Adepte des librairies tierces comme DevExpress, ce mana-

ger voulait absolument introduire DevExpress dans l'application pour surcharger graphiquement l'application avec un Ribbon comme dans Office 2007. J'ai donc procédé à un photomontage entre une application Visual C++ MFC qui supporte le Ribbon et l'application .NET Windows Forms pour vendre ma solution. La réaction a été la suivante : « *ah c'est en C/C++, c'est dommage* ». D'autres réactions ont été plus contrastées « *C++ est interdit sur le projet – je vais en parler à Qualité Méthode et Architecte, le service qui valide les technologies* »

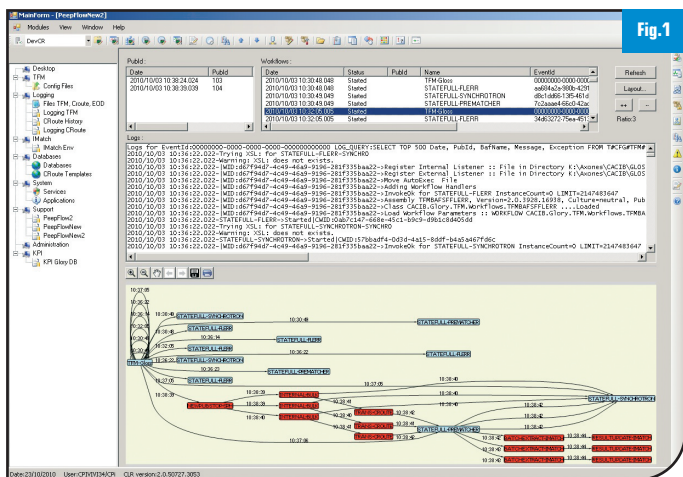
En 1998, Xavier, jeune chef de projet chez Microsoft me charge de réaliser un framework pour les fondations d'une application de gestion de prospects [Fig.3]. Cette application est faite en Visual C++ 4.2 sous Windows NT 4 Option Pack 4 avec un accès à un site central IBM AS 400. L'ensemble de la solution est une solution client-serveur avec une utilisation des réseaux SNA. A l'époque, nous utilisons SNA Server pour la gestion du protocole SNA et la gestion des transactions avec le système IBM. La partie

visible de l'application est un module graphique qui fait de nombreuses opérations en local et à distance sur le serveur central. Pourquoi comparer une application C# avec C++ ? Les deux sont équivalentes car codées de la même façon. L'application C# contient une fenêtre (MainFrame) qui présente un treeview sur la gauche et propose à l'utilisateur de lancer des modules ou plug-ins. Chaque plug-in s'insère dans une gestion de fenêtre MDI (de multiples fenêtres). L'application C++ n'est pas MDI mais SDI – cela ne change pas grand-chose avec les MFC.

La méthode principale des deux applications est la même et se nomme `CreateOrReplaceView` et prend en paramètre le nom du plug-in à afficher.

C++, langage de l'innovation

Pourquoi C++ est-il le langage de l'innovation ?
Mon application compilée en 1998 se lance
toujours aujourd'hui... le binaire n'a pas
demandé de framework à installer ni de run-
time spécifique car tout est déjà livré en



WinForm .NET 2.0 Application de Monitoring avec MS-Glee (2010)

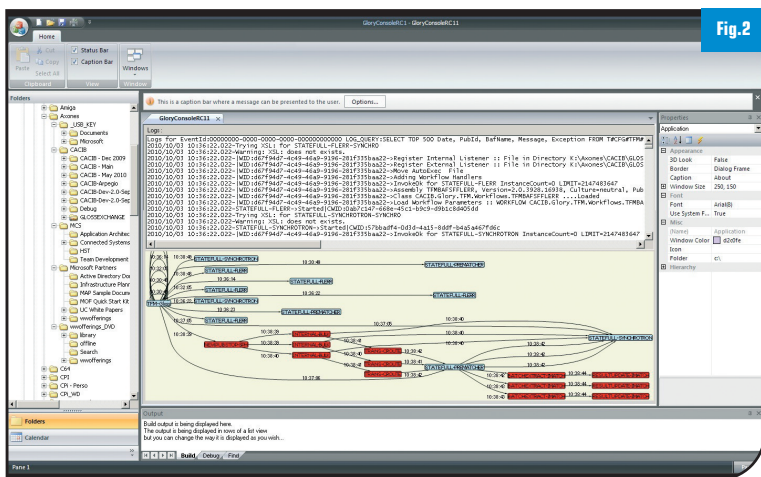


Photo-montage d'une application MFC en C++ de Monitoring avec MS-Glee (2010)

standard par le système d'exploitation et il s'agit juste de la DLL des MFC MFC42.DLL. Cette librairie est utilisée dans Windows et donc naturellement distribuée. Pourquoi mon application créée en 2010 ne veut pas se lancer et me demande un framework 3.0 et framework 2.0 que je ne possède pas sur l'ordinateur avec lequel j'écris cet article ? Et pourtant tout a été écrit en 2010 ! Honteux ! Voici l'explication. En C++, le concepteur de la solution ne sait pas sur quelle plateforme sa solution va tourner et il peut y avoir de très faibles contraintes au départ du développement de l'application. Vous êtes sur un socle Windows, et les API Win32 sont communes, pour la grande majorité des applications à de nombreuses versions de Windows (NT4, 2000, XP, 2003, 2008, etc). MFC42.DLL est toujours redistribué alors que plus aucun outil de développement ne target MFC42.DLL, pourquoi ? Un indice. Tout ce qui est fait n'est plus à faire. En C++.

Deprecated vs C++ ?

Le raisonnement est moins compliqué que dans d'autres langages car je m'astreins à livrer une solution qui fonctionne dans un environnement donné et que cet environnement ne sera jamais deprecated. Les API Windows Win32 ne sont jamais deprecated mais enrichies. Exemple : fonction `CreateThread()`, fonction `_beginthreadex()`. Les deux fonctions permettent de gérer des threads et pourtant aucune des deux ne serait supprimée de `Windows.h` car elle a été livrée. Oui vous avez bien entendu, une API désignée en 1992 et livrée sous Windows NT 3.51 comme `CreateThread` est toujours valable sous Windows 7. Et pourtant, il y a de nombreuses façons de créer et de gérer les threads. Pourquoi cette fonction

n'a pas été touchée ? La réponse est simple : une librairie est faite pour être utilisée et à partir du moment où elle est livrée, elle est immortelle.

En 1974, le langage C a permis de fabriquer Unix. En 2011, je peux toujours faire un `printf`(« hello world ») ; même si les puristes du C++ diront qu'il faut faire un `std::cout << « hello world cpp » << endl` ; Est-ce que les programmeurs C++ ont plus de respect pour les anciens ? Je ne crois pas que les programmeurs C++ soient câblés différemment des programmeurs C#. On a des habitudes, on a des outils, on a des mythes et des croyances. Il y a des légendes dans toutes les civilisations.

Exemple de projet critique en C++

Comment le design des applications a-t-il évolué depuis 10 ans ? Pour être parfaitement honnête avec vous, je vais me replonger dans un programme de communication Windows Server / Linux Redhat que j'ai réalisé pour une entreprise, appelons là, le Château de JacquesC. Les dépêches d'agence arrivent sur un télex, et anciennement les dépêches sont imprimées au kilomètre. Dans les années 90, les prises série sont enfichées dans une carte série 4 ports sous une machine Linux et les dépêches sont présentées sur un client lourd sous Windows fait en MFC, car c'est un framework qui permet de développer à moindre coût des applications graphiques sous Windows. Dans les années 2000, une connaissance me propose de mettre la main à la pâte pour réécrire une application sensible. Je réécris l'application de communication sous Windows ainsi que la partie graphique sur un client léger sous forme d'un site Web en ASP/COM. La

mission est compliquée car il s'agit de présenter des informations critiques à des gens qui occupent des fonctions élevées au sommet de l'état. A l'époque, les plateformes Microsoft proposent Visual Studio 6.0 comme compilateur C++ et on m'impose une solution à base de client léger pour des impératifs de déploiement. C'est là que j'ai eu des craintes, car comment faire de l'interop entre une machine Linux, Windows et un client léger. C++ m'a sauvé. Si je ne réussissais pas ma mission, on m'a fait comprendre que je finirais ma carrière dans un bagne à Cayenne donc j'ai fait très attention sur mes choix technologiques ! Quels choix s'imposaient à moi ? Commençons par les contraintes :

- Récupération d'une application existante
- Réécriture à moindre coût
- Disponibilité et réactivité des interactions utilisateurs

Une évidence s'imposait à moi : le langage C++. Je ne connaissais que peu l'environnement technique des postes de travail et j'avais juste la main sur le serveur. J'ai donc apprivoisé le serveur. Comment ? Avec les API Win32 :

- Initialisation du programme : base de registres
- Données de configuration : base de registres avec ACL
- Fichier de logs et traces : sur `c:\temp`
- Composants multithreads de communication : ipc TCP/IP
- Composants multithread de cache : mémoire partagée
- Composants multithread pour le site IIS : COM et OLE
- Interface graphique pour client léger : ASP et COM

La mise au point de composants techniques

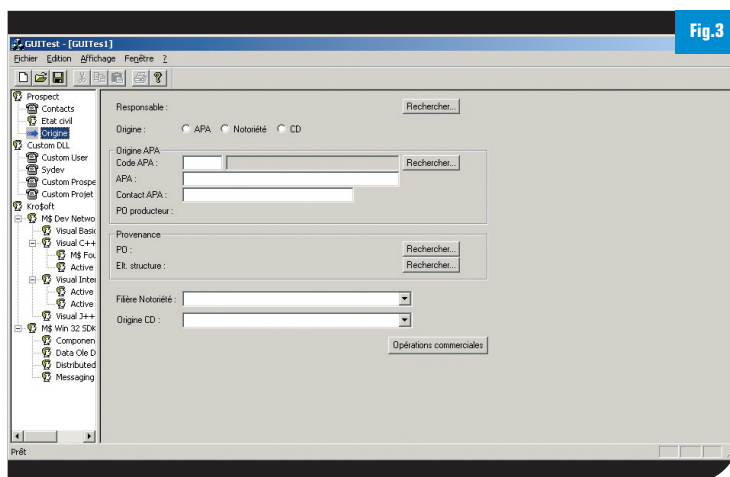


Fig.3

Application MFC en C++ de gestion des prospects (1998)

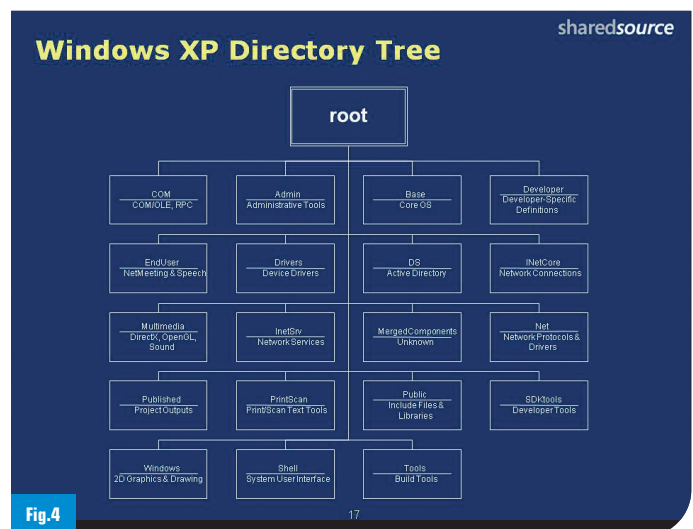


Fig.4

n'a pas été compliquée car les composants sont simples et ressemblent à n'importe quel sous module d'une application serveur de communication :

- Gestion de logs multithreads
- Outils de synchronisation (single writer multiple reader)
- Gestion des sockets
- Gestion des threads et des objets de contexte (paramètre générique)

La partie la plus délicate a été non pas le C++ mais la partie graphique en ASP, COM et HTML. La partie commune entre ASP et COM est basée sur la mémoire partagée entre le module de communication et les zones tampons, mais en accès protégés.

Un programme de communication est une boîte noire – en réalité c'est un service Windows qui est démarré en automatique (auto start). Le service web est géré par IIS. Les pages ASP font une interaction entre IIS et le service Windows via le biais des composants COM. Qu'est ce qu'un composant COM, c'est une classe C++ qui fait office de service RPC – le raccourci est fait mais le concept est là.

Retour vers le passé

Le langage C++ permet de développer une application avec tellement peu d'adhérence, si ce n'est le système d'exploitation, que la durée de vie de cette application peut être facilement de 2, 5 ou 10 ans. Ceci n'est pas rare. Il se peut que l'application passe par divers stades de maintenance ou différentes versions de compilateurs, la majorité des environnements de développement, sur les plateformes Windows sont Microsoft Visual Studio ou Intel.

Je prends le cas d'une base Visual Studio sur Windows. Une application développée en 1998 sous Visual C++ 6.0 peut tourner en 2010 sous Visual C++ 2010.

Il va falloir, éventuellement, car ce n'est pas une obligation, faire quelques améliorations car certaines API sont considérées *unsafe* et doivent être évitées (problème de buffer overrun) mais rien n'interdit de recompiler en l'état et ... ça marche.

C'est un peu comme si vous pouviez acheter chez un constructeur automobile un modèle sorti en 2000. Cela peut paraître non conforme ou étrange mais le code qui a été

réalisé en 2000 ne doit pas forcément être réécrit parce que de nouvelles technologies ou langages sortent sur le marché.

Le C++ est un langage massivement utilisé dans diverses industries dites sensibles et il n'est pas question de remettre en cause une approche commerciale globale parce qu'un architecte pense que cela fera mieux de programmer en .NET (ou Java).

Et si on migrerait le tableau de bord de l'A380 en .NET dirait Bob... Ça ne va pas être possible. La durée du produit est de 30 ans. Garantie sur facture.

La stabilité d'une architecture ou d'une entreprise est garantie par ses produits. Les fondamentaux d'une entreprise comme Microsoft sont Windows, Office et les produits serveur. Depuis 2000, Microsoft fait 50 milliards de chiffre d'affaires annuel. Le service ne représente que 2% de son CA. La gamme Windows pèse 15 milliards, Office pareil et l'ensemble des produits serveurs aussi. Le reste est représenté par la division des jeux vidéo et le Service.

Posons nous la question, à la place de Microsoft, s'il est possible de remettre en

Indice TIOBE

Position Feb 2011	Position Feb 2010	Delta in Position	Programming Language	Ratings Feb 2011	Delta Feb 2010	Status
1	1	=	Java	18.482%	+1.13%	A
2	2	=	C	14.986%	-1.62%	A
3	4	↑	C++	8.187%	-1.26%	A
4	7	↑↑↑	Python	7.038%	+2.72%	A
5	3	↓↓	PHP	6.973%	-3.03%	A
6	6	=	C#	6.809%	+1.79%	A
7	5	↓↓	(Visual) Basic	4.924%	-2.13%	A
8	12	↑↑↑↑	Objective-C	2.571%	+0.79%	A
9	10	↑	JavaScript	2.558%	-0.08%	A
10	8	↓↓	Perl	1.907%	-1.69%	A

Operating System Components

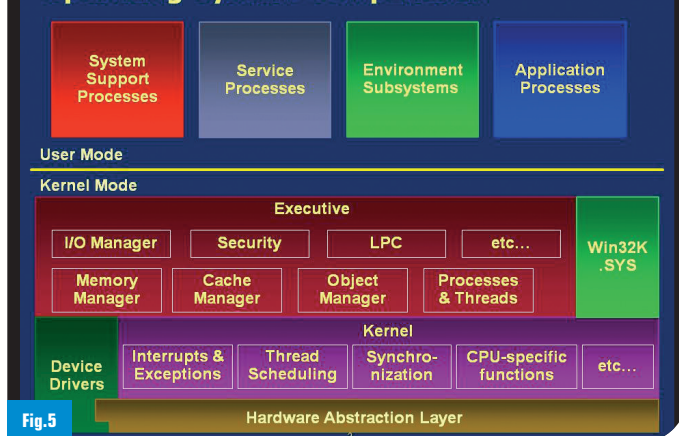


Fig.5

Process and Thread Structures

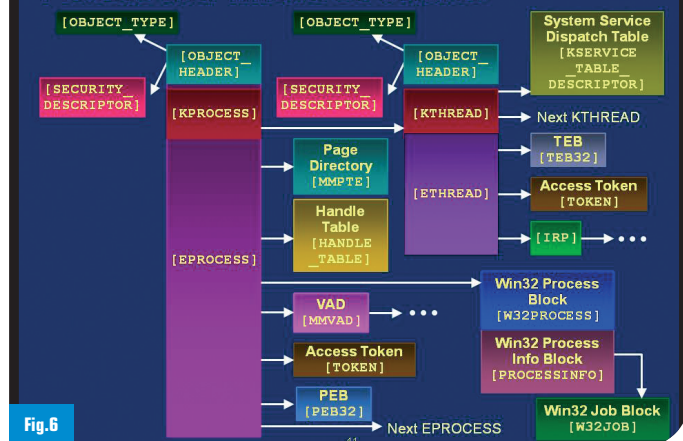


Fig.6

... Calling Windows/Posix APIs

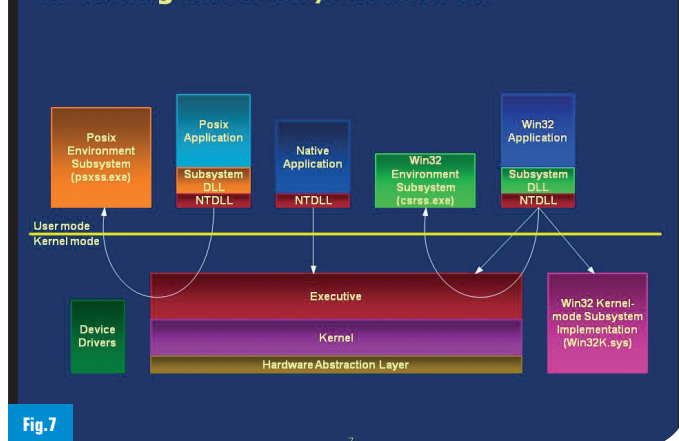


Fig.7

question un business model qui fait de Microsoft l'une des entreprises les plus compétitives au niveau mondial, en décidant de migrer la suite Office (Word, Excel, Outlook, PowerPoint, Visio) en .NET car le langage C# et le framework sont des outils stables et matures.

Et que le marché les a adoptés plutôt positivement.

Les composants Shell de Windows

Il est des fonctionnalités sous Windows dont on pourrait se demander pourquoi elles sont toujours faites en C++ ? L'explorateur de fichier, WordPad, Paint, etc. Posez-vous la question et essayons de répondre simplement.

Prenons WordPad et la liste des fonctionnalités qui vont poser un problème :

- Aperçu avant impression et impression du document
- Visualisation des objets liés OLE

Prenons Paint :

- Manipulation de larges zones mémoire ou buffer pour les images
- Gestion de l'impression

Prenons l'Explorateur Windows qui est res-

ponsable, entre autres, des éléments suivants :

- Le bureau Windows et la présentation des groupes de programmes
- La gestion des périphériques (USB, etc.)
- La gestion des opérations sur les fichiers

Windows

L'arborescence des sources de Windows permet de mieux comprendre l'adhérence des différents composants. Voici l'arbre de source Windows XP, à titre d'exemple. Tout est en C et C++ : [Fig.4].

La vue de cette arborescence nous ramène sur la théorie des systèmes d'exploitation et plus particulièrement l'architecture des systèmes d'exploitation Windows 32/64 bits [Fig.5]. Pour un développeur C++ sur un environnement Windows, une application est un processus qui contient un thread principal. Un processus contient des informations relatives à son environnement d'exécution et le programmeur C++ peut manipuler les API Win32 pour affiner, précisément le comportement de son application. La sécurité est couverte à travers la notion des processus et des threads. Pour plus d'information, consultez les ouvrages de Jef-

frey Richter sur la programmation système [Fig.6]. De plus, le langage C ayant permis de développer le système Unix, il existe sous Windows un sous-système qui permet de faire exécuter des programmes C POSIX. Un programme peut s'exécuter sur Unix et sous Windows juste en le recompilant. Étonnant non ? [Fig.7]. Sur un système d'exploitation, il y a des périphériques matériels et là, le langage C/C++ est le langage de l'innovation car il faut faire appel aux couches basses du système d'exploitation.

Conclusion

Le compromis sur la performance n'est pas possible. Une application est liée à son système d'exploitation car c'est son environnement d'exécution. Comme toute société, nous vivons dans le rêve, l'illusion et les mirages et le marketing... Les O.S. sont faits en C/C++. Les applications qui durent sont

faites en C++. Le monde appartient à C++.



■ Christophe Pichaud
Consultant sur les technologies Microsoft
MCSD, MCSD.NET



Concours de programmation parallèle inter-écoles ACCELER8

Venez affronter les développeurs d'autres écoles sur des problèmes écrits par des ingénieurs Intel.

40 netbooks à gagner.

plus d'informations : <http://software.intel.com/fr-fr/>





C++ est-il en train de redevenir LE langage ?

C'est une question que l'on est en droit de se poser en regardant le classement TIOBE des langages pour février 2011.

C++ remonte à la troisième place. Et si Java reste le numéro un, les agitations de sa communauté pourrait bien lui nuire à terme. Alors C++ est-il en train de redevenir le numéro un ?

Quand C++ est apparu sur la scène informatique, ce fut l'engouement. On l'employait partout. C'était LE langage informatique, surtout dans les domaines industriels. Parce qu'il apportait, à très peu de choses près, une performance voisine de C, avec l'expressivité de la programmation objet et ses facilités pour la réutilisation du code, C++ fut considéré comme le langage avec lequel on pouvait tout faire. Et C++ a continué de monter en puissance, notamment avec l'arrivée de la programmation générique et de ses fameux templates. Mais il s'est aussi vite taillé une réputation de langage difficile.

Paranoïa ou insouciance ?

C'est que le programmeur doit toujours faire extrêmement attention à ce qu'il écrit avec C++. En outre, le programmeur doit gérer moult détails souvent fastidieux, tels que l'allocation et la libération de ressources, et il faut bien le dire, C++ venait avec une bibliothèque standard un peu pauvre. Alors quand le "facile" Java (ou autre langage similaire) est apparu, avec son ramasse-miettes et ses riches bibliothèques, on a vite prédit la mort de C++. Car on pouvait programmer facilement, sans trop se soucier de la mémoire, des ressources. Puis le temps passant, on s'est aperçu qu'un Java pouvait s'avérer très lourd, et aussi qu'il n'était pas toujours aussi simple que cela. On s'est aperçu qu'un ramasse-miettes, ça ne le faisait pas toujours, comme par exemple dans le cas classique de connexions à une base de données libérées moins vite par le ramasse-miettes qu'elles ne sont allouées par le code client. Alors finalement on se dit que plutôt que de devoir faire systématiquement attention à invoquer une méthode close, se

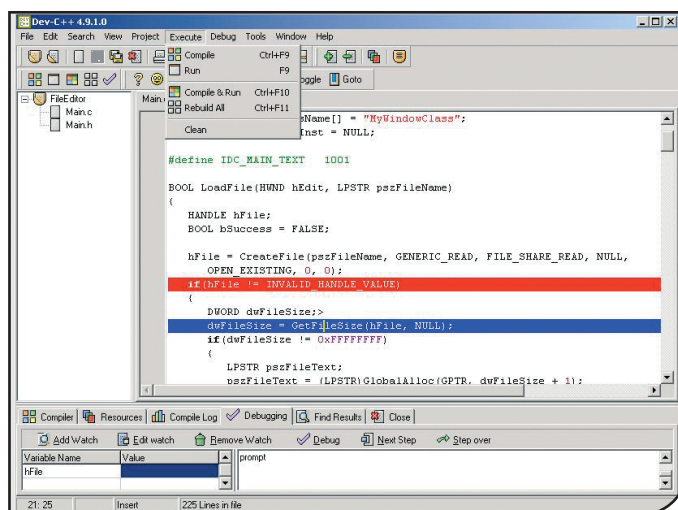
reposer sur le caractère déterministe des destructeurs de C++ et sur la technique typique C++ "d'acquisition de ressources par initialisation", c'était bien aussi. Et voilà que l'insouciance change (presque) de côté. :) Et puis, malgré les augmentations fantastiques des possibilités du matériel, on s'est aperçu que la mémoire restait toujours une denrée rare, et que les performances demeuraient toujours un problème.

In medio stat virtus.

Et finalement ce vieux C++, malgré sa réputation de complexité est toujours là, et on lui trouve un regain d'intérêt. Complexe ? Ça se discute. Après tout, ses spécifications sont beaucoup moins volumineuses que celles de Java. Mais difficile à enseigner certainement, et cela est sans doute une des raisons de sa baisse de popularité. Mais on a

5 ou 10 ans, selon la loi de Moore, mais l'optimisation va redevenir très, très sexy, quand on s'apercevra combien l'abstraction nous coûte cher." Pour continuer à monter en puissance, les processeurs ont dû devenir multi-cœurs. Or, C++ avec son futur standard C++1x (qui tarde un peu à venir il est vrai) va bientôt être doté d'une bibliothèque standard pour la programmation parallèle, et il devrait revenir s'imposer dans ce domaine, malgré l'existence d'un ADA sans doute injustement délaissé, ou de frameworks (Async, Axul, CCR, etc.), tous d'une grande qualité pour ce qui est de l'écriture de code, mais aux performances à l'exécution moins bonnes que le code natif, et volumineux par les runtimes sur lesquels ils s'appuient. C++ devrait aussi faire un retour en force dans l'embarqué, même si ce n'est pas encore vraiment la tendance. Car dans l'embarqué,

plus un runtime est petit et mieux c'est, plus les applications sont performantes et mieux c'est, à la fois pour l'utilisateur et pour la durée de vie de la batterie. Et puis il y a les implémentations de langages. Ainsi la machine virtuelle de Java est écrite en C++ et ça ne va pas changer de sitôt! L'essentiel des langages sont implémentés en C++ (ou C). Java a bien prouvé sa capacité de langage d'implémentation (Groovy, Clojure, Jython, etc.), mais les problèmes de performances demeurent. Aujourd'hui C++ n'est



toujours besoin de lui. Herb Sutter, gourou C++ chez Microsoft, l'avait ainsi exprimé à la PDC de 2009: "Le besoin d'optimisation contraindra à revenir à de la programmation de bas niveau, la piste de l'augmentation de fréquence des processeurs s'épuisant sérieusement. On tiendra encore peut-être

plus le langage à tout faire qu'il a été, et c'est tant mieux. Mais il restera un des tous premiers langages, parmi les indispensables, parce qu'il est mieux que C, et parce que l'informatique, finit toujours par rechercher la performance.

■ Frédéric Mazué

Les outils des Décideurs Informatiques

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

Cas clients
Actu triée par secteur | *Avis d'Experts*



Actus / *Evénements* | *Newsletter* | *Vidéos*



www.solutions-logiciels.com

☐ **OUI, je m'abonne** (écrire en lettres capitales)

Envoyer par la poste à : Solutions Logiciels, service Diffusion, 22 rue René Boulanger, 75472 PARIS - ou par fax : 01 55 56 70 20

1 an : 50€ au lieu de 60€, prix au numéro (Tarif France métropolitaine) - Autres destinations : CEE et Suisse : 60€ - Algérie, Maroc, Tunisie : 65€ - Canada : 80€ - Dom : 75€ - Tom : 100€
10 numéros par an.

☐ M. ☐ Mme ☐ Mlle Société

Titre : Fonction : ☐ Directeur informatique ☐ Responsable informatique ☐ Chef de projet ☐ Admin ☐ Autre

NOM Prénom

N° rue

Complément

Code postal : Ville

Adresse mail

☐ Je joins mon règlement par chèque à l'ordre de SOLUTIONS LOGICIELS ☐ Je souhaite régler à réception de facture

Le multiplateforme : nouvelle génération ou illusion ?



“Ecrire le code une fois, l’exécuter partout”, voilà un des slogans de Java. Le multiplateforme a du bon et du moins bon. Cherche-t-on toujours un code unique pour toutes les plateformes, au risque d’aplanir les spécifications et de perdre en richesse native ou au contraire, faut-il adapter pour chaque plate-forme ?

Cette dualité est plus réelle que jamais, notamment sur les plateformes mobiles de type smartphone. Adobe mise sur une approche unique avec Flash, alors que Microsoft, Apple, misent au contraire sur une adaptation, un développement natif. Oui, on perd en portabilité mais on gagne en fonctionnalité, en performances.

Quand on développe pour un système, le développeur optera pour le langage et l’outil natif. Ce sera souvent un

réflexe. Ensuite, tout va dépendre du projet, de sa nature. Si l’application n’a pas vocation à être portée, pourquoi se priver du modèle de développement de la plate-forme cible ?

Et même si le projet doit être porté sur plusieurs cibles, le choix du langage ne va pas de soi. Java est loin d’être le seul candidat car le C++ redevient une valeur sûre, de même que des technologies comme Mono.

Vous l’aurez compris, le multiplateforme est un sujet de plus en plus complexe et où les solutions supposées « universelles » ne le sont pas forcément. Dans ce dossier, nous allons tenter d’en savoir plus et de définir une grille de lecture du multiplateforme.

■ François Tonic



Pourquoi et comment le multiplateforme ?

Qu'entend-t-on par programmation multiplateforme ? Il s'agit de déployer une application ayant un code identique sur plusieurs systèmes cibles (targets), par exemple : sur Windows, Linux ou sur Android et Windows Phone 7.

La première chose est de savoir si vous avez besoin de porter votre code, votre application sur d'autres systèmes. Si vous vous limitez à un système, une plate-forme, ce dossier n'est pas fait pour vous. Malgré tout, même au sein d'une même plate-forme, vous aurez des précautions à prendre, des adaptations : passer d'une distribution Linux à une autre n'est pas forcément simple, le modèle iPhone et iPad diffère sur certains points, en Android, les différentes versions dans la nature compliquent le développement et le niveau fonctionnel à adopter.

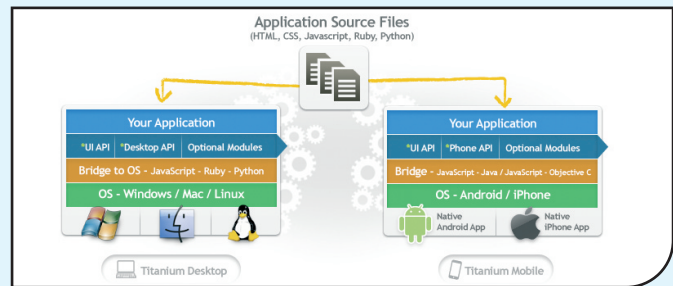
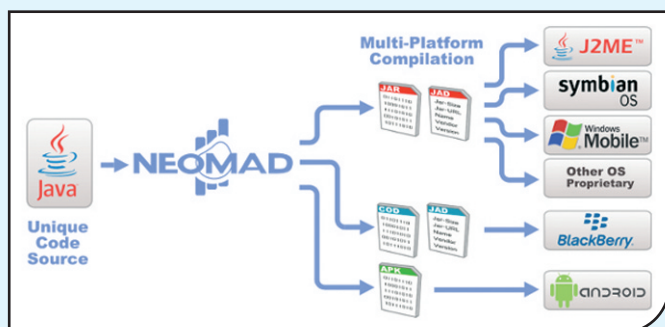
Les contraintes du multiplateforme sont nombreuses :

- le langage doit fonctionner de manière prédictible et de la même manière sur chaque target
- l'interface doit elle aussi rester identique
- si recompilation, le code ne doit pas changer
- prise en compte des modes de déploiement sur chaque target
- il ne faut pas que des limitations API / fonctionnelles sur des systèmes cibles gênent le développement
- vérification à la volée des erreurs de targets éventuelles

La complexité, l'hétérogénéité technologique gênent le multiplateforme, il faut donc savoir le gérer convenablement. D'autre part, le développement web sera par définition multiplateforme même si cela n'empêche pas de tester sur chaque système et navigateur car le support de telle ou telle technologie peut être inexistant ou limité. Ainsi la 3D ne reposera pas forcément sur la même technologie (OpenGL, WebGL, DirectX ?). Pareillement sur l'interface, comment assurer une bonne portabilité de l'interface ?

Une sélection de langage « desktop » :

Java	Théoriquement comportement identique partout. En réalité, il faut se méfier de la qualité des JVM et de leur niveau sur chaque target. A surveiller de très près. Et l'interface peut subir des aléas selon le système.
C#, VB.Net	Pas de support en dehors de Windows, sauf à passer par Mono pour le C#
Mono	Implémentation open source de C#. Disponible sur Linux, Windows, MacOS X.
C++	Langage présent partout et capable d'être très facilement repris sur différents systèmes. Programmation parfois difficile.
Objective C	MacOS X et iOS



Le cas particulier des smartphones

Le développeur aura l'abondance du choix pour du pur natif et du multiplateforme. Pour le multi-terminal / systèmes mobiles, les solutions se multiplient : Service2Media, NeoMades, Mobile Distillery, Mottwin. Ce dernier est constitué de deux parties, sans oublier la nécessité d'avoir un runtime installé sur le smartphone :

- MOS Runtime assurant la réplication intelligente des données, ainsi que tous les aspects administration et supervision
- MOS Application Builder : SDK pour le développement d'application utilisant la technologie Mottwin.

Plusieurs de ces solutions reposent sur l'usage de Java (et notamment de Java ME) ce qui n'est forcément pas une bonne chose pour les performances.

Comme nous le verrons dans la suite de ce dossier, le C++ reste une valeur sûre et efficace du multiplateforme, avec les performances qu'offre le langage. D'autre part, pour l'interface, on pourra utiliser des frameworks spécifiques de type Qt. Et surtout, le C++ étant une norme, les fondations sont stables et n'évoluent pas constamment, garantie d'un bon fonctionnement même après 5 ans ! Ce qui ne sera pas le cas avec un code C# ou Java.

Le pour et le contre

Pour	Contre
On paie pour un environnement de développement	Niveau fonctionnel moindre
On code une fois ou presque	Coût additionnel (pour chaque plateforme)
Modèle de développement unique	Quid des nouvelles fonctions ?

Utilisez le plus possible des bibliothèques, API, frameworks disponibles sur l'ensemble des targets voulus et ayant le même niveau de support partout. Si vous avez besoin de recompiler le code sur chaque plate-forme, utilisez la même version de compilateur. Ces bibliothèques assurent un niveau d'abstraction suffisant pour rendre votre code le plus indépendant possible, par contre en cas de mapping avec le système, vérifiez que tout se déroule correctement. De nombreux composants tiers risquent de ne pas être disponibles partout.

■ François Tonic

Restlet et le multiplateforme

Tout le monde connaît bien le slogan de Java: « write once, run anywhere ». Écrire une fois le code et l'exécuter sur n'importe quel système ou plateforme supportant Java en se basant sur une machine virtuelle. Cet aspect est un de ceux qui a fait la popularité de Java en permettant à une même application de s'exécuter sur plusieurs systèmes d'exploitation.

Différents types d'environnement fournissent désormais un support de la technologie. Les applications standard (Java SE) et d'entreprise (Java EE) restent toujours d'actualité mais elles peuvent désormais être exécutées dans des contextes plus larges allant de la plate-forme cloud au terminal mobile ou même au navigateur Web. Dans cet article, nous allons décrire concrètement les possibilités et difficultés de l'utilisation d'applications Java dans ces différents contextes. Nous verrons que tout n'est pas aussi simple et que des adaptations sont la plupart du temps nécessaires.

Nous nous basons dans cet article sur Restlet Framework afin d'illustrer l'utilisation d'un même code source dans différents environnements. Cet outil propose un socle léger et très flexible afin d'implémenter et d'accéder aux applications RESTful et par là même de tirer parti de toute la puissance du Web. Les fondations de cet outil ont été décrites dans le n° 137 du magazine.

Éditions et extensions Restlet

Restlet est basé sur un seul code source Java visant à fournir une boîte à outils dans l'utilisation de REST et de HTTP. L'outil a opté pour une organisation modulaire par le biais de la structuration suivante :

- Un noyau léger fournissant le cœur du framework, composé d'une API Java et d'un moteur implémentant cette dernière;
- Un ensemble d'extensions permettant d'utiliser des standards ou des technologies particulières dans ce contexte.

Cette approche permet de ne proposer que les mécanismes souhaités et utiles pour l'application. Elle permet également par là même, d'adresser les différentes limitations de certains environnements.

En parallèle des extensions, Restlet fournit le mécanisme d'édition. Il correspond au fait de fournir des artéfacts cohérents et validés pour l'exécution dans un environnement particulier en prenant en compte ses éventuelles limitations et en ne sélectionnant que les extensions utilisables. Pour ce faire, les méthodes ou

blocs de code non supportés sont automatiquement supprimés par l'outil de build en se basant sur différentes méta données spécifiées à ce niveau.

Le code suivant illustre comment un commentaire spécial permet la suppression du constructeur de la classe ClientResource dans le cas de l'édition GWT de Restlet. En effet, la classe URI n'est pas supportée dans ce contexte. Cette suppression se base sur l'instruction *ifnndef* :

```
// [ifndef gwt] method
/**
 * Constructor.
 *
 * @param context The context.
 * @param uri The target URI.
 */
public ClientResource(Context context, java.net.URI uri) {
    this(context, Method.GET, uri);
}
```

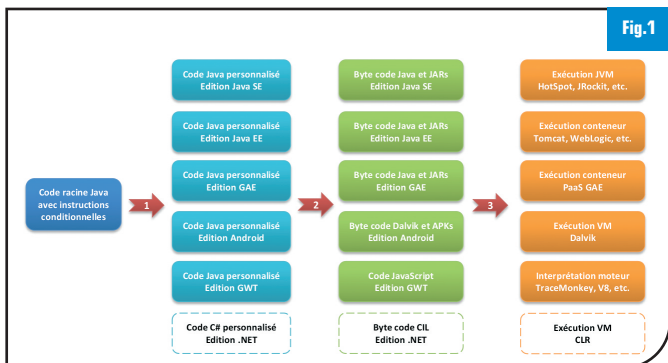
Cette approche nécessite une revue de tout le code source de l'outil, ce qui peut s'avérer complexe. A cet effet, un outil complémentaire a été mis en œuvre, ce dernier étant capable de vérifier automatiquement que toutes les classes et méthodes utilisées dans une édition sont bien supportées par l'environnement cible d'exécution. En suivant ce mécanisme, Restlet fournit les différentes éditions suivantes :

- **Édition Java SE** : édition dédiée à la version Standard Edition de Java. Fournit toutes les briques pour utiliser Restlet dans des applications Java déployables sur une simple JVM, sans aucune dépendance.
- **Édition Java EE** : enrichit l'édition précédente avec les briques relatives à la version Enterprise Edition de Java. Offre des facilités pour utiliser Restlet dans des applications Java d'entreprise déployables dans un conteneur de Servlet ou un serveur d'applications complet.
- **Édition Android** : édition fournissant un portage de Restlet assez exhaustif pour l'environnement Android permettant un déploiement sur smartphone ou tablette.
- **Édition Google App Engine** : édition fournissant un portage de Restlet pour la plate-forme de cloud computing de Google.
- **Édition Google Web Toolkit** : édition permettant l'utilisation de la partie cliente de Restlet dans des applications GWT et donc dans un navigateur Web sans aucun plugin, en pur JavaScript.

La figure détaille les trois étapes clés de transformation du code source racine Java et instructions exécutables ou interprétables [Fig.1]. Vous pouvez remarquer dans cette figure comment une version C# de Restlet est envisagée en suivant une approche similaire avec en plus une conversion de syntaxe.

Regardons maintenant les spécificités de ces différentes éditions afin d'utiliser Restlet respectivement dans le Cloud, dans des terminaux mobiles et dans des navigateurs.

Pour chaque édition, nous fournirons une liste non exhaustive des





extensions supportées. La liste complète est accessible à l'adresse : http://wiki.restlet.org/docs_2.0/313-restlet.html.

Utiliser Restlet dans le Cloud

Tout d'abord, la plateforme Google App Engine permet d'implémenter et d'héberger des applications distribuées sur des serveurs Google. Elle fournit en natif des mécanismes de répartition de charge et de persistance dans une base de données de type BigTable. Cette plateforme est particulièrement adaptée pour créer et héberger des applications à la demande (SaaS).

Google App Engine supporte la technologie Java mais avec certaines restrictions au niveau des API utilisables. Pour ce faire, Restlet met à disposition une édition dédiée utilisant uniquement les API supportées. Dans ce contexte, les extensions suivantes sont, entre autres, supportées :

- *org.restlet.ext.freemarker* : support de FreeMarker;
- *org.restlet.ext.jackson* : support de l'outil Jackson;
- *org.restlet.ext.servlet* : support de l'API Servlet.
- *org.restlet.ext.spring* : support du framework Spring;
- *org.restlet.ext.velocity* : support de Velocity;

Puisqu'une application Google App Engine correspond globalement à une application Servlet, cette édition est similaire à l'édition Java EE. La mise en œuvre d'une ressource au niveau du serveur nécessite ainsi l'utilisation de la classe `ServerServlet` pour déployer l'application Restlet. Cette classe se configure dans le fichier `web.xml`, comme décrit ci-dessous :

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee" version="2.5">
  <display-name>Contact Application</display-name>

  <servlet>
    <servlet-name>RestletServlet</servlet-name>
    <servlet-class>
      org.restlet.ext.servlet.ServerServlet
    </servlet-class>
    <init-param>
      <param-name>org.restlet.application</param-name>
      <param-value>
        com.programmez.restlet.ContactApplication
      </param-value>
    </init-param>
  </servlet>

  <servlet-mapping>
    <servlet-name>RestletServlet</servlet-name>
    <url-pattern>/*</url-pattern>
  </servlet-mapping>
</web-app>
```

Il est à noter que Restlet est également utilisable dans les environnements Amazon EC2 et Azure, offrant une réelle portabilité entre les trois grandes plates-formes de cloud [Fig.2].

Restlet pour les terminaux mobiles

Actuellement le langage Java est revenu sur les clients et notamment sur les terminaux mobiles. La plateforme Android est particulièrement populaire de par sa disponibilité sur des téléphones

aux capacités similaires à l'iPhone d'Apple et sa gratuité et disponibilité en open source. Elle correspond à une pile complète d'outils allant du système d'exploitation et d'une machine virtuelle spéciale dénommée Dalvik jusqu'à différentes applications ciblant ce type de terminaux. La force d'Android réside dans sa capacité à proposer des applications extrêmement riches et innovantes telles que la synchronisation de ses données entre plusieurs postes.

Android fournit un sous-ensemble d'API Java utilisables et restreint donc les extensions Restlet utilisables dans ce contexte. Seules les extensions suivantes sont supportées :

- *org.restlet.ext.atom* : support de la syndication Atom;
- *org.restlet.ext.crypto* : support de la cryptographie;
- *org.restlet.ext.jaas* : support de la sécurité JAAS;
- *org.restlet.ext.json* : support des représentations JSON;
- *org.restlet.ext.odata* : support du protocole OData;
- *org.restlet.ext.rdf* Support de RDF.
- *org.restlet.ext.xml* : support des représentations XML et XSLT;
- *org.restlet.ext.net* : support de la classe `URLConnection`;
- *org.restlet.ext.httpclient* : support de l'outil `HttpClient`;
- *org.restlet.ext.jackson* : support de Jackson pour JSON.

De plus, certaines aptitudes de Restlet à découvrir automatiquement les convertisseurs et connecteurs présents ne fonctionnent pas. Cela est dû au fait qu'Android ne prend pas en compte les fichiers de description présents dans le répertoire `META-INF/` services des fichiers JAR lors de la construction des paquets APK.

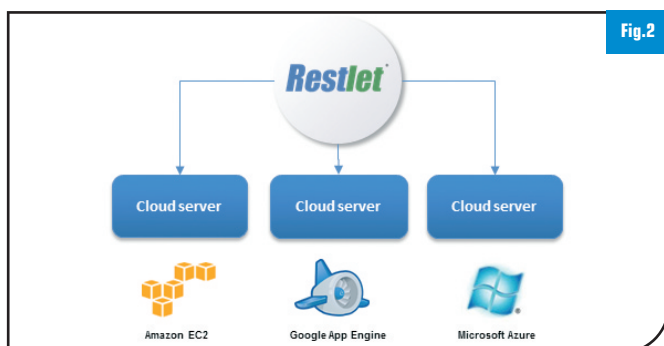
De ce fait, il est nécessaire d'enregistrer à la main ces connecteurs et convertisseurs Restlet en utilisant directement l'instance du moteur, comme décrit dans le code ci-dessous :

```
Engine.getInstance().getRegisteredConverters().add(
    new JacksonConverter());
Engine.getInstance().getRegisteredClients().clear();
Engine.getInstance().getRegisteredClients().add(
    new HttpClientHelper(new Client(Protocol.HTTP)));
```

Indépendamment des API supportées, il est à noter qu'il est nécessaire de spécifier une permission afin de pouvoir exécuter des requêtes distantes. Cet aspect se réalise dans le fichier `AndroidManifest.xml` permettant de configurer de l'application, comme décrit ci-dessous :

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="org.restlet.android"
  android:versionCode="1"
  android:versionName="1.0">
  <application android:icon="@drawable/icon" android:label="@string/app_name">
    <activity android:name=".ContactActivity"
      android:label="@string/app_name">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
  <uses-permission android:name="android.permission.INTERNET" />
</manifest>
```

Fig.2



Restlet dans les navigateurs Web

Actuellement, le langage Java intègre d'une nouvelle manière les navigateurs Web, donc côté client, avec la technologie GWT. Il s'agit d'une approche différente de celle des applets dont on se servait couramment il y a quelques années. En fait, Java est utilisé comme langage pour le développement des applications, la partie client étant ensuite compilée en JavaScript.

Bien entendu dans ce cas, l'ensemble des API Java utilisées est restreint puisqu'elles doivent avoir une correspondance possible en JavaScript et que les navigateurs imposent une stratégie de sécurité très stricte (Single Origin Policy, mono-threading, etc.). Le

périmètre du support est également moindre puisque seule la partie cliente de Restlet est utilisable. Faire fonctionner son support serveur dans un navigateur est un non-sens.

De ce fait, seules les extensions suivantes sont utilisables :

- *org.restlet.GWT* : module correspondant au portage de la partie cliente de l'API Restlet.
- *org.restlet.JSON* : module correspondant au portage du support des représentations JSON.
- *org.restlet.XML* : module correspondant au portage du support des représentations XML.

Pour utiliser Restlet dans une application GWT, il convient de spécifier les dépendances dans le fichier de configuration comme décrit ci-dessous :

```
<module>
  <inherits name='com.google.gwt.user.User' />
  <inherits name='org.restlet.GWT' />
  <inherits name='org.restlet.JSON' />
  <entry-point class='org.restlet.gwt.client.SimpleContact' />
</module>
```

Une fois la spécification des modules choisis réalisée, il est possible d'utiliser Restlet d'une manière similaire à celle des applications Java classiques. Certaines méthodes ne sont cependant plus disponibles de par des API non supportées et, comme les

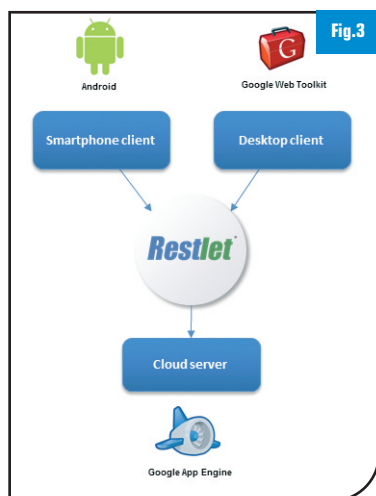


Fig.3

technologies AJAX sont employées, il est nécessaire d'utiliser Restlet de manière asynchrone et de bien configurer le support des requêtes cross domain pour vos applications [Fig.3].

L'édition GWT offre également des facilités spécifiques afin de plugguer un appel de manière transparente derrière une interface. Cet aspect tire parti de la fonctionnalité dénommée Deferred Binding de GWT permettant de

générer du code à la volée lors de la compilation de Java en JavaScript. Ainsi l'appel d'une ressource REST se réalise simplement de la manière suivante :

```
ContactResourceProxy contactResource
= GWT.create(ContactResourceProxy.class);

contactResource.getClientResource().setReference(
    "/contacts/123");

contactResource.retrieve(new Result<Contact>() {
    public void onFailure(Throwable caught) { /* Erreur */ }
    public void onSuccess(Contact contact) { /* Succès */ }
});
```

Il est à noter que l'édition GWT utilise le mécanisme asynchrone de Restlet pour récupérer la réponse d'une requête. Ce mécanisme est également disponible dans les éditions classiques, la version GWT utilisant un portage de cette fonctionnalité.

Perspectives vers d'autres langages

Bien que le code racine de Restlet soit écrit en Java, des portages automatisés vers JavaScript, Objective-C et C# sont étudiés en se basant sur des mécanismes de conversion de langage. Cette approche vise à favoriser la maintenance et l'évolutivité de ces édi-

tions tout en permettant une personnalisation aussi poussée que nécessaire pour chaque cible. L'approche utilisée se base sur l'outil JDT (Java Development Tools), le parseur Java d'Eclipse, afin de charger en mémoire la structure des classes et méthodes. Le code dans le langage cible est ensuite généré à l'aide de templates récursifs. Il existe néanmoins des difficultés lors de ces types de conversion puisque les langages ne sont pas toujours identiques au niveau de la mise en œuvre des concepts objet. C'est le cas par exemple pour JavaScript qui se base sur les noms de méthodes et non sur les signatures pour les distinguer. De plus ce langage ne supporte pas tous les mêmes types et structures de contrôles. Des adaptations sont alors nécessaires.

En résumé

Dans cet article, nous avons décrit les problématiques de mise en œuvre d'outils basés sur le langage Java dans différents environnements d'exécution. Nous avons pris l'exemple de Restlet Framework, un outil proposant un socle léger et très flexible afin d'implémenter et d'accéder aux applications RESTful.

Cet outil est écrit en Java et met à disposition le concept d'éditeurs offrant différentes adaptations du code de l'outil afin de permettre son utilisation dans des contextes allant des applications Java simples jusqu'aux terminaux mobiles et navigateurs Web. Il fournit également une approche modulaire par l'intermédiaire d'extensions. En fonction de l'environnement cible et ses restrictions correspondantes, les extensions utilisables varient. Ces deux concepts permettent de garantir une utilisation correcte de l'outil pour un environnement supportant Java.

Cet article et l'étude du cas de Restlet Framework ont permis de mettre en lumière que des adaptations sont généralement nécessaires afin de garantir un bon fonctionnement du code Java dans différents environnements du fait de leurs éventuelles restrictions. Ces environnements peuvent correspondre à des plateformes serveurs comme des terminaux mobiles ou des navigateurs.

L'intérêt de Java comme langage d'écriture du code racine de l'outil réside d'abord dans la possibilité de le compiler dans du bytecode pour des JVM classiques ou pour Dalvik, ainsi qu'en JavaScript via GWT. De plus sa JVM offre un runtime non seulement portable mais très performant et pouvant être utilisés par d'autres langages comme Groovy, Python (projet JPython), PHP (projet Quercus), Scala, Closure, etc. Pour finir, différentes plateformes basées sur Java telles que Java SE, Java EE, Java ME, GAE et Android permettent de s'adapter à des usages particuliers. Java se positionne ainsi comme une solution multi-plateforme d'avenir.

Note : les documentations pour les éditions décrites dans cet article sont disponibles aux adresses suivantes :

- GAE : http://wiki.restlet.org/docs_2.0/252-restlet.html;
- GWT : http://wiki.restlet.org/docs_2.0/144-restlet.html;
- Android : http://wiki.restlet.org/docs_2.0/266-restlet.html.

■ **Thierry Templier** (templth@yahoo.fr)

Architecte Java, Java EE et Web, Co-auteur des ouvrages *Spring par la pratique* et *JavaScript pour le Web 2.0* (Eyrolles).

■ **Jérôme Louvel** (jerome.louvel@noelios.com)

Leader Technique – Restlet Framework, Co-fondateur – Noelios Technologies, Co-auteur de l'ouvrage *Restlet in action* (Manning) et contributeur à l'ouvrage *RESTful Web Services* (O'Reilly)

■ **Thierry Boileau** (thierry.boileau@noelios.com)

Développeur principal – Restlet Framework, Co-fondateur – Noelios Technologies, Co-auteur de l'ouvrage *Restlet in action* (Manning)



ASP.NET multiplateforme : Mono prêt pour la production !

Le multiplateforme existe aussi sur les technologies .NET. S'il est un domaine dans lequel une interopérabilité totale est envisageable, c'est bien dans les serveurs web : pas de dépendance aux technologies d'affichage, utilisation des protocoles réseau qui sont parmi les plus normés, etc. Cet article présente un retour d'expérience lors d'une mise en production multiplateforme d'une application de gestion écrite en .NET. Le déploiement le plus courant est sur une plateforme Windows, avec la pile .NET de Microsoft, mais de plus en plus, la solution est installée sur des serveurs Linux utilisant la solution open source Mono.

Mono était présent aux TechDays 2011, signe d'une maturité bien réelle, même si elle n'est pas encore très reconnue. La session de cette année, présentée par Jean-Baptiste Evain de Novell était réellement enthousiasmante, surtout pour les développeurs d'applications mobiles. Pourtant, lors du traditionnel sondage, peu parmi les personnes présentes utilisaient Mono, et encore moins en production. Notre expérience nous a convaincu que ce n'est qu'une question de temps, les arguments de Mono étant très persuasifs. La maturité est au rendez-vous, y compris pour des déploiements en mode hébergé.

Définition du besoin

Petit retour dans le temps... Nous sommes en 2008 et faisons face à de nombreuses sollicitations de nos clients remontées par nos commerciaux : « *qu'en est-il de la compatibilité de nos applications sur les systèmes d'exploitation Linux ?* » Devons-nous redévelopper nos applications avec un nouveau langage ou garder notre puissance de développement en C# avec les technologies .NET et continuer d'offrir des produits performants, mais mono-plateforme ?

A ce moment, le projet Mono commence à faire parler de lui. A cela s'ajoute notre besoin plus important d'hébergement avec la montée en puissance de nos solutions SaaS sur lesquelles nous souhaitons également bénéficier de plateformes plus attractives. Le défi devient encore plus intéressant en menant de front un passage de nos applications .NET sur Linux et cela pour nos 200 clients hébergés de l'époque. L'enjeu est de taille !

Méthode de mise en œuvre

La première démarche au lancement du projet a été de qualifier la faisabilité d'un portage sur Linux avec l'implémentation de Mono. Les premiers tests techniques sont encourageants et montrent qu'il est possible de faire tourner nos applications .NET sur des serveurs Linux. Le projet engagé, nous nous confrontons très vite aux problèmes de choix de plateforme. Pour être parfaitement opérationnel dans une installation sous Linux, le mode graphique se révèle vite un choix déterminant, nous amenant petit à petit à l'abandon des recompilations de binaires, en particulier les librairies Gluezilla qui sont déjà compilées sur SuSE Enterprise 64 bits. A l'inverse, l'outil Mono Tools for Visual Studio (de Novell égale-



ment) ne nous a pas convaincu. Nous avons décidé de le mettre en place pour faciliter la résolution de problèmes sur un serveur ASP.NET Mono, mais le débogage à distance posait trop de problèmes, et nous avons finalement continué avec la bonne vieille méthode des traces ad-hoc.

Les fournisseurs Oracle ont certainement été le point posant le plus de problèmes. Dans les premières versions de Mono, le support était même tout simplement inutilisable. En particulier, nous avons eu un problème avec une erreur OCI-Handler qui remontait de manière apparemment arbitraire, ainsi qu'une limitation très forte de la montée en charge. De l'aveu même de l'ancien mainteneur du provider Oracle de Mono, Novell a assigné une très faible priorité au projet, considérant que si des utilisateurs payaient les licences pour un SGBD Oracle, ils pouvaient tout à fait payer une licence pour un fournisseur .NET commercial. C'est ce que nous avons fait en acquérant la licence du provider ORA.NET de Devart. La situation s'est tout de même bien améliorée depuis. Non seulement, nous avons enfin réussi à trouver le problème récurrent qui venait d'une option Oracle de prise en compte des caractères accentués incompatible avec le fournisseur Mono, mais en plus

celui-ci a bien évolué, et nous sommes désormais revenus au fournisseur natif.

La qualification de PostgreSQL, sous sa forme EnterpriseDB, a également été une étape difficile. Au final, les liens avec la base de données auront été, et de très loin, le gros point de difficulté pour la qualification de Mono sur nos solutions, alors qu'elles ne font pourtant que des appels extrêmement simples aux fournisseurs. Cette première étape de qualification nous ayant convaincus du bien-fondé de l'approche Linux + Mono, nous sommes alors passés à une phase d'intégration.

Nous avons commencé par mettre à jour notre processus d'installation. L'application cible utilise pour ce besoin une application spécialement conçue par MGDIS, sous la forme d'un assistant générique à l'installation et l'exploitation. Ayant suivi depuis le début le difficile chemin du support de WinForms dans Mono, nous étions plus que circonspects, mais avons été, comme plusieurs fois, extrêmement surpris de la qualité de Mono. Le module d'installation, dont nous montrons une capture d'écran (correspondant au remplissage des settings) fonctionne sans aucune modification sous Mono 2.4.3 [Fig.1].

Outre l'installation qui doit être simple et rapide, les performances doivent aussi être au rendez-vous. Nous nous sommes alors lancés dans une seconde phase d'intégration, matérialisée par une série de tests de performance et de montée en charge. Une batterie de tests automatisés déjà réalisée sur des configurations Windows nous permet alors d'analyser le comportement en charge. Les résultats sont très intéressants puisqu'ils confirment globalement la bonne exécution sous Linux (nous verrons les quelques points de difficulté un peu plus loin), ainsi que l'obtention de temps d'exécution comparables à nos environnements de référence Windows [Fig.2]. Des relevés chronométrés mono-utilisateur sur une de nos applications de gestion et de pilotage intégré (PROGOS) nous montrent des résultats très satisfaisants avec une différence de seulement 3% sur les temps d'exécution de l'application entre une plateforme Linux / Mono / EnterpriseDB et un Windows 2003 / IIS 6.0 / Oracle 10.2.0.4.

Si les résultats unitaires sont très probants, la montée en charge reste également maîtrisée avec la version 2.6.7 de Mono. Sur Mono, la charge reste linéaire entre 1 et 75 utilisateurs simultanés avec seulement 3% de dégradation des temps de réponses (moyenne sur 249 cycles de tests), ce qui reste acceptable en sachant que nos expérimentations sont plus exigeantes que la réa-

lité d'usage. Nous soulignons toutefois que le point de rupture sera plus vite atteint sur Mono que sur une configuration Windows lors d'un stress applicatif plus conséquent.

	1 user	30 users	75 users	Dégradation à 75 users
Linux	355	357	368	3,1%
Windows	360	358	359	-0,28%

Temps moyen d'injection d'un cycle (en secondes)

Il est important de noter qu'en charge, la plateforme de référence sous Windows est en fait plus rapide qu'avec un seul utilisateur, ceci étant certainement dû à des effets de cache / pool. Ce n'est pas le cas avec l'environnement Mono.

Depuis l'adoption de Mono (version 1.9) jusqu'à la mise en production actuelle (version 2.6.7), nous avons consommé quelques 120 jours.homme. Actuellement, 440 clients utilisent notre service hébergé sous Linux (OpenSuSE 11.2, après la qualification initiale sur la SuSE Enterprise) 64 bits.

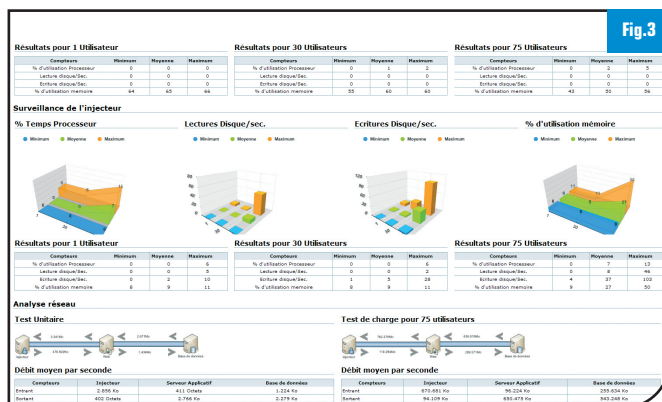
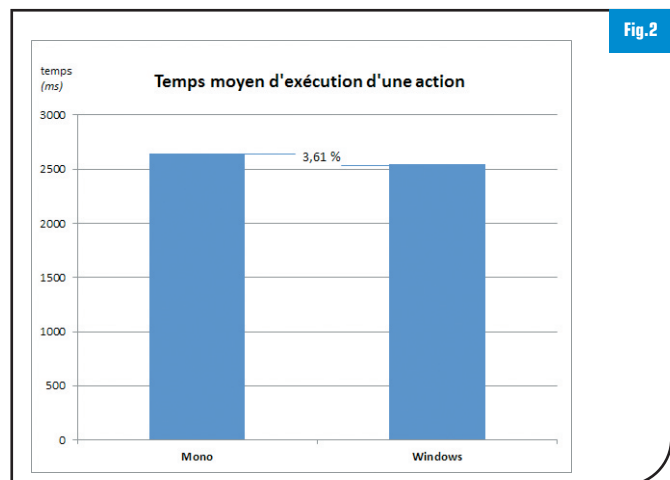
A noter au passage que la roadmap de Mono est depuis toujours très bien respectée. Nous ne sommes jamais restés en attente d'une évolution prévue et sur laquelle nous comptons, ce qui mérite d'être souligné, vu la complexité d'un tel projet.

Difficultés

L'utilisation d'une usine de tests fonctionnels nous a permis de valider très vite de nombreux scénarios métier sur la plateforme Mono, et de remonter quelques difficultés que nous avons dû prendre en compte [Fig.3].

Du point de vue de l'utilisation de Mono elle-même, les difficultés ont principalement concerné les providers de connexion aux bases de données, comme cela a été expliqué plus tôt. En particulier, la gestion de l'Unicode n'était pas parfaite dans la version 2.4.3, que ce soit sur le fournisseur Oracle ou sur le fournisseur pour PostgreSQL. Les champs de type LONG ont également été difficiles à manipuler, même si ceci est une constante sur les bases de données, et pas seulement dû à l'utilisation de Mono.

Les fournisseurs de données se sont révélés parfois très exigeants sur le code. Par exemple, le fournisseur Oracle interdit d'appeler une requête paramétrée avec des paramètres en plus de ceux contenus effectivement dans la requête, ce qui rend plus complexe l'écriture dynamique de requête. Les valeurs de paramètres devaient également être parfaites, car aucune troncature automatique n'était prévue. Nous avons bien sûr dans le code quelques endroits où des manipulations de chemins étaient réali-





sées en dur, avec mise en place d'un séparateur de répertoires directement dans la chaîne, parfois en utilisant `Path.DirectorySeparatorChar`, ce qui ne posait pas trop de problèmes, mais parfois directement avec le symbole « \ », qui ne fonctionne bien sûr pas en Linux, où le séparateur est le symbole « / ». La solution a simplement consisté à auditer le code sur ses utilisations de `System.IO`, et à modifier les endroits mal écrits pour qu'ils se basent bien sur `Path.Combine`.

```
class Program
{
    static void Main(string[] args)
    {
        // Ne devrait plus exister
        string CheminKO = args[0] + @"bin";

        // La bonne façon de faire, pour Mono mais aussi en général
        string CheminOK = System.IO.Path.Combine(args[0], «bin»);
    }
}
```

La gestion de l'emplacement du répertoire temporaire peut également poser problème si la lecture se fait sur la variable d'environnement au lieu d'utiliser la bonne API de .NET :

```
class Program
{
    static void Main(string[] args)
    {
        // Ne devrait plus exister
        string CheminTempKO = Environment.GetEnvironmentVariable(«TEMP»);

        // La bonne façon de faire, pour Mono mais aussi en général
        string CheminTempOK = System.IO.Path.GetTempPath();
    }
}
```

Jusqu'à la version 2.4.3, le chargement des assemblages était très contraignant, à savoir que toutes les dépendances devaient être présentes. Bien évidemment, ceci peut poser problème si on n'a pas un découpage absolument idéal des responsabilités des assemblages. Le problème a toutefois été corrigé sur les nouvelles versions. Une autre difficulté, pas en termes de fonctionnalités mais de performances (ce qui est rare pour Mono), est venue du parcours des relations sur un `DataSet`. Les fonctions permettant de remonter les entités pointées par une clé étrangère étaient extrêmement lentes, et il a donc fallu que nous surchargions les méthodes utilisées pour atteindre des performances satisfaisantes. En pratique, celles-ci étaient même tellement meilleures que nous les avons utilisées également en mode Microsoft.

```
// Pas de problème de lenteur tant qu'on boucle sur les tables
foreach (MonDataSet.MaitreRow LigneMaitre in DSSOFI.Maitre)
    foreach (MonDataSet.EsclaveRow LigneEsclave in DSSOFI.Esclave)
    {
        // ...
    }
```

```
// Par contre, le Select ci-dessous posait des problèmes de
performance
foreach (MonDataSet.MaitreRow LigneMaitre in DSSOFI.Maitre)
    foreach (MonDataSet.EsclaveRow LigneEsclave in DSSOFI.Esclave.
Select(«idmaitre='» + LigneMaitre.id + «'«, null, DataView
RowState.CurrentRows))
    {
        // ...
    }

// Ces problèmes apparaissent aussi lorsqu'on passe par les
fonctions Get[ObjetFils]
foreach (MonDataSet.MaitreRow LigneMaitre in DSSOFI.Maitre)
    foreach (MonDataSet.EsclaveRow LigneEsclave in LigneAxe.
GetEsclaveRows())
    {
        // ...
    }
```

Nous avons expliqué plus haut que WinForms, bien que cela semble incroyable, est très bien supporté par Mono. Nous avons toutefois rencontré un petit problème, qui venait de notre façon de concevoir notre assistant générique : le mode ligne de commande fait appel de manière transparente à des contrôles utilisateurs WinForms. En mode Microsoft, sous Windows, cela ne pose pas de problème, car il y a toujours une couche graphique à disposition. En mode ligne de commande Linux, nous nous sommes retrouvés à devoir installer un serveur X et rajouter les bibliothèques Gluezilla pour que notre programme fonctionne, même s'il n'affichait strictement rien. Ce dernier cas est assez révélateur : globalement, il faut avouer que ces difficultés ont surtout été de notre fait, et souligner que Mono fonctionne sans que nous ayons changé quoi que ce soit à notre processus de compilation, toujours basé à ce jour sur le SDK Microsoft. Il n'y a aucun code spécifique à Mono, et l'API est 100% compatible. Mono, cette fois, a agi en tant que révélateur de portions de code insuffisamment génériques. Le seul point sur lequel nous avons eu un manque est la signature cryptographique des fichiers manifestes de ClickOnce. Par contre, les classes de cryptographie standard fonctionnent bien, comme le montre le code ci-dessous :

```
// Ne pose pas de problème
SHA1 Engine = SHA1CryptoServiceProvider.Create();
Console.WriteLine(Convert.ToBase64String(Engine.ComputeHash
(byteArray)));

// Mono ne supporte pas les fichiers PFX en entrée
X509Certificate2 myCert2 = new X509Certificate2(@"MGDIS.pfx");
Console.WriteLine(myCert2.GetCertHashString());
```

Au final, il reste bien sûr quelques endroits dans notre code qui dépendent du mode Windows ou Linux, mais de manière extrêmement limitée.

```
public static bool ExecutionSousLinux()
{
    return Environment.OSVersion.Platform == PlatformID.Unix;
}
```

Pour les retrouver, nous avons réalisé une requête CQL dans l'excellent outil NDepend, en excluant notre module d'installation, car il est logique que celui-ci montre des spécificités d'exécution en fonction de l'OS, même si elles sont en pratique très limitées. Les résultats sont représentés [Fig.4], et on voit clairement que les utilisations sont très peu nombreuses : 13 sur les presque 5 millions d'instructions IL testées.

Il s'agit de quelques cas de comportements légèrement différents, et que nous allons détailler ci-dessous. Attention cependant : même si l'outil Mono Migration Assistant (votre meilleur allié pour ce type de campagne de qualification de Mono) vous dit que toutes les fonctions de votre assemblage sont prises en compte, ceci ne veut pas dire que les tests passeront parfaitement, même si c'est bien sûr un très bon signe. Un premier cas est celui des WriteXml sur les DataSet : Mono ne met pas le PreserveWhitespace par défaut. Autre chose sur le XML : Mono n'encode pas les contenus passés à un XmlTextWriter, alors que la pile .NET Microsoft le fait si nécessaire. La discussion n'est pas de savoir quelle méthode est la meilleure : elles sont différentes, et il faut le savoir, c'est tout. Pour les accents, la ligne suivante pose problème en Mono :

```
ChaineSansAccents = Encoding.ASCII.GetString(Encoding.GetEncoding(1251).GetBytes(ChaineEntree));
Nous avons donc utilisé plutôt un code comme ci-dessous :
ChaineSansAccents = Encoding.ASCII.GetString(Encoding.GetEncoding(1251).GetBytes(ChaineEntree));
string ChaineNormalisee = ChaineEntree.Normalize(NormalizationForm.FormD);
StringBuilder ChaineBuilder = new StringBuilder();
for (int Cpt = 0; Cpt < ChaineNormalisee.Length; Cpt++)
{
    char Caractere = ChaineNormalisee[Cpt];
    if (CharUnicodeInfo.GetUnicodeCategory(Caractere) != UnicodeCategory.NonSpacingMark)
        ChaineBuilder.Append(Caractere);
}
ChaineSansAccents = ChaineBuilder.ToString();
```

Un autre cas de différence est lorsque nous souhaitons retrouver le chemin physique des librairies en mode ASP.NET. Microsoft les place dans un sous-répertoire « bin », alors qu'en Mono, les assemblages sont directement dans le chemin du répertoire virtuel. En mode Microsoft, le StreamWriter va créer le répertoire si nécessaire, alors que Mono ne le fait pas. De même, String.Insert ne provoque pas d'erreur sur MS .NET avec une chaîne à insérer vide, alors qu'en Mono ce sera le cas :

```
string Chaine = «test»;
Chaine = Chaine.Insert(2, string.Empty);
```

Attention, tout ce qui est remonté ici concerne la version 2.4.3 de Mono, qui est longtemps restée notre version de référence. Il s'agissait en effet d'une version Long Term Support, et de la première sur laquelle nous avons été en mesure de passer en production.

Résultats

De manière générale, Mono est moins permissif que la pile Microsoft. Nous avons déjà évoqué ceci un peu plus haut : les difficultés rencontrées étaient le plus souvent liées à notre façon de coder, et Mono nous a finalement permis d'améliorer nos pratiques. Dans certains cas, ce rigorisme était peut-être excessif : par exemple, sur un fournisseur donné, la nécessité de passer les paramètres SQL dans le même ordre que celui de leur apparition dans la requête. Mais dans l'ensemble, il est remarquable que nous n'avons jamais eu à reconstruire un code au complet par manque d'une API en Mono par rapport à la version Microsoft de .NET. Et le cas de la modification de code pour des raisons de performance s'est restreint aux seuls parcours de clés étrangères sur les DataSet. Du point de vue des performances unitaires et en charge, la pratique a confirmé les résultats obtenus lors de notre phase de qualification et d'intégration. En plus de sa grande qualité fonctionnelle, Mono était également au rendez-vous pour les performances. Il faut souligner encore une fois que Mono ne souffre d'aucun manque de notre point de vue d'éditeurs de solutions de gestion, à part Code Access Security (désormais abandonné par Microsoft lui-même) ou le SSO Windows (ce qui est logique). Non content de cette complétude, Mono se permet même le luxe de proposer des services qui n'existent pas nativement dans la pile .NET : Cecil et IKVM en sont les exemples les plus connus, mais de nombreux outils comme Gendarme sont des ajouts d'égale importance.

Conclusion

Les bons résultats de la mise en œuvre de Mono sur plusieurs de nos plateformes nous ont permis d'envisager des projets encore plus ambitieux, et nous sommes aujourd'hui à l'aube de la diffusion d'une « box » contenant nos applications pré-déployées et prêtes à l'emploi. Cette « MgBox » repose sur un noyau Linux CentOS et embarque un Mono 2.6.7. La robustesse de Mono sera alors mise à l'épreuve pour assurer le bon fonctionnement de nos applications directement chez nos clients. Nos derniers tests montrent que cette implémentation répond parfaitement au besoin d'une trentaine de connexions simultanées sans dégradation significative des performances. Nous sommes également à ce jour en attente de validation du nouveau ramasse-miettes, prévu officiellement pour la version 3.0, et qui promet des améliorations conséquentes des performances. Le GC actuel est un peu limité technologiquement, et l'arrivée d'une collection à générations devrait permettre à Mono de rattraper le seul retard manifeste sur la pile ASP.NET native, à savoir la montée en charge.

■ Anthony Labarre, certifié ISTQB et ITIL, est responsable du pôle Tests, Intégration et Assistance.

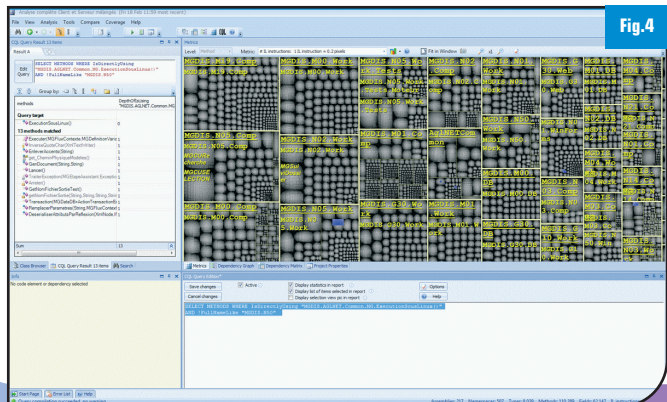
■ Jean-Philippe Gougoux, certifié MCPD.NET et MCTS, est responsable du pôle Architecture / Formation / Innovation.



MGDIS® Editeur de logiciels d'aide à la décision et de pilotage pour conduire agilement la politique de votre organisme.

<http://www.mgdis.fr> - <http://blogs.dotnet-france.com/jeanphilippeg/>

Fig.4





Adobe Flash Platform et les applications multiscreen

A l'origine, la technologie Flash a été créée par Macromedia pour concevoir et développer des applications multiscreen. En fait, la première version de Flash a été conçue dans les années 90 pour exécuter des interfaces sur tablettes. L'histoire veut que 20 ans plus tard, les tablettes deviennent un standard pour interagir avec les applications informatiques.

Le fait de dessiner des interfaces vectorielles permet de répondre facilement aux contraintes du développement multiscreen : différentes résolutions, différentes densités de pixels à l'affichage (dpi)... Mais ces défis techniques ne représentent que la face visible de l'iceberg.

Désormais, les applications internet riches devront s'exécuter sur des smartphones, des tablettes, des ordinateurs, des télévisions interactives, des bornes interactives... tous ces appareils disposent de différentes tailles d'écrans mais aussi de différentes capacités : accéléromètre, gps, multi-touch. Nous interagissons aussi différemment avec ces écrans : souris, clavier, doigts, télécommande, gestes dans l'espace. Il est possible avec la Flash Platform de développer une fois pour plusieurs écrans, mais ce n'est pas une solution miracle. C'est un raccourci mental et motivé par des enjeux d'industrialisation. La plupart du temps, les développeurs chercheront à optimiser leurs développements pour réutiliser un maximum de code qui ciblera un maximum d'écrans. Cet article présente plusieurs astuces pour développer des applications multiscreen, avec ou sans l'aide du framework Flex (qui se met à jour en version 4.5 pour supporter le multiscreen).

Principes de base pour le multiscreen

Il existe différentes approches pour développer une application multiscreen :

- 1 L'architecture de l'application permet de séparer totalement la couche de présentation, cette dernière étant placée au dessus de la logique applicative.
- 2 Développer des composants et des bibliothèques qui peuvent être réutilisés dans un maximum d'environnements.
- 3 Développer un seul projet, un seul code qui s'adaptera automatiquement à tous les environnements.

Les deux premières approches sont certainement les plus classiques et les plus logiques pour un développeur. Par exemple, pour développer un client pour LinkedIn, vous développeriez une couche de communication avec OAuth (open authentication), la logique de communication, et une couche de présentation des résultats dans différentes classes. Chaque plateforme aurait alors une couche de présentation adaptée. Si vous décidez d'adopter la troisième approche qui est plus osée mais possible, un code pour tous les écrans, voici quelques conseils issus des études de Christian Cantrell. Il a développé un jeu de reversi en Flash qui s'exécute dans le navigateur, sur le bureau (grâce à AIR), sur un iPad et un iPhone (grâce à l'iOS packager) et sous Android (AIR pour Android) et avec moins de 1500 lignes de code.

Un même code pour plusieurs écrans

Le premier conseil consiste à surveiller le changement d'orientation d'écran, une nouveauté pour les développeurs d'applications. L'API d'Adobe AIR permet de facilement détecter cet événement car la *Stage* génère un événement *StageOrientationEvent.ORIENTATION_CHANGE*. Vous pouvez aussi plus classiquement surveiller si la *Stage* est redimensionnée :

```
this.stage.addEventListener(Event.RESIZE, doLayout);
```

Pour récupérer l'orientation, comparer la largeur à la hauteur de la scène :

```
private function getOrientation():String
{
    return (this.stage.stageHeight >
        this.stage.stageWidth) ? PORTRAIT : LANDSCAPE;
}
```

Certaines plateformes possèdent des API qui permettent d'étendre le comportement de votre application comme l'accéléromètre sous Android ou iOS. L'API Flash fournit toujours une propriété booléenne pour vous assurer du bon fonctionnement d'une fonctionnalité matérielle sur une plateforme.

```
if (Accelerometer.isSupported)
{
    this.accelerometer = new Accelerometer();
    this.accelerometer.setRequestedUpdateInterval(1500);
    this.accelerometer.addEventListener(AccelerometerEvent.UPDATE,
        onAccelerometerUpdated);
}
```

La propriété *isSupported* est aussi présente sur les objets *GeoLocation*, *Multitouch*, l'accès à la caméra ou *StageWebView* par exemple. Pour l'affichage, Christian Cantrell utilise ensuite l'API de dessin de Flash pour générer les formes en fonction de la taille de l'écran. A partir de la taille de la *Stage*, Christian en déduit la taille de la zone de jeu (*board*). Il utilise l'API de dessin pour dessiner les pions et les transforme en bitmap pour des raisons d'optimisation des performances.

```
var cellSize:uint = (this.board.width / 8);
var stoneSize:uint = cellSize - 4;
var tmpStone:Sprite = new Sprite();
tmpStone.graphics.beginFill(WHITE_COLOR);
```



```
tmpStone.graphics.drawCircle(stoneSize/2, stoneSize/2, stoneSize/2);
tmpStone.graphics.endFill();
tmpStone.filters = [this.stoneBevel];
var tmpStoneBitmapData:BitmapData = new BitmapData(tmpStone.
width, tmpStone.height, true, 0x000000);
tmpStoneBitmapData.draw(tmpStone);
this.whiteStoneBitmap = new Bitmap(tmpStoneBitmapData);
```

Le code source est disponible et commenté sur le portail <http://developer.adobe.com>.

Flex 4.5, une nouvelle version pour gérer le multiscreen

Le framework Flex a permis dès son origine aux développeurs de créer des applications qui s'adaptent automatiquement aux différentes résolutions d'écrans. Depuis Flex 4, la définition de l'aspect visuel d'un composant est conservée dans une classe séparée, une classe de skin. Flex 4.5 fournit un nouveau jeu de skins pour créer des applications utilisables sur mobiles et qui gèrent le tactile. Néanmoins, les développeurs minutieux souhaiteront aussi adapter leurs interfaces et leurs comportements au DPI (ou ppi, 'pixels per Inch', la densité de pixels à l'écran), pour que l'interface soit toujours nette si vous utilisez des bitmaps et que le texte soit toujours parfaitement lisible.

Les exemples détaillés dans cet article sont issus de la documentation de Flex 4.5 encore en bêta. Il se peut qu'il y ait quelques différences quand la version finale de Flex 4.5 sera disponible au printemps. En tant que développeur, vous avez juste à spécifier pour quelle densité vous optimisez votre application. Si vous indiquez 160dpi et que votre application s'exécute sur un écran de 240dpi, Flex va automatiquement mettre à l'échelle votre application avec un facteur de 1,5.

```
<?xml version="1.0" encoding="utf-8"?>
<s:MobileApplication xmlns:fx="http://ns.adobe.com/mxml/2009"
  xmlns:s="library://ns.adobe.com/flex/spark"
  firstView="views.popupHome"

  applicationDPI="160">
  ...
</s:MobileApplication>
```

Les bitmaps comme vos icônes seront aussi automatiquement redimensionnées mais elles perdront en netteté. Pour détecter au runtime la densité de l'écran et charger en conséquence les bons graphiques, vous pouvez utiliser la classe FlexGlobals.

```
var curDensity:Number = FlexGlobals.topLevelApplication.run
timeDPI;
```

Et en déduire facilement le facteur de mise à l'échelle.

```
var currentFlexScalingFactor:Number = FlexGlobals.topLevel
Application.runtimeDPI /
  FlexGlobals.topLevelApplication.applicationDPI;
```

Vous pouvez aussi alimenter votre application avec plusieurs bitmaps de plusieurs densités et les charger automatiquement en fonction du DPI. Il suffit d'utiliser le nouveau composant MultiDPIBitmapSource et de le combiner avec le composant Image.

```
<s:Image>
  <s:source>
    <s:MultiDPIBitmapSource
      source160dpi="logo.png"
      source240dpi="logo240.png"
      source320dpi="logo320.png" />
  </s:source>
</s:Image>
```

Pour une définition plus fine des composants à utiliser dans votre application, Flex 4.5 va aussi introduire un concept connu des développeurs HTML, les CSS Media Queries. Ainsi les développeurs pourront définir dans des fichiers CSS les éléments et propriétés propres à certains OS et certaines densités d'écran. Le media de type « screen » sera supporté avec deux propriétés personnalisées : os-platform et application-dpi. En fonction de la plateforme (iOS, Android...) et de la densité, vous pourrez facilement modifier le comportement visuel de votre application.

```
<fx:Style>
  @namespace s "library://ns.adobe.com/flex/spark";
  @namespace mx "library://ns.adobe.com/flex/mx";

  s|Button
  {
    fontSize: 12;
  }

  @media all and (application-dpi: 160)
  {
    s|Button
    {
      fontSize: 10;
    }
  }
  @media (application-dpi: 240) and (os-platform: "IOS")
  {
    s|Button
    {
      fontSize: 11;
    }
  }
  @media (application-dpi: 240) and (os-platform: "Android")
  {
    s|Button
    {
      color: 13;
    }
  }
</fx:Style>
```

Les autres défis du multiscreen et roadmap

Les bonnes pratiques de développement pour le multiscreen sont facilitées par deux choses. Les API du runtime (comme la gestion du dpi par exemple) et la structure de son code (framework MVC comme Parsleys, la création de bibliothèques partagées, etc.). L'autre défi du multiscreen que se propose de résoudre la Flash Platform est le déploiement de vos applications sur plusieurs plateformes,



ainsi que l'expérience de debug. La nouvelle version de l'IDE Flash Builder (plug-in Eclipse) permet de développer des applications en ActionScript3, avec ou sans le framework Flex, et de les tester sous différents OS en plus des classiques Windows, MAC OS et Linux. La plateforme Android est déjà disponible. Vous pouvez tester votre application soit en utilisant l'émulateur AIR pour Android disponible sur Windows et MAC, ou sinon vous pouvez directement déployer et déboguer votre application sur un mobile Android (minimum Android 2.2) équipé d'Adobe AIR. Il existe déjà plus de 1500 applications AIR sur le marketplace Android. L'autre plateforme qui est aussi prête est le BlackBerry Tablet OS. RIM lance sa première tablette, la PlayBook, basée sur l'OS QNX. Le runtime AIR est présent par défaut et permet d'exécuter vos développements. Flex est le framework de développement conseillé pour le PlayBook. Pour iOS, Adobe a développé l'iOS Packager qui convertit une application Flash en application native iOS. Ce packager sera mis à jour au printemps pour revenir au niveau de AIR pour

Android au niveau des fonctionnalités mais aussi des performances. En effet, Adobe avait stoppé la R&D sur cette technologie suite aux restrictions d'Apple finalement levées 6 mois plus tard. Quant à la télévision, le Flash Player est présent dans la Google TV et les futures télévisions Samsung embarqueront par défaut Adobe AIR pour exécuter vos développements.

Pour conclure je dirais qu'il est parfaitement réaliste de développer une fois pour plusieurs écrans avec l'Adobe Flash Platform. Mais il ne faut pas que cela devienne un raccourci de développement. Le multiscreen doit aussi être l'occasion de profiter de toutes les nouvelles API et capacités des appareils. Grâce à la Flash Platform, vous pouvez développer une fois, mais surtout industrialiser vos développements et minimiser vos investissements pour proposer des interfaces riches sur un maximum de plateformes et d'appareils de nouvelle génération.

■ Michael Chaize - Adobe France



Témoignage

David Deraedt : un consultant Flash dévoile son approche

David est consultant sur les produits et technologies web, Flash et html, chez Adobe France. Avant de rejoindre Adobe, il a été développeur ActionScript, puis formateur et consultant Flex indépendant. Cela fera 10 ans cette année qu'il travaille dans le web.

Comme beaucoup de développeurs, il passe aujourd'hui énormément de temps sur la question de la création d'applications mobiles -et plus généralement le multiscreen- car c'est quelque chose de terriblement excitant, mais aussi un vrai challenge technique.

Tu as développé avec la Flash Platform une application multiscreen. Peux-tu nous décrire cette application et nous expliquer ce que tu entends par multiscreen ?

Une application multiscreen est une application qui existe, sous une forme ou une autre, sur des terminaux aussi différents que le smartphone, la tablette, la télévision, ou encore le bon vieux PC.

Face à la quantité d'appareils et de systèmes d'exploitation disponibles, il me paraît difficile de redévelopper entièrement une application de A à Z pour chaque contexte avec les outils natifs disponibles pour chaque plateforme. J'ai donc cherché à évaluer différentes technologies dites "cross-platform", c'est-à-dire compatibles avec différents systèmes, afin d'évaluer les avantages et les inconvénients de chacune de ces technologies. Mes investigations se sont portées à la fois sur les solutions basées sur les standards du web et sur la Flash Platform. Concernant cette deuxième solution, j'ai notamment développé un mini-jeu pour enfants permettant d'apprendre l'alphabet.

Quels sont tes sentiments par rapport à l'approche « un seul code pour tous mes écrans » ?

Je pense que c'est séduisant sur le papier, mais que dans les faits, c'est bien plus complexe. Les terminaux actuels sont tellement différents que, quand bien même une technologie nous permettrait de porter le code d'une application de manière parfaitement optimisée, ce ne serait pas forcément une bonne idée. En effet, je ne pense pas que les utilisateurs veuillent la même expérience quel que soit le contexte. Lorsqu'on utilise un smartphone, dans le métro, pour chercher une information rapidement, on ne s'attend pas à la même expérience que sur son ordinateur, au travail, avec son clavier et sa souris, ou encore que sur sa télévision avec sa télécommande. On peut chercher à accéder à un contenu commun, à un même service, mais pas de la même manière. Même entre deux appareils comparables, par exemple deux smartphones comme l'iPhone et un téléphone Android, il existe des subtilités dans l'intégration au système qu'il faut prendre en compte sous peine de frustrer ou de décevoir l'utilisateur.

D'autres leçons tirées de ton développement ?

Aujourd'hui, aucune solution cross-platform ne peut remplacer définitivement les outils natifs de chaque appareil. Il existera toujours des contextes dans lesquels il vaudra

mieux développer avec ces outils natifs, qui sont par définition les mieux à même de créer du contenu pour leurs plateformes respectives. Notamment, les questions de performances sont fondamentales, et les solutions cross-platform ne peuvent égaler les solutions natives, pour autant que l'on sache bien s'en servir. Cependant, ces solutions, quand elles sont utilisées à bon escient, peuvent permettre des gains de productivité considérables et un confort de développement accru.

Au final, pour quels types d'applications mobiles recommanderais-tu la Flash Platform ?

Aujourd'hui, je pense que les options d'export mobile de Flash Pro sont bien adaptées aux contenus créatifs et originaux, tels que les jeux vidéos, les applications artistiques, les animations interactives, et d'une manière générale tout ce qui ne correspond pas aux applications mobiles classiques à base de boutons, de listes déroulantes et d'éléments d'interfaces utilisateurs natifs.

Pour créer ce deuxième type d'applications, je pense qu'il faudra attendre l'arrivée de "Flex hero", la nouvelle version de notre framework de RIA, qui est précisément dédié à la création d'applications mobiles à base d'éléments d'interfaces utilisateurs, mais là encore, dans un contexte cross-platform.



Retour sur les TechDays 2011

Les recrutements repartent très fort !

Les recruteurs, SSII et éditeurs font face à une reprise très forte des projets et sont nombreux à rechercher des talents. En particulier sur les technologies Microsoft. Les TechDays 2011 ont été l'occasion de rencontrer plusieurs recruteurs.

En 2007 le G20 pariait déjà sur les SSII, considérées comme des acteurs importants de croissance économique, en effet leur développement permettrait la création de plusieurs milliers d'emplois pour la période 2010/15, témoignant ainsi de la santé des secteurs qui feraient appel à elles.

Aujourd'hui, les projets explosent dans tous les secteurs. Les entreprises investissent sur du long terme, il s'agit de faciliter la gestion interne et regrouper au mieux les services pour les rendre de plus en plus opérationnels et compétitifs. Quant au produit fini il devient de plus en plus contrôlable et identifiable ; ce qui permet une meilleure gestion de la relation client.

Comment les SSII répondent à ces demandes ? Qui sont-elles ? Avec qui travaillent-elles ? Comment gèrent-elles cette croissance d'activité ? Quels sont leurs besoins ? Qui sont leurs futurs collaborateurs ? Des professionnels du recrutement que nous avons rencontrés nous répondent. Alors que les entreprises favorisent un travail de plus en plus collaboratif et polyvalent les SSII recherchent des experts de plus en plus pointus.

Ingénieurs .net et en solutions Microsoft sont devenus une espèce rare

Malgré un marché en pleine expansion, les SSII évoluent dans un microcosme accentué par une pénurie de compétences techniques particulières en programmation. La concurrence y est rude et les candidats sont recherchés. Les ingénieurs .net sont devenus une espèce rare alors même que les entreprises explosent en termes de projets et d'objectifs.

Finalement cette situation pousse ces entreprises à se dépasser pour fidéliser leurs

salariés, leurs ingénieurs : salaires, formation, connaissances, avantages, atmosphère de travail, développement de carrières. C'est un secteur où la gestion des ressources humaines devient un nouvel enjeu de croissance. Chacun son processus de recrutement : cooptation, réseau, chasseurs, jobboard, relation école. Chacun ses objectifs : fidéliser, pérenniser, attirer ses collaborateurs. Industrialisation, expertise, qualité, suivi. De la grande multinationale à la petite PME, il y en a pour toutes les personnalités et l'emploi y est prospère. Reste un problème : comment expliquer cette pénurie, avec un taux de chômage aussi élevé ?

Les écoles françaises forment-elles suffisamment d'étudiants aux nouveaux défis IT, l'éducation nationale doit-elle mieux répondre aux exigences de l'économie et de ces entreprises ?



Le point avec les SSII

OSIATIS est une SSII présente depuis 1998, dont le cœur de métier est l'infogérance, l'infrastructure, l'ingénierie applicative. Elle compte 3 000 personnes en France, aux Pays Bas et au Luxembourg. Avec 233 millions de chiffre d'affaires, elle emploie en France 450 ingénieurs en conception et développement orienté .net et méthode agiles, 250 stagiaires et contrats en alternance. Cette SSII travaille essentiellement avec de grosses entreprises : Total, Carrefour, La banque postale, Dassault, Bred, et investit de plus en plus le domaine public. Depuis

“ Pour les seniors, nous ne regardons plus les diplômes mais l'expérience ”

98, Elle intervient dans l'infogérance des postes de travail mais tend vers une valeur ajoutée qui demande des profils de plus en plus pointus, dans le domaine du Cloud et de la virtualisation. Cet engagement fort, se traduit par un partenariat avec Microsoft, c'est une nouvelle direction qui était à l'honneur sur le salon TechDays 2011 au Palais des Congrès. « *L'évolution des recrutements est claire depuis septembre* », explique Virginie Leforestier consultante en

recrutement, « *les chiffres ont doublé depuis l'année dernière* ». 500 personnes devraient être embauchées sur l'année 2011, de bac+2 à bac+5. Les bacs+2 sont formés et débutent sur Help desk, les

jeunes ingénieurs diplômés débutent en virtualisation, par contre il y a peu de débutants en expertise technique.

Osiatis a des partenariats avec des écoles et rencontre les étudiants, ce qui permet une approche plus qualitative. Il est important de partager des valeurs communes. Un recrutement « *c'est une rencontre, c'est donc du service et pas seulement un diplôme* ». La transparence, l'engagement, le respect, le progrès, la communication, la souplesse... Autant de qualités recherchées pour être un bon consultant.

Une fois qu'il est trouvé, Osiatis fait tout pour le garder. Le taux d'investissement en formation est de 3% de la masse salariale, et 0,2% de plus que l'année précédente, ainsi les salariés ont la possibilité de rester toujours à la pointe de l'évolution technologique. Et on reste à leur écoute, si jamais il y a une demande de changement ou d'évolution de carrière...

AVANADE s'occupe des systèmes d'intégration Microsoft et travaille essentiellement

“ Depuis Mai 2009 il y a plus de projets que de ressources ”

pour Accenture. Spécialisée en infrastructure, ERP, application, CRM, ses consultants sont des experts techniques avec une spécialisation en mode projet. La société compte 350 experts, avec l'objectif d'atteindre les 450 à la fin 2011. Les salariés se divisent en trois catégories, les bac+2/3, les Juniors (2/3 ans d'expériences) et les Seniors (plus de 5 ans d'expérience). Il y a eu un ralentissement des recrutements en 2009, même si les projets ont diminué, la logique de recrutement a prévalu pour ne pas passer à côté des bons profils, « c'était justement le moment de récupérer les meilleurs » nous dit Marion Enjoiras, Manager du recrutement. Aujourd'hui, l'activité est croissante dans tous les domaines : infrastructure, applicatif, CRM, fonctionnel, ERP et dernièrement en Virtualisation et cloud. On peut noter une évolution claire des projets depuis octobre/novembre 2010 dans tous les secteurs surtout dans les entreprises du CAC 40. « Nous recherchons des personnes passionnées par les technologies Microsoft, .net. Pour travailler sur des projets en développement de logiciels, il faut de la personnalité, savoir travailler en équipe et avoir une bonne communication pour pouvoir être en relation avec le client. Les soft skills sont importants chez nous, des geeks purs on en a, mais ils sont limités en termes de carrière. » Un suivi est proposé pour l'évolution de carrière, il y a de l'échange de connaissances en interne et chacun peut bénéficier de formations. La société a reçu le titre de « Top Employeur » 2010. « Nous recrutons grâce aux jobboards, aux cabinets de recrutements, nous passons aussi par les relations écoles, et de plus en plus par la cooptation... ».



CAP GEMINI, c'est 100 000 employés dans le monde, 25 000 en France dont 400 ingénieurs sur les technologies Microsoft. La société fait de plus en plus appel aux

étudiants en alternance et aux stagiaires. Elle mise beaucoup sur la formation en interne, les enseignements dans les écoles d'ingénieurs sont toujours sur des langages Java et C++, alors que la plupart des systèmes en entreprises sont en .Net. On peut

ajouter à cela une pénurie de spécialistes SharePoint. R. Muller responsable SP chez Cap Gemini nous parle de la difficulté à trouver des consultants alors que les entreprises sont de plus en plus demandeuses de cette plateforme. Entre les entreprises qui travaillent avec la version 2010 et celles qui ont actualisé leur licence 2007, les interventions ne sont pas toujours faciles. « C'est comme dans le foot, il y a très peu de compétences SP, alors les entreprises font des transferts de salariés avec des salaires qui ne sont pas raisonnables mais la rareté des compétences face à la demande croissante des entreprises provoque cette surenchère ». Les secteurs avec lesquels Cap Gemini travaille sont essentiellement la finance, la banque, l'énergie, quelques entreprises du BTP et de plus en plus avec le secteur public. Pendant la période de crise, le siège a décidé de faire un gel du recrutement alors que l'activité dans le domaine des IT n'était pas réellement touchée. « La crise n'a pas fondamentalement changé les usages des DSI ». Ce qui a pénalisé leur croissance, les chefs de projets ont souffert de ce gel d'un point de vue opérationnel.

Les recrutements aujourd'hui passent essentiellement par les relations écoles, le réseau, les sites, les salons mais aussi la cooptation. Pour les plus expérimentés (5 ans d'expérience) nous faisons appel à des cabinets de chasse car le milieu des IT est un microcosme. « Pour les seniors nous ne regardons plus les diplômes mais l'expérience qu'ils ont pu avoir chez les concurrents. Si une personne n'a pas fait de service mais qu'elle a essentiellement travaillé dans une entreprise, ce sera plus difficile car le milieu du service IT est différent ».

« Nous ne croyons pas au 'tout-Cloud', il va y avoir du mixte car il y a plusieurs problématiques qui vont se présenter, par exemple la confidentialité, la géographie aussi car Microsoft n'a pas de contrats dans tous les pays. » C'est comme l'Offshore, les entreprises ont tout transféré puis elles se sont rendu compte qu'il fallait s'adapter aux besoins de chacun des clients et de leur activité...

NELITE a fêté ses 3 ans en janvier 2011, c'est une société de conseil et d'intégration en infrastructure Microsoft, à taille humaine, et qui tient à le rester. Son objectif principal est de garantir un fort niveau

d'expertise. Elle comprend deux niveaux : les managers et les consultants techniques qui deviennent de plus en plus spécialisés. L'entreprise garantit une infrastructure avec une nouvelle technologie et l'accompagne tout au long de sa mise en place. Elle développe de plus en plus un rôle de conseil et s'ouvre aux Etats-Unis et au Maroc, le but étant de garder une stratégie commune pour chacune des structures et de centraliser la direction. Le groupe a pour objectif de passer de 80 à 120 personnes d'ici fin 2011. Jean-Michel Renard nous parle des profils recherchés : « On cherche des personnes qui ont une compréhension des besoins métiers, il faut une affinité technique mais surtout une personnalité. Un attrait pour les techniques Microsoft est nécessaire (système center, communication unifiée, active directory), on connaît nos clients alors on forme en fonction des besoins ». Nelite propose à ses consultants des ateliers forums toutes les semaines, il y a une transmission et une vulgarisation des technologies « Nous ne sommes pas élitistes sur les études mais davantage sur la capacité à apprendre. On essaye de fidéliser nos salariés en leur proposant un environnement technologique et de connaissance riche. On propose une rémunération correcte, pour les juniors de 38 à 48 K€. Si un consultant est reconnu dans l'écosystème, et qu'il semble dans la dynamique de l'entreprise, on n'hésite pas à mettre le prix pour l'embaucher ». Avant, la chasse représentait 80 % de ses recrutements, aujourd'hui c'est 20%, car Nelite a de plus en plus de notoriété.

SOFTFLUENT est un éditeur développant un applicatif sur Visual Studio 2010. Cet outil développé sur .net permet de faciliter la mise en place d'applications. L'éditeur propose également du service sur les technologies Microsoft, « on a transformé le job du développeur, car il faut être à l'écoute du client et coder d'un point de vue métier, ce n'est plus de la technique pure. » La société travaille avec tout ce qui est autour de .Net, SQL, Azure mais pas encore Share Point. Softfluent pourrait passer de 15 à 20 consultants d'ici la fin de l'année. « On recherche des technophiles passionnés par les métiers du client... Il faut comprendre le métier du client pour lui trouver une solution technologique »

« On prend des jeunes ingénieurs développeurs et on les forme à un savoir-être ! Mais pour cela il faut qu'ils soient autonomes, polyvalents, pédagogues et techniques... Nous sommes des hommes du service », explique Lionel Poli, directeur commercial. Nous ne sommes pas intéressés par des collaborateurs passionnés par l'argent mais par les technologies et le service. On ne donnera pas 40 K€ à un 'gamin' qui sort de l'école ». Les collaborateurs de Softfluent « travaillent sur des missions de 3 à 6 mois grand maximum, il y a un suivi, une écoute pour savoir ce qu'ils veulent faire demain... » L'entreprise n'a pas de turn over et elle a pour objectif de se développer intelligemment avec un service de qualité et pas forcément en volume. Softfluent travaille actuellement à des projets avec de grandes banques ou des starts up, sur des surfaces utilisateurs ou de la refonte sur Dynamics et du business intelligence.

EXAKIS compte 300 personnes, son effectif a augmenté de 5 à 10 % chaque année depuis 10 ans. C'est l'un des plus gros intégrateurs de l'offre Microsoft en infrastructure, développement, sécurité SI, AMOA, pilotage... La société travaille essentiellement avec des entreprises du CAC 40 dans de multiples secteurs, la grande distribution, l'énergie, la finance, etc. Elle se développe de plus en plus vers l'industrialisation des postes de travail, la virtualisation, le Cloud et les problématiques Azure.

« Chez Exakis la capitalisation est un mot clé, l'objectif est de doubler d'ici deux ans ». Son organisation est rodée, les processus sont clairs et un suivi professionnel a été mis en place, l'entreprise est dans une industrialisation du recrutement. Pour Sophie Sainte Thérèse chargée du recrutement « Le vecteur différentiel n'est pas que le salaire, il y a aussi l'action marketing social. Il faut former et certifier les collaborateurs, il existe pour cela des managers de proximité et l'équipe RH reste à l'écoute... Nous recherchons des personnes passionnées par l'univers Microsoft qui sortent d'une école d'ingénieurs avec un master informatique, idéalement avec un stage de 6 mois effectué sur des produits Microsoft. »

Pour les experts, ils doivent faire partie de l'écosystème et être connus dans le monde Microsoft. Ils doivent pouvoir parler anglais, faire preuve de compréhension et analyser les problématiques sur lesquels ils interviennent. C'est le « geek consultant » qui est

recherché aujourd'hui, le technicien communiquant. Le « bouche à oreille », le réseau interne, les relations écoles, la cooptation, les réseaux sociaux, la chasse sont les canaux les plus utilisés, « il faut sortir des circuits classiques ».

MCNEXT est une société d'ingénierie et de conseil en solutions Microsoft créée il y a 4 ans, elle comprend 4 pôles d'activités : Sharepoint, .net, BizTalk et Décisionnel qui est le pôle d'accompagnement global dans la mise en place de solutions de pilotage. Chaque pôle est dirigé par un leader technique reconnu dans le milieu comme leader dans sa spécialité. Aujourd'hui l'entreprise compte une centaine de collaborateurs et a le projet de recruter une quarantaine de personnes en 2011 pour répondre aux demandes croissantes du marché. C'est une entreprise à taille humaine en pleine évolution comme toutes celles proposant des solutions de technologie Microsoft. Leader technique dans son domaine elle est certifiée Goldpartner.

Pour MCNEXT, garder un très haut niveau technique est un mot d'ordre ; c'est un défi qu'ils essaient de relever au quotidien, 80 à 90% de leurs collaborateurs sont certifiés. Ils proposent des formations MCT (Microsoft Certified Trainer), « nous accompagnons chacun dans son souhait de carrière, nous connaissons tout le monde il y a un management de proximité » pour Magalie Perret, chargée de recrutement chez MCNEXT, la division de l'activité par pôle permet de mieux connaître les collaborateurs et de les accompagner au quotidien dans leurs demandes.

Des colloques et des séminaires sont organisés pour faire de la vulgarisation autour des technologies .Net et de la méthode agile. Pas de profil type, mais il faut tout de même avoir un bac+3 au minimum, type MAGE, école d'ingénieur ou informatique, car les clients sont exigeants. Les clients sont des



entreprises du CAC 40 (Grandes banques, industrie, tourisme, énergie, services, telecom), « aujourd'hui il y a plus de clients que de collaborateurs, cela permet à chacun de travailler là où il se sent le mieux selon ses compétences et ses ressources ». Leurs canaux de recrutements sont multiples, cv-thèque, annonces sur jobboard, partenariat avec des écoles comme Supinfo, Epitech, présence dans des salons, etc. Pour Magalie Perret, les personnes sont une priorité « il n'y a pas de parcours tracé, il faut accompagner chacun vers ses objectifs personnels, on est une société sérieuse mais on ne se prend pas au sérieux... avec le rire, on est plus efficace ».

CODIT est une société de services d'ingénierie informatique basée en Belgique depuis 1999 et en France depuis 2004. Son activité principale consiste en l'intégration de système de technologies certifiées Microsoft et plus particulièrement BizTalk. Elle compte 40 personnes aujourd'hui et veut se développer pour répondre aux demandes croissantes des grands comptes avec qui elle

“ Il n'y a pas de parcours tracé, il faut accompagner chacun vers ses objectifs personnels ”

travaille. Avec une équipe de 10 personnes basée en France, Codit souhaite augmenter le nombre de ses collaborateurs de 50%, soit 5 consultants de plus pour la fin 2011, « Nous avons plus de demandes que de capacités à livrer... », nous

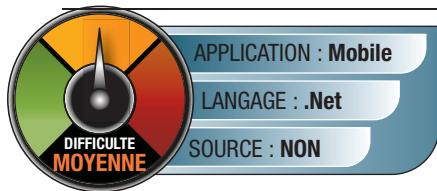
explique Mr Boubil, Account Manager. L'entreprise travaille avec les secteurs de l'industrie, les finances, l'agroalimentaire, l'industrie pharmaceutique, et les projets sont en perpétuelle évolution depuis sa création. Les ingénieurs recherchés doivent avoir un profil de développeur .Net et BizTalk, « mais nous sommes conscients que les jeunes diplômés ne connaissent pas tous BizTalk, s'ils ont déjà travaillé sous C# et ont une bonne connaissance de SQL Server : xml, xsd, xslt, nous leur offrons une formation qui leur permettra une montée en compétences. »

« La cooptation est le canal de recrutement principal, qui mieux que les collaborateurs eux-mêmes peut savoir si des personnes sont capables de répondre aux besoins de l'entreprise, nos collaborateurs sont nos principaux ambassadeurs ».

■ Sonia Nabih

Analyse de l'ergonomie sur Windows Phone 7

Il y a deux ans, Microsoft a décidé de rénover l'interface de Windows Phone en repartant de zéro. Ce choix radical entraîna une refonte complète des principes utilisés jusqu'alors par l'équipe de Redmond. En s'inspirant de designs tels que la signalétique du métro de New York (mise en pratique préalablement sur le Zune HD et la XBOX 360), une ihm novatrice a été mise en place en minimisant au maximum tout ce qui n'est pas de l'information. Nous allons voir ensemble les différents principes de l'interface Métro, et notamment les choix ergonomiques qui ont été faits, pour proposer une expérience différente des smartphones actuels.



En nommant son interface Métro, la nouvelle équipe en place a décidé de mettre en avant la modernité, la lisibilité et la fluidité des informations que l'on retrouve dans la signalétique urbaine. Pour cela, les points centraux suivants ont été posés pour définir la base et le caractère que doit dégager Métro :

- "Clean, Light, Open and Fast" & "Content, Not Chrome"
Ne garder que les éléments qui sont nécessaires en privilégiant le contenu (l'information) au contenant, en réduisant celui-ci au maximum. Privilégier l'interaction directe entre l'utilisateur et l'information.
- "Celebrate Typography"
Axer l'effort sur la qualité graphique de la typographie afin de la rendre lisible et agréable.
- "Authentically Digital"
Minimiser les métaphores qui reproduisent des objets réels, éviter les effets graphiques qui vont dans ce sens.
- "Alive in Motion"
Donner du caractère à l'ihm en utilisant des transitions rapides entre les pages qui mettent en avant à la fois l'interaction et le contenu.

Il est à noter que des « personnas » ont été utilisés pour définir les aspects centraux de Metro (Un persona est un personnage fictif représentant un type d'utilisateur employé dans la conception centrée utilisateurs). En regardant de plus près ces aspects, on se rend compte que les deux premiers axes majeurs ressemblent à l'heuristique n°8 de Jakob Nielsen (expert en utilisabilité des interactions homme-machine) :

« Les dialogues ne devraient pas contenir des informations qui sont non pertinentes ou rarement utilisées. Toute information superflue dans un dialogue entre en compétition avec les informations pertinentes et diminue la visibilité relative de ces dernières. » *

*Jakob Nielsen, "Chapter 5 : Usability Heuristics", in Usability Engineering, Academic Press, 1993.

Cette heuristique intitulée « Design esthétique et minimaliste » représente l'un des dix principes généraux pour la conception d'interactions homme-machine utiles et utilisables. C'est cette direction

qui a été suivie afin qu'aucun contenant ne vienne parasiter l'information. L'utilisation des métaphores a ainsi été mise de côté pour privilégier cet aspect. De cette façon, l'utilisateur se retrouve plus impliqué pour appréhender et interagir avec ses informations et seulement celles-ci. Cette orientation est clairement à contre-courant de ce qui se fait actuellement depuis plus de 25 ans dans le domaine de l'interaction homme-machine. Généralement, la métaphore physique est souvent utilisée pour orienter l'utilisateur dans les interactions qu'il peut effectuer en lui rappelant un contexte réel. L'effort important a donc été porté sur la qualité de la typographie. Une écriture claire et visible, inspirée de la signalisation métropolitaine. Accentuant ainsi, la mise en avant du contenu derrière lequel les visuels non informatifs sont secondaires.

LE SMARTPHONE, UNE UTILISATION À PART

L'utilisation d'un Smartphone n'est pas la même que celle d'un ordinateur. L'utilisateur a généralement, en situation de mobilité, besoin d'accéder très rapidement à son information. Pour garantir ce constat d'utilisation, un cahier des charges strict des spécifications techniques minimales a été imposé afin d'assurer des transitions donnant une sensation de rapidité et de réactivité sur ses terminaux. Nous verrons dans les faits que si la sensation de réactivité est excellente, des améliorations sont à apporter du côté de la rapidité des animations.

TUILES, HUBS, RÉACTIVITÉ ET CONNECTIVITÉ : LE CŒUR DE WINDOWS PHONE 7

Un écran de déverrouillage qui donne le ton

Lorsque l'on sort de veille un téléphone Windows Phone 7, on se rend instantanément compte de l'expérience qui va nous être proposée. Nous avons accès à l'heure, la date, le prochain événement calendrier ainsi que les sms et mails non lus. Un nombre de paramètres supérieurs ou égaux à ce que l'on trouve chez la concurrence, ce qui est fort appréciable. Contrairement aux autres OS mobiles, aucune indication n'est présente sur la manière de déverrouiller son téléphone (pas de métaphore). Il faut effectuer un geste vers le haut pour soulever « le panneau » (cette animation est cependant réactive et fluide).

Le fonctionnement et la disposition de cet écran résument parfaitement le parti pris de Redmond avec WP7. Seulement, ce choix

entraîne irrémédiablement une affordance qui sera plus faible que sur les systèmes utilisant la métaphore graphique tels que iOS ou encore, dans une moindre mesure, Android.

Les tuiles : plus que des icônes

En appuyant sur le bouton Windows du téléphone, nous accédons à la page d'accueil composée de ce qu'on appelle « les tuiles ». Les tuiles ont un rôle semblable aux icônes que l'on retrouve sur les autres OS mobiles. A la différence près, que le contenu d'une tuile peut être dynamique et visible sur la page d'accueil. Ainsi il est possible d'obtenir des informations sans même interagir avec cette icône, ce concept est à saluer car il peut permettre de gagner du temps en regardant par exemple le prochain rendez-vous professionnel directement de sa page d'accueil. La philosophie Microsoft est pleinement appliquée en proposant des tuiles de grande taille et de couleurs uniformes mettant en valeur seulement la typographie.

On peut donc qualifier ces tuiles « d'icônes affichant du contenu dynamique » sans toutefois être configurables (ce ne sont pas des widgets). Comme sous iOS, il est possible de changer le placement de la tuile en pressant dessus un certain temps. Les feedbacks sont bien présents pour ne jamais mettre en porte-à-faux l'utilisateur.

A droite des tuiles, se trouve une flèche qui nous emmène sur une page classique où est disposé l'ensemble des applications par ordre alphabétique. Cette partie sert plutôt de page mémento pour les applications qui sont peu utilisées.

La page d'accueil permet de mettre en avant la réactivité et la fluidité du système lors du défilement des différentes tuiles. Axe central de WP7, la réactivité est vraiment au rendez-vous. Un excellent point qui ne fait que renforcer l'expérience utilisateur sur smartphone. Certaines animations sont cependant bien trop longues (même si réussies) et engendrent une sensation de frustration alors que le système est pourtant bien réactif.

Hubs et connectivité : un agencement efficace

Les différents hubs représentent le souci porté pour proposer une rupture avec l'existant. Pour résumer, un hub est un centre d'intérêt qui regroupe des informations liées entre elles (ou en rapport avec le centre d'intérêt). Celles-ci sont représentées sur plusieurs écrans de la gauche vers la droite.

La page suivante (ou antérieure) est présente légèrement sur le côté pour faire comprendre à l'utilisateur qu'un geste vers cet endroit lui permet d'y accéder.

Cet agencement panoramique est novateur et particulièrement astucieux à l'heure du web 2.0 et de ses réseaux sociaux. Par exemple, il est possible d'envoyer un sms à un contact tout en regardant ses informations postées sur les réseaux sociaux dans les différentes pages du centre d'intérêt. Là encore, nous voyons à quel point l'expérience utilisateur de ce système est en accord avec les besoins apparus ces dernières années.

On regrettera que cette partie aille légèrement à contre-courant avec les guidelines UI. Les fonds d'écran de ces hubs étant souvent trop chargés et n'offrent ainsi pas le même niveau de clarté et de visibilité que l'on pouvait observer précédemment.

Navigation web et multimédia : réactivité et simplicité

La navigation sur internet étant l'un des nerfs de la guerre des OS mobiles, internet explorer a été particulièrement soigné. Le naviga-

teur est globalement rapide et très réactif quelle que soit l'interaction effectuée. Il est réellement agréable de surfer sur le web avec ce système. On regrettera que la taille des touches du clavier de saisie soit un peu petite en mode paysage. Côté multimédia, la disposition et le fonctionnement sont semblables à ce qui se fait sur le Zune HD. On est donc sur une valeur sûre avec une interface moderne, très agréable et homogène par rapport à ce que nous avons vu précédemment. Globalement, l'ensemble de l'interface de Windows phone 7 est cohérent par rapport aux axes centraux fixés par l'équipe de Redmond.

WINDOWS PHONE 7 : UNE INTERFACE INTUITIVE ET HOMOGÈNE AVEC DES DÉFAUTS DE JEUNESSE

La refonte globale de l'interface de Windows phone 7 a bien eu lieu. En se basant sur des critères ergonomiques forts et sur un parti pris à contre-courant (pas de métaphore). Microsoft propose une expérience utilisateur totalement en phase avec les nouveaux besoins des utilisateurs de smartphones.

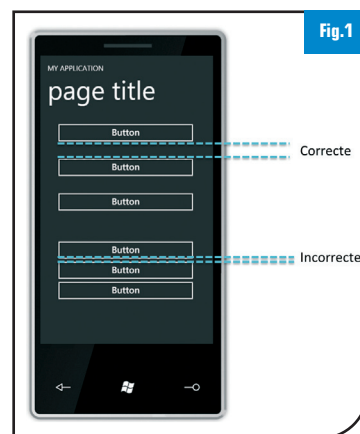
L'interface de Windows Phone 7 est axée sur la simplicité, la réactivité, la connectivité et la mise en valeur de l'information. Certains défauts de jeunesse sont bien évidemment présents : animations trop longues, affordance par moment faible, lisibilité qui fait parfois défaut. Mais dans l'ensemble, le pari est réussi. Windows phone 7 est réellement intuitif et agréable, tout en ayant le mérite d'être différent de la concurrence.

Cependant il ne faut pas perdre de vue que beaucoup de fonctionnalités sont manquantes, et c'est bien là le défi de ces prochains mois pour la firme de Redmond : arriver à garder cette interface intuitive et cohérente. Un enjeu majeur vis-à-vis de la concurrence, à n'en pas douter.

Respect des tailles minimales

Dans les règles que propose Microsoft on retrouve, par exemple, des tailles minimales à respecter pour une application tactile.

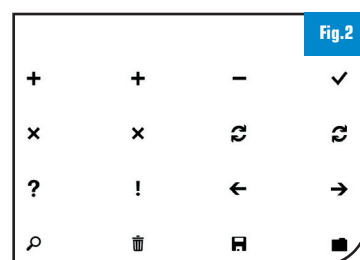
- Une zone "touchable" par l'utilisateur ne doit pas être inférieure à 34px² (et franchement c'est un minimum ! Faites le test).
- Chaque élément graphique doit faire 60-100% de la zone tactile.
- L'espace minimum entre deux zones "touchables" est de 8px [Fig.1].



Utilisation des icônes standard

Microsoft met à disposition une suite d'icônes vectorielles installée avec le SDK à cet endroit : C:\Program Files (x86)\Microsoft SDKs\Windows Phone\7.0\Icons [Fig.2].

Ces 32 icônes permettent de garder une certaine



homogénéité entre les applications WP7, entre autres, pour l'application *bar*.

La construction d'une page

Savoir construire une page est essentiel et nous avons de la chance puisqu'à la création d'un nouveau projet, Visual Studio/Blend le fait pour nous [Fig.3]. Comme nous pouvons le voir dans l'application ci-dessous, il y a trois zones importantes délimitées ici par un trait rouge :

- Le frame control : géré par le système
- Le page control : l'entête de notre page sous la forme Titre/Sous-titre
- Le contenu

La séparation entre page control et le contenu est souvent réalisée par une grid contenant deux lignes.

Laisser le frame control à l'utilisateur

Le frame control permet d'afficher l'heure, l'état du réseau... toutes les informations essentielles à votre téléphone ! Malheureusement, certains développeurs abusent (au moins une application sur deux sur le market actuel) du mode « fullscreen » que nous pouvons activer de cette manière :

```
1 shell:SystemTray.IsVisible="True"
```

Le problème de ce mode est qu'il cache tout simplement le frame control. Résultat, lorsque nous lançons une application, nous n'avons plus l'heure et nous sommes obligés de revenir sur la *home*. Heureusement, toutes les applications ne fonctionnent pas comme cela, à l'image de Beezz, superbe application twitter, qui s'ouvre en fullscreen mais qui laisse la possibilité à l'utilisateur d'afficher le frame control en touchant le haut de l'écran. Comment faire ? C'est très simple :

- Mettre l'option `SystemTray.IsVisible` à `false`
- Implémenter une méthode `DisplaySystem` dans vos IHM pour afficher la barre
- Mettre un bouton transparent au-dessus de tous les écrans pour capter le « touch » et appelez la fonction

La fonction `DisplaySystem` ressemble à ceci (utilisation de RX Framework disponible en ajoutant la référence à `Microsoft.Phone.Reactive.dll`) :

```
1
2 public static void DisplaySystemTray()
3
```

```
4 {
5
6 Microsoft.Phone.Shell.SystemTray.IsVisible = true;
7
8 RootFrame.Margin = new Thickness(0, -32, 0, 0);
9
10 Observable.Start(() =>
11 {
12 {
13
14 Thread.Sleep(4000);
15
16 RootFrame.Dispatcher.BeginInvoke(() =>
17 {
18 {
19
20 RootFrame.Margin = new Thickness(0, 0, 0, 0);
21
22 Microsoft.Phone.Shell.SystemTray.IsVisible = false;
23
24 });
25
26 });
27
28 }
```

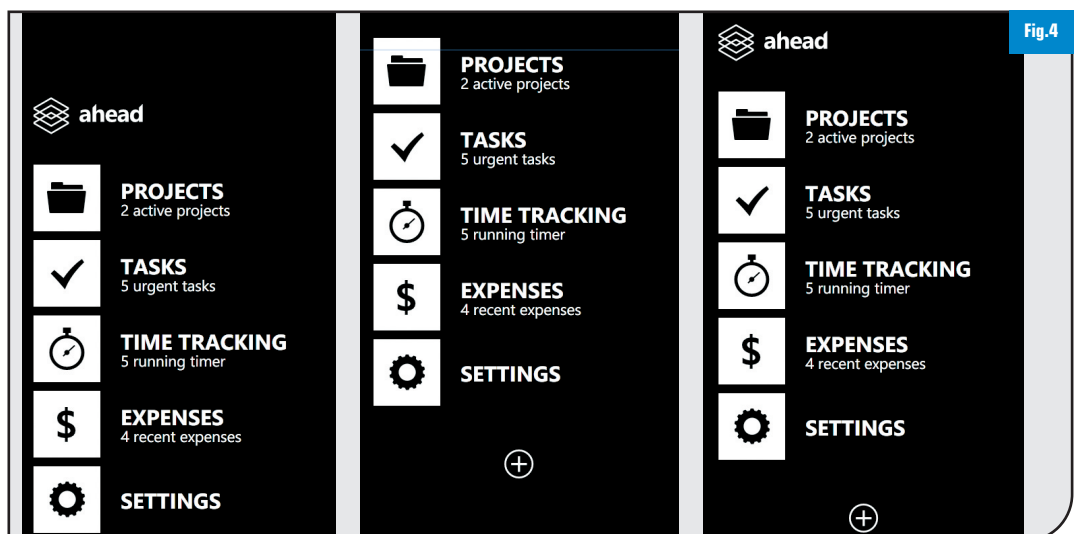
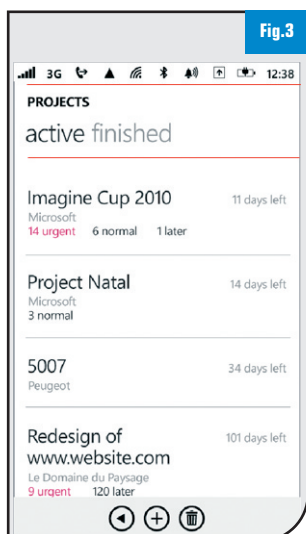
Remonter une grosse liste en une fois

À l'instar des applications iPhone, le développeur WP7 doit penser à ses utilisateurs. Prenons l'exemple d'une longue liste de données, il est désagréable de devoir remonter cette liste à la main. C'est au développeur de proposer soit un bouton pour remonter d'une seule traite tous les éléments, soit comme l'iPhone proposer une zone invisible en haut de l'application. Pour mettre en œuvre cette seconde option, rien de plus simple. Après avoir ajouté un bouton transparent en haut de la fenêtre, il suffit d'appeler la méthode `ScrollIntoView` de la listbox.

```
1 lst.ScrollIntoView(lst.Items.First());
```

L'effet de scrolling

Lorsque le développeur crée un menu (par exemple avec `StackPanel`), il pense souvent à un menu qui affiche des éléments les uns en des-



sous des autres. Faites-le test et donnez le téléphone à un utilisateur lambda. Très souvent, le premier réflexe va être de faire un geste du pouce. Une sorte de scrolling sur le menu pour voir s'il n'y a pas d'élément plus bas. Et là c'est le drame, nous venons d'un seul geste de frustrer l'utilisateur qui se retrouve avec un menu « tout plat ». Le simple fait de positionner ces éléments dans un scrollviewer plus grand que le contenu va lui permettre de faire ce scrolling : [Fig.4]

Une interface fluide

Une application WP7 possède deux threads. L'un qu'on appelle Thread UI piloté par le CPU est responsable de l'exécution de la plupart du code. L'autre s'appelle le thread de composition qui exploite le GPU et permet des opérations telles que le BitmapCaching ou la gestion des animations. Une interface reste fluide tant que le thread UI n'est pas trop occupé. Pour cela l'astuce est de toujours laisser activé le « FrameCounter » de votre WP7 qui vous informe en temps réel de l'occupation des deux threads :

048 048 001632 015 002 00.4113

Le premier (048) représente le Thread de composition et le second (048 aussi) représente le thread UI. Plus la valeur est basse, plus c'est la cata !. L'objectif est de garder le thread de composition autour de 60 fps (à partir de moins de 30 l'expérience utilisateur peut être affectée). Pour le thread UI il faut éviter de descendre en dessous de 15 fps. Il existe de nombreux trucs et astuces pour éviter de bloquer votre IHM, en voilà quelques-uns en vrac :

- Préférez un contrôle WebBrowser au lieu du WebBrowserTask trop long à charger
- Le panorama est plus gourmand que le pivot (non virtualisé)
- Évitez plus de 5 items dans un panorama
- Utilisez des images .jpg au lieu de .png
- Ne vous sentez pas obligé de mettre une splashscreen si votre application se charge vite
- N'utilisez pas de contrôle map/browser dans un panorama (valable pour l'UX aussi)
- Attention aux performances de la progressbar <http://www.jeff.wilcox.name/2010/08/performanceprogressbar/>

Configuration du clavier On-Screen

L'un des pré-requis lors de la manipulation de textbox est de bien cibler le type de clavier à afficher à vos utilisateurs. Rien de plus agaçant que d'arriver sur un clavier non pré-configuré pour l'utilisateur ! Pour cela plusieurs types de clavier sont configurables tels que « Web Address » ou encore « Search ».

```

1 1. <TextBox>
2
3 2. <TextBox.InputScope>
4
5 3. <InputScope>
6
7 4. <InputScopeName NameValue="TelephoneNumber" />
8
9 5. </InputScope>
10
11 6. </TextBox.InputScope>
12
13 7. </TextBox>
```

Utiliser les animations... Mais pas trop !

Les animations c'est bien, mais attention à ne pas tomber dans le **trop**. Silverlight est puissant et il en faut peu pour retomber dix ans en arrière à la découverte du .GIF avec les sapins de Noël clignotants. Petite astuce de développeur, une animation sert souvent à lancer une action un peu longue derrière. Abusez-en quand cela est vraiment utile. Essayez également de rester dans l'esprit des animations standard pour ne pas choquer l'utilisateur.

Utiliser l'effet tilt

L'utilisateur n'aime pas que son IHM soit « plat ». Il aime sentir qu'il appuie sur quelque chose quand il sélectionne un élément d'une liste par exemple. N'oubliez pas donc d'implémenter le Tilt Effect qui a pour but d'enfoncer ce que l'utilisateur est en train de cliquer. Ce n'est qu'un booléen à initialiser en plus. Pour cela, direction le blog de Peter Torr's <http://blogs.msdn.com/b/ptorr/archive/2010/08/11/updated-tilt-effect.aspx>

L'orientation de votre IHM

L'orientation du téléphone peut se faire en activant ou non le mode dans l'entête de notre fenêtre. Pensons à nos utilisateurs en essayant de l'activer là où ceci est possible. Par contre, par défaut, le changement d'orientation est un peu « brut », il ne faut pas hésiter à ajouter une petite animation type :

- Utiliser la propriété de rotation et aller de 0 à -90
- Durée de 500ms
- Easing animation type S-Curve

Le respect du thème utilisateur

L'utilisateur peut choisir un thème sombre ou blanc ainsi qu'une couleur préférée. Que ce soit au niveau du logo de notre application jusqu'au bouton il faut essayer de favoriser au maximum la couleur choisie par l'utilisateur. Il y a parfois sur le market des applications qui se retrouvent avec un logo rouge alors que la *home* de l'utilisateur peut être bleue : rien que pour cela, il ne mettra pas l'application sur sa homepage. Quand nous utilisons un fond noir il faut employer la couleur système représentée par PhoneBackgroundColor. Si on veut une couleur de contraste nous utiliserons PhoneContrastBackgroundColor et pour favoriser sa couleur préférée : PhoneAccentColor. A noter que Blend simule très bien le changement de thème.

Conclusion du développeur

Mes applications sont souvent assez vilaines que ce soit en termes de design ou d'ergonomie. Apprendre les bases de METRO et les appliquer c'est avoir la garantie que nos IHM seront propres et innovantes. J'ai d'ailleurs repris ces principes dans mes applications desktop et web.

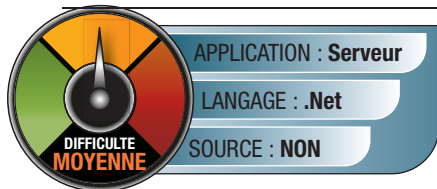
Désormais, quand je crée une application j'essaie de toujours privilégier l'information et retirer les « bruits » tout autour. J'ai un post-it accroché à mon bureau pour me rappeler ce principe de base : *"La perfection n'est pas atteinte quand il n'y a plus rien à ajouter, mais quand il n'y a plus rien à enlever."* — Antoine de Saint-Exupéry.

■ Julien Dollon & Thomas Vidal

Exakis Toulouse – Pole Expérience Utilisateur

IIS Express : le nouveau serveur Web pour le développeur

Le serveur Web pour le développement IIS Express est apparu avec l'outil WebMatrix et son utilisation devrait prochainement se démocratiser chez tous les développeurs Web. Le Service Pack 1 de Visual Studio 2010, propose désormais l'intégration d'IIS Express au sein de l'IDE pour faciliter les développements Web. Bonne nouvelle !



Le SP1 de Visual Studio 2010 apporte son lot de nouveautés et se focalise surtout sur l'amélioration de l'expérience développeur sur des plateformes comme

SharePoint, ASP.NET ou Silverlight. Nous pouvons retenir parmi ces différents ajouts :

- Des outils de développement pour la version 4 de Silverlight.
- La prise en charge des tests unitaires pour la version 3.5 du Framework .NET.
- Le support de l'IntelliTrace pour les projets SharePoint et les architectures 64bit.
- Un outil d'analyse des performances pour Silverlight.
- Une aide locale nommée Help Viewer, accessible hors connexion.
- La mise à jour du compilateur Visual Basic permettant de cibler des plateformes sur lesquelles le runtime VB n'est pas disponible.

Le SP1 corrige également une centaine de bugs concernant le développement Web (notamment sur les éditeurs JavaScript/CSS/HTML, la vue Design, etc.). A cette liste s'ajoute l'intégration native d'IIS Express au sein de Visual Studio 2010. Avant de présenter dans le détail ce nouvel outil, il convient de dresser un état des lieux des possibilités qu'ont à ce jour les développeurs pour exécuter et tester leurs applications Web lors du développement.

IIS EXPRESS OU LA DISPARITION DE CASSINI

Aujourd'hui les développeurs testent leurs applications Web à l'aide d'un des deux serveurs suivants :

- Le serveur de développement ASP.NET Cassini.
- Le serveur Web IIS.

Ces deux serveurs ont leurs avantages et leurs inconvénients. Cassini, intégré à Visual Studio, a l'avantage d'être léger et facile d'utilisation. Cependant il possède certaines limitations :

- Ne peut héberger qu'une seule application ASP.NET par port.
- Ne prend pas en charge HTTPS ni l'authentification.
- Répond uniquement aux requêtes localhost.

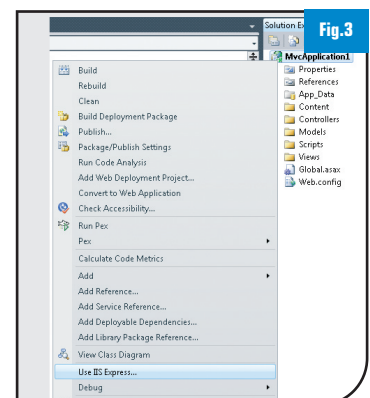
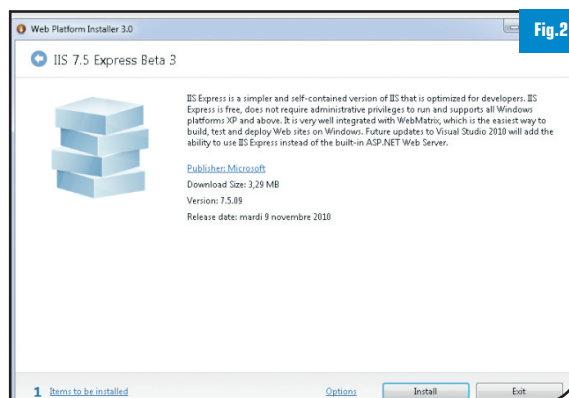
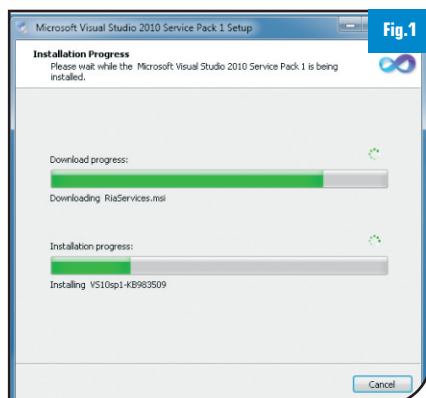
En plus d'être pauvre en fonctionnalités, le principal inconvénient de Cassini est de ne pas représenter l'application telle qu'elle sera une fois hébergée sur le serveur Web de production.

Utiliser IIS comme serveur de développement permet d'utiliser toutes les fonctionnalités d'un véritable serveur Web et tous les modules qui viennent avec. Néanmoins IIS n'est pas conçu pour les développements et impose certaines contraintes comme la nécessité d'avoir un compte administrateur pour accéder à la configuration et au débogage des applications. De même, son installation dépend fortement de la version de l'OS du poste de développement.

Scott Guthrie résume très bien la situation actuelle : « *De nombreux développeurs Web souhaiteraient avoir la facilité d'utilisation du serveur Cassini, mais toute la puissance et les fonctionnalités de IIS. C'est à cette demande que répond IIS Express, qui combine les meilleurs caractéristiques des deux options et facilitera l'exécution et le débogage des applications Web* ».

UTILISER IIS EXPRESS POUR VOS DÉVELOPPEMENTS

Les développeurs ASP.NET apprécieront sans aucun doute le fait de pouvoir travailler sur IIS de façon bien plus simple que précédemment. IIS Express sera compatible avec Visual Studio 2010 et Visual Web Developer 2010 Express. Un patch pour l'intégration dans Visual Studio 2008 n'étant pas à l'ordre du jour, son utilisation avec cet IDE est plus compliquée. Toutefois il est possible de lancer un



site en ligne de commande. La création d'un bath DOS ou d'une commande Visual Studio est aisée. IIS Express, comme Cassini, ne nécessite aucune étape de configuration ce qui le rend très facile à lancer pour exécuter vos développements. Pour résumer :

- Son téléchargement et son installation est rapide (un peu plus de 3Mo à télécharger).
- Il ne nécessite pas de compte administrateur pour exécuter ou déboguer une application depuis Visual Studio.
- Il offre les fonctionnalités d'un serveur Web complet, incluant le support SSL, la réécriture d'URL et les autres extensions et modules des versions 7.x d'IIS.
- Il offre le même modèle d'extensibilité et supporte le fichier de configuration Web.config comme IIS 7.x
- Il peut être installé conjointement avec un serveur IIS local comme le fait Cassini sans aucun conflit.
- Il fonctionne sous Windows XP et versions supérieures, permettant ainsi l'utilisation des fonctionnalités d'IIS 7.x sur tous les OS Windows, ce qui n'était pas possible précédemment.
- Il supporte les modes Classic et Integrated du Pipeline ainsi que l'authentification basique ou Windows.

INSTALLATION

Avant toute chose, il vous faut télécharger le Service Pack 1 de Visual Studio 2010 à cette adresse : <http://go.microsoft.com/fwlink/?LinkId=207130>. Pour rappel, il s'agit d'une version bêta, pensez donc à utiliser une machine virtuelle pour vos tests en attendant la release. L'installation est relativement longue avec environ 500 Mo à télécharger via l'outil d'installation [Fig.1]. Une fois fait, il faut maintenant installer IIS Express car il n'est pas inclus dans le Service Pack. IIS Express est actuellement en version 7.5 bêta et ne fait pas partie intégrante de WebMatrix. Vous pouvez donc installer IIS Express sans WebMatrix à l'aide du Web Platform Installer de Microsoft via cette adresse : <http://www.microsoft.com/web/gallery/install.aspx?appid=iisexpress>. Le téléchargement nécessaire est d'un peu plus de 3Mo et l'installation très rapide [Fig.2].

UTILISATION

Pour utiliser IIS Express, rien de plus facile : un clic-droit sur le projet et on sélectionne "Use IIS Express" [Fig.3]. Puis un simple appui sur la touche F5 lance l'application avec IIS Express. Il n'y a pas de configuration à faire comme c'est le cas pour Cassini. Visual Studio 2010 nous informe alors que le paramétrage a bien été pris en compte et que notre projet Web sera hébergé sur localhost [Fig.4].

Tout comme Cassini, pendant son exécution, IIS Express fait apparaître une icône dans la zone de notification de la barre de tâches [Fig.5]. On peut ainsi voir toutes les applications actuellement héber-

gées par le serveur de développement. Les fonctionnalités qu'offre cet outil sont les suivantes :

- **Sites** : Affiche la liste des sites en cours d'exécution. Sélectionnez un site spécifique pour afficher les options « Browse Applications » et « Stop Site ».
- **Exit** : Ferme tous les sites en cours d'exécution.
- **Browse Applications** : affiche l'adresse URL de chaque application. Un simple clic permet d'ouvrir celles-ci dans le navigateur.
- **Stop Site** : arrête le site sélectionné. Cette action supprime également le site de la liste des sites de la zone de notification
- **Show All Applications** : Affiche une fenêtre de dialogue avec la liste des applications en cours d'exécution, incluant l'identifiant du processus (PID), le nom du site et l'URL [Fig.6]. Sélectionner un site permet d'afficher plus de détails comme la version du Framework .NET, le path et le parent. Ce dernier étant l'application qui est utilisée pour lancer l'instance d'IIS Express qui héberge le site.
- **Stop All** : stoppe toutes les applications en cours d'exécution qui sont listées et ferme la zone de notification.

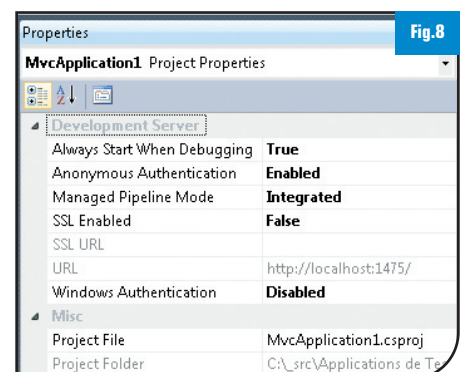
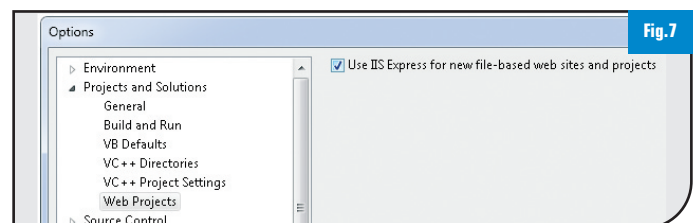
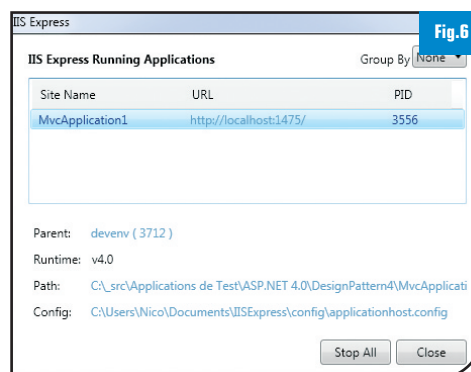
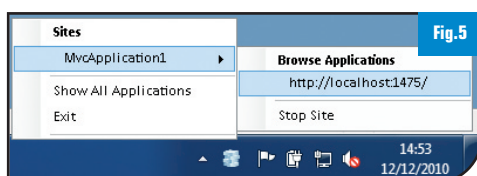
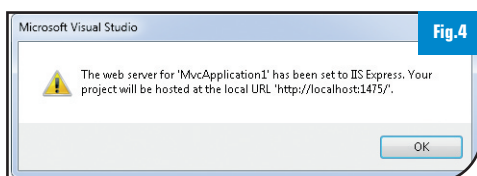
Pour utiliser IIS Express comme serveur Web de développement par défaut à la place de Cassini, rien de plus simple. Il suffit d'aller dans le menu Tools, puis Options, puis Projects and Solutions, puis Web Projects [Fig.7]. L'utilisation de SSL est également très simple avec ce serveur Web. L'écran propriétés de notre projet Web (presser F4 une fois le projet sélectionné) offre plusieurs options. Il permet d'activer ou non SSL, de choisir le mode d'authentification (Windows ou basique) et d'activer l'authentification anonyme. Il est également possible de choisir le mode du pipeline [Fig.8].

L'arrivée d'IIS Express est donc comme un rayon de soleil chez les développeurs ASP.NET (que ce soit Web Forms ou MVC). Ce nouveau serveur Web permet maintenant d'utiliser toute la puissance d'IIS 7.x, ses modules et ses extensions, pendant les développements tout en gardant l'extrême simplicité de Cassini. Pour les entreprises cela représente également une économie puisque les sites seront exécutés dans un environnement proche de celui de production, permettant ainsi de détecter les problèmes à l'avance.

■ Nicolas Esprit

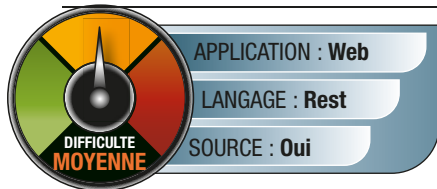
Consultant / Chef de projet - ICT7 Luxembourg

<http://www.nicolasesprit.com>



S'inspirer d'Atom et de Google pour créer des services Web Restful

Les solutions de WebServices basées sur les principes REST ont le vent en poupe. Le cloud, la crise, le Web 2.0, un bon livre, les plateformes mobiles, ont remis au goût du jour cette approche qui n'est pourtant pas récente. Les principes REST ont été formulés il y a maintenant plus de 10 ans par Roy T. Fielding.



REST a été plutôt mal compris pendant des années. Ce n'est pas un produit, ce n'est pas une spécification. Les débats sur le sujet ont pu ressembler à des bagarres

de villages gaulois. Même l'utilisation du mot REST est sujette à polémique. C'est une approche, d'un style de design des systèmes distribués, inspiré des principes du Web. Ces principes peuvent être utilisés pour créer de nouveaux types de services sur ce même Web : les services Web Restful.

ATOM, GOOGLE ET NOUS

Atom et Google

Atom est un bon exemple de mise en œuvre des principes REST. Ce n'est pas qu'un format pour la syndication de contenus périodiques tels que les blogs ; c'est aussi un protocole, dans le sens où les interactions entre le client et le serveur sont spécifiées.

Google, comme d'autres, s'est approprié REST et Atom, pour exposer des services Web dans des domaines très variés, tels que la vidéo (youtube), la finance, la cartographie, le médical, les documents (google Spreadsheets).

Nous

Alors pourquoi ne pourrions-nous pas nous inspirer de Google pour construire nos propres services web et ainsi, profiter de leurs propres efforts d'ingénierie. La question est de savoir dans quelle mesure notre situation est similaire à celle de Google.

Envisageons les hypothèses suivantes :

- **Première hypothèse** « *Google se trompe, et on se trompe également en faisant pareil* ». Si Google se trompe, ainsi que tous ceux qui ont choisi le format standard Atom ; ou si une meilleure approche, un meilleur format vient à apparaître, on peut espérer qu'une approche de migration soit proposée par Google ou des communautés. Le coût de cette erreur sera moindre que toutes celles que l'on fera en mettant en place une solution spécifique. De la même manière on trouvera des discussions sur le net à propos des choix de Google, ce qui permet de faire avancer encore plus le sujet.
- **Deuxième hypothèse**, « *Google a raison, mais nous nous trompons en l'imitant* ». La question fondamentale du domaine de validité d'une solution est posée. Google utilise avantageusement REST et Atom dans ses systèmes, où les consommateurs finaux de l'information sont des humains. Google n'utilise pas Atom (et pas REST à priori) pour resynchroniser ses Datacenters. Autre

exemple, des services suivant tous les principes REST peuvent être de la sur-ingénierie dans un projet qui a pour unique ambition d'alimenter un unique client développé par la même équipe développant le serveur.

- **Troisième hypothèse**, « *Google se trompe, mais on a raison en faisant comme eux* ». Il s'agit d'une variante peu probable du domaine de validité d'une solution illustrée au cas précédent. Cette hypothèse est difficilement défendable, et ne mérite pas d'être développée ;)).
- **Quatrième et dernière hypothèse**, « *Google a raison et on a raison* ». On a donc vérifié que notre domaine d'application est proche de celui de Google. On profite des efforts de Google et de tous ceux qui sont allés dans cette direction.

LES GOOGLE WEB API

Google a bien documenté ses API et propose même une page Web [1] pour « jouer » avec son protocole. Un formulaire permet de sélectionner l'opération (query, insert, update, delete) et de définir l'url de la requête. En validant le formulaire, la requête s'exécute, et les données http de la requête et de la réponse sont affichées.

Des interactions standard sont largement documentées [2] et [3].

Le contrat uniforme via les URI, les headers et les status

La description d'une API Web REST englobe :

- les URI (décrivent le sujet adressé et les états de l'application)
- les verbes (GET, PUT, DELETE, POST, ...) qui décrivent les actions sur ces sujets
- les représentations échangées (qui sont les données nécessaires structurées selon Atom ou autre)
- les headers échangés (utilisés principalement par les composants d'intermédiation comme les proxies et les négociations standard entre client et serveur)
- Les status et les codes d'erreur (ok, created, not found, not modified, etc...)

Dans une implémentation REST, les comportements sont uniformisés. Cela garantit des comportements standard, et favorise la stabilité de l'API, ainsi qu'une action efficace et transparente des composants d'intermédiation.

Les header http standard sont pleinement utilisés pour permettre d'optimiser les échanges (compression gzip, conditional get via last-

[1] <http://www.google.com/base/api/demo/html/demo.html>

[2] <http://code.google.com/apis/gdata/docs/2.0/basics.html>

[3] <http://www.google.com/base/api/demo/html/demo.html>

modified, cache-control,...), d'assurer la sécurité et la négociation des contenus échangés.

Des extensions sont introduites, comme http le permet (X-Google-Key, X-Frame-Options, X-HTTP-Method-Override, etc.) pour répondre à de nouvelles problématiques. Ces extensions sont destinées à être manipulées par des composants d'intermédiation, et non par du code spécifique côté serveur ou clients. L'impact est donc minimal. On notera également le header spécifique G-DataVersion, qui définit la version de la méthode de l'API.

La forme des URI est également normalisée. La normalisation va au-delà de la normalisation de l'adressage des ressources, en imposant un jeu de paramètres standard à toutes les API. Google spécifie une syntaxe de langage de requêtes [4], une pour la pagination, et d'autres pour des comportements plus avancés comme le « partial reading [5] ».

Implémenter l'ensemble du langage de requête de Google est certainement un exercice délicat, mais rien n'empêche de rester compa-

tible en implémentant un sous-ensemble suffisant. Il faut mesurer l'investissement en considérant si l'on peut capitaliser sur de nombreuses recherches. Peut-être même existe-t-il un analyseur syntaxique publié réutilisable.

Le Feed et l'Entry

Les deux documents échangés sont le *feed* Atom (qui décrit essentiellement les collections et les résultats de recherche) et l'*entry* Atom (qui décrit un élément dans un feed, ou un élément détaillé accédé tout seul, ou retourné au serveur pour mise à jour).

Toutes les API sont basées sur ce même modèle, et sont éventuellement étendues avec des balises spécifiques.

Surmonter la tentation du spécifique permet d'assurer la stabilité de l'API. Utiliser une spécification publique préserve de cette tentation, ou du moins met en exergue tout écart. La quantification de la stabilité de l'API devient possible, en définissant le ratio d'éléments spécifiques sur le nombre d'éléments génériques utilisés. Et quand on peut mesurer, on peut commencer à décider (ne serait-ce qu'en comparant le ratio de « stabilité » de nos API avec celles de Google). A noter que Google introduit une opération standard de « batch », qui permet de demander N opérations en une seule requête au serveur. Celle-ci est décrite en respectant la structure de Feed, dont chaque « Entry » représente une opération sur une ressource. Le sens REST puriste de cette implémentation peut être discuté, mais en faisant comme Google, si on n'a pas de meilleure approche, on se met dans une situation connue.

Détail d'un feed Google Spreadsheets

L'API Google Spreadsheets permet de consulter et de mettre à jour des Google Spreadsheets. Ci-dessous l'affichage d'une sortie Atom représentant un Spreadsheet. Les différents éléments intéressants sont numérotés et présentés dans le tableau ci-contre.

Au final, un seul élément (12) est spécifique à Google Spreadsheets. Les autres sont issus d'Atom, d'autres standards (Opensearch) ou de Google, mais sont partagés avec l'ensemble des API Google.

Élément	Observation
(1) gd:etag	Google introduit, ou du moins étend l'usage de l'etag (qui est normalement utilisé comme header http pour certaines stratégies de cache). Cela permet de sécuriser les mises à jour concurrentes.
(2) id	Balise d'Atom, représentant l'ID de la ressource. A noter que le contenu représente l'URL et non un ID technique. La question du rôle d'un Id en plus de l'URI est posée. Celui-ci peut avoir son sens pour communiquer, tracer des informations, via d'autres moyens que la navigation web.
(3) updated	Balise Atom, pour la date de mise à jour.
(4) title	Balise Atom, utilisée comme titre ou comme explicatif du contenu. Pratique pour mettre en avant des avertissements.
(5) link	Quelques exemples de liens. Les liens permettent de naviguer entre les ressources. C'est un des points importants trop souvent oubliés. Ce sujet souvent abordé sous l'intitulé « Connexité » dans les API REST. La balise link permet de bien nommer chaque lien. Il existe des usages courants, repris par Atom et Google (page suivante, précédente, représentation alternative, etc.), mais il ne faut pas hésiter à introduire les siens.
(6) author	L'auteur
(7) opensearch:totalResults opensearch:startIndex opensearch:itemPerPage	Balises issues d'Opensearch. Elles décrivent de manière standard la pagination.
(8) entry gd:etag	Une balise entry et un attribut etag. L'etag permet de gérer la concurrence de mise à jour sur un entry uniquement. A noter que l'entry ne renseigne pas sur le type de son contenu.
(9) id	L'id de l'entry, qui est également l'URI de la ressource.
(10) updated	Date de mise à jour
(11) category	Élément permettant de spécifier le type de l'entry. En retardant le type dans la structure XML, sans le spécifier comme une balise, cela permet de gérer n'importe quel type d'entry et des feeds contenant des données très variées.
(12) gsx:name gsx:hours gsx:item gsx:ipm	Descrit les données de la ligne (row). Ce sont des champs spécifiques à Spreadsheet. On remarquera que les noms de balise ne sont pas prédéfinis (il s'agit du nom des colonnes). Les données de ces champs sont également présentes dans le champ « content ». Le champ content est destiné à un affichage texte ; ici les données ont vocation à être traitées élémentairement.
(13) content	Le contenu de l'entry, discuté au numéro 12, exprimé en texte pour affichage brut textuel.
(14) link	Quelques liens propres à l'entry

```
<feed xmlns="http://www.w3.org/2005/Atom"
  xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/"
  xmlns:gsx="http://schemas.google.com/spreadsheets/2006/extended"
  xmlns:gd="http://schemas.google.com/g/2005"
  gd:etag='W/"D0cERnk-eip7ImA9WBBXGEg."'> (1)
  <id>https://spreadsheets.google.com/feeds/list/key/worksheetId/private/full</id> (2)
  <updated>2006-11-17T18:23:45.173Z</updated> (3)
  <title type="text">Sheet1</title> (4)
  <content type="application/atom+xml;type=feed" src="https://spreadsheets.google.com/feeds/list/key/worksheetId/private/full"/>
  <link rel="alternate" type="text/html"
    href="https://spreadsheets.google.com/ccc?key=key"/> (5)
  ...
  <link rel="self" type="application/atom+xml"
    href="https://spreadsheets.google.com/feeds/list/key/worksheetId/private/full"/> (5)
  <author> (6)
    <name>Fitzwilliam Darcy</name>
    <email>fitz@gmail.com</email>
```

[4] <http://code.google.com/apis/base/docs/2.0/query-lang-spec.html>

[5] <http://code.google.com/apis/gdata/docs/2.0/basics.html#PartialResponse>

```

</author>
<openSearch:totalResults>8</openSearch:totalResults> (7)
<openSearch:startIndex>1</openSearch:startIndex>
<openSearch:itemsPerPage>8</openSearch:itemsPerPage>
<entry gd:etag=' "S0wCTlpIIip7ImA0X0QI" ' > (8)
  <id>https://spreadsheets.google.com/feeds/list/key/worksheetId/private/full/rowId</id> (9)
  <updated>2006-11-17T18:23:45.173Z</updated> (10)
  <category scheme="http://schemas.google.com/spreadsheets/2006"
    term="http://schemas.google.com/spreadsheets/2006#list"/> (11)
  <title type="text">Bingley</title>
  <content type="text">Hours: 10, Items: 2, IPM: 0.0033</content> (13)
  <link rel="self" type="application/atom+xml"
    href="https://spreadsheets.google.com/feeds/list/key/work
sheetId/private/full/rowId"/> (14)
  <link rel="edit" type="application/atom+xml"
    href="https://spreadsheets.google.com/feeds/list/key/work
sheetId/private/full/rowId/version"/>
  <gsx:name>Bingley</gsx:name> (12)
  <gsx:hours>10</gsx:hours>
  <gsx:items>2</gsx:items>
  <gsx:ipm>0.0033</gsx:ipm>
</entry>
...
<entry gd:etag=' "AxQDSXxjfy7ImA0ChJVSBI." ' >
...

</entry>
</feed>

```

Feed Atom représentant un Spreadsheet (6)

Le JSON atomisé comme structure alternative à l'Atom.

Google propose aussi un format JSON respectant la même structure qu'Atom. Le JSON reste donc dérivé par l'Atom, ce qui en assure la stabilité. Une seule moulinette pourra donc produire du JSON ou de l'Atom. Peut être même Google a-t-il défini une infrastructure d'intermédiation (des proxies particuliers), interceptant les contenus « abstraits » (pas encore de l'Atom, pas encore du JSON) et les remplaçant par des contenus JSON, ou Atom selon le cas.

EN CONCLUSION

L'expérience Google montre qu'il est possible de proposer des API sur des sujets qui n'ont rien à voir à partir du standard Atom. Cela offre l'avantage de la stabilité et de la mise en commun de solutions, et donc des économies d'infrastructures et de code. Il faut savoir résister à la tentation du spécifique. Sans copier bêtement Google, savoir pourquoi on fait différent à un moment donné, traduit une maturité qui ne peut que bénéficier aux projets de services Web.

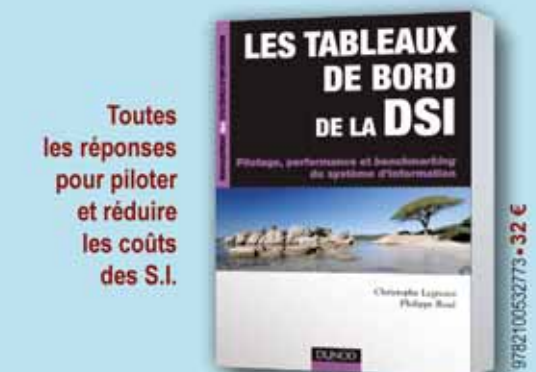
Google vient de signifier que ses API Web seront dépréciées (7) dans quelques mois. Atom reste au cœur de la future API. Les principes énoncés précédemment demeurent applicables. Reste à voir si les futures orientations sont à considérer comme à inclure ou à exclure de nos prochains domaines d'application et d'inspiration.



■ **Thierry Sorgnard**
Consultant Architecte Senior, Valtech.

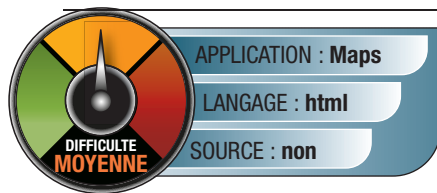
(6) http://code.google.com/apis/spreadsheets/data/3.0/developers_guide.html#ListingSpreadsheets

(7) http://code.google.com/apis/base/faq_depr.html



Créer une carte avec des marqueurs grâce à Google Maps API version 2 et 3 et Dojo Toolkit

Vous êtes dans une phase où le site de votre entreprise commence à prendre une certaine envergure : vous décidez alors de l'agrémenter en rajoutant une carte pour, par exemple, indiquer aux utilisateurs où se trouvent vos filiales, bâtiments annexes ou même vos points de ventes. C'est alors que Google Maps intervient, puisqu'il permet d'intégrer des cartes interactives à vos pages avec un lot d'options et de contrôles divers (pour changer le type de carte par exemple), tout en fournissant un ensemble de librairies plus que complètes, écrites en JavaScript.



Etant donné que l'API version 2 est considérée comme dépréciée (« deprecated ») depuis mai 2010, - c'est-à-dire qu'il est fortement déconseillé de l'utiliser

pour des questions de sécurité ou même de performances, - nous verrons donc les différences essentielles entre la version la plus récente et celle utilisée dans cet article, avec ou non des exemples de code. Notez aussi que même si elle n'est plus d'actualité, vous pouvez envisager de l'utiliser à des fins de tests, en parallèle avec les explications fournies. Enfin, depuis la mise en service de la troisième version, il n'est désormais plus nécessaire de générer une « clé d'API » en s'inscrivant, c'est-à-dire une clé d'authentification unique à votre nom de domaine/adresse IP, qui permettait en fait d'accéder aux différents services proposés par Google, dont celui-ci. Ainsi, il est plus simple de passer vos pages en pré-production ou en production, si vous développez en local ou sur un nom de domaine indépendant de votre site internet. Mais terminons cette introduction théorique et passons maintenant à la pratique.

CRÉER LA PAGE ET L'ÉLÉMENT CONTENEUR POUR LA CARTE

Comme beaucoup d'objets en JavaScript, il est nécessaire de créer un conteneur, qui dans notre cas sera une balise de type « div » avec un identifiant que l'on peut choisir (qui sera réutilisé plus tard dans notre exemple). Mais au préalable, il faut créer la page web dans laquelle sera positionnée notre balise. La page de démonstration qui suit est écrite en HTML5 :

```
<!DOCTYPE html>
<html><head><!--META TAGS --></head>
<body>... </body></html>
```

La syntaxe du type de document (déclaré en ligne 1) a été fortement simplifiée pour cette version du langage HTML. Dès à présent, dans le corps de notre page, nous allons créer l'élément conteneur pour notre future carte de la manière suivante, tout en lui définissant son identifiant (« id ») ainsi qu'une feuille de style grâce à l'attribut adéquat, dans laquelle on indique une largeur et une hauteur de base :

```
<div id="map" style="width: 700px; height: 480px;">
```

CHARGEMENT DES RESSOURCES NÉCESSAIRES AU FONCTIONNEMENT DE GOOGLE MAPS

Lorsque votre structure de page est en place, vous devez charger les différentes ressources nécessaires pour que votre première carte puisse apparaître dans la zone définie précédemment. Dans la version 2, il existe deux manières différentes de procéder pour déclarer le script de chargement des bibliothèques. La première méthode consiste à charger dynamiquement les bibliothèques en utilisant la technologie « AJAX » (Asynchronous JavaScript And XML). Quant à la seconde version, celle-ci a un comportement classique et récupère le tout de manière statique. Dans les deux cas, le bout de code doit être placé dans l'en-tête de la page, donc entre les éléments ouvrant et fermant « head ». Premier morceau de code :

```
<head><script type="text/javascript" src="http://www.google.com/jsapi?key=API_KEY"></script></head>
```

Second morceau de code :

```
<head><script type="text/javascript" src="http://maps.google.com/maps?file=api&v=2&key=API_KEY&sensor=false"></script></head>
```

Lorsque le paramètre détecteur (ou « sensor ») est mis à vrai (ou « true » en anglais), Google pourra déterminer la position géographique approximative de l'utilisateur pour centrer la carte sur celle-ci par défaut.

Dans Google Maps API version 3, on spécifiera d'abord un élément « meta » (métadonnées) qui spécifie l'échelle initiale de la carte (par exemple, 1.0 pour 100%) et si l'utilisateur a le droit de redimensionner (paramètre « user-scalable » prenant la valeur « yes » ou « no »). Ensuite, on place le même script que dans le second morceau de code ci-dessus, à la différence près qu'il ne contiendra pas le paramètre « key » où l'on spécifiait la clé d'API. A noter également que l'adresse source a légèrement été remaniée par rapport à l'ancienne, et que le détecteur (sensor) peut toujours être utilisé :

```
<head><meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
<script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=set_to_true_or_false"></script></head>
```

Remarque : un paramètre « libraries » peut être rajouté dans l'URL source du script pour charger des bibliothèques supplémentaires,

comme « geometry », qui offre des fonctions de calcul sur des valeurs géométriques scalaires, ou « adsense » qui vous permettra d'inclure des publicités textuelles qui varient selon le contexte. On peut alors instancier des objets des classes du même nom.

INTÉGRER, INITIALISER ET AFFICHER LA CARTE DANS LE CONTENEUR

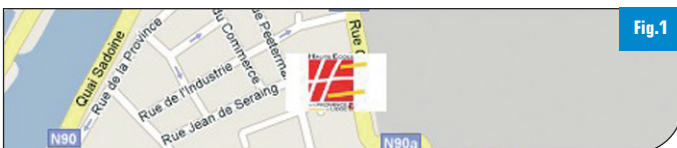
Afin d'initialiser la carte, nous allons tout d'abord créer une fonction dans un élément « script », qu'on appellera « initMap ». Celle-ci va s'occuper de charger la carte et de centrer la position à une latitude et une longitude qu'on précise. Dans notre cas, nous centrerons la position sur la « Haute-Ecole Rennequin Sualem », dont la latitude est 50.6113123 et la longitude 5.5104172. Tout d'abord il faut instancier l'objet « carte » en spécifiant dans quel conteneur celle-ci va apparaître, ensuite on utilise la fonction « setCenter » sur la variable créée, qui prend en paramètre une instance d'objet « GLatLng » qui reprend concrètement la position exprimée en latitude-longitude. Le deuxième paramètre de la fonction « setCenter » est le zoom pour l'affichage (au minimum 0 pour avoir un aperçu du monde entier jusqu'à des valeurs supérieures à 21 pour le type par défaut) :

```
function initMap() {
    var map = new google.maps.Map2(document.getElementById('map'));
    map.setCenter(new GLatLng(50.6113123, 5.5104172), 15);
}
```

Si le chargement a été effectué dynamiquement, on doit obligatoirement appeler une fonction avant l'initialisation de notre carte, qui s'occupera de charger les bibliothèques de Google Maps ainsi que la fonction « initMap » en tant que fonction « callback ». Cela signifie que cette dernière sera chargée lorsque toutes les ressources seront prêtes à l'emploi. Enfin dans la fonction « google.load », on spécifie le nom de l'API à charger, ensuite en second paramètre la version de celle-ci.

```
<script type="text/javascript"> /* AJAX Loader */
    google.load("maps", "2");
    google.setOnLoadCallback(initMap);
</script>
```

La première carte est enfin prête ; elle devrait afficher le résultat suivant (l'image affiche un marqueur avec une icône personnalisée, mais en temps normal c'est un marqueur rouge que l'on peut apercevoir) : [Fig.1]



Pour la version actuelle, on continuera à utiliser les objets « Latlng » et « Map » mais les classes ont été renommées pour s'adapter aux futures mises à jour de Google. Dans le code d'initialisation de la carte, on peut aussi spécifier des options telles que le zoom, la position initiale ou le type de carte. Les options sont séparées par des virgules et stockées dans une simple variable. Quant au type de carte, on peut par exemple utiliser « ROADMAP » (qui est l'affichage normal en 2D), « SATELLITE », « HYBRID » (affiche les contrôles

pour passer du premier au second type) ou « TERRAIN ». Contrairement à Google Maps version 2, il est obligatoire de spécifier un type de base lors de l'initialisation, puisqu'il n'en existe aucun par défaut. Voici un exemple d'adaptation de la fonction « initMap » :

```
var myLatlng = new google.maps.LatLng(50.6113123, 5.5104172);
var myOptions = { zoom: 8, center: myLatlng, mapTypeId: google.
    maps.MapTypeId.ROADMAP };
var map = new google.maps.Map(document.getElementById("map"),
    myOptions);
```

CRÉER ET PLACER DES MARQUEURS (MARKERS)

La création d'un marqueur, ou balise, s'avère être bien plus simple depuis l'arrivée de la version 3. En effet, les différentes options du marqueur sont maintenant définies directement dans l'instanciation de l'objet « Marker », et il n'est plus nécessaire de déclarer de variable reprenant l'ensemble des options. Ainsi, on écrira le code suivant pour créer un marqueur. On doit spécifier la carte sur laquelle on souhaite voir apparaître le marqueur, ainsi que la position, qui n'est autre qu'un objet « Latlng », et enfin un titre qui s'affichera dans une bulle d'aide lorsque l'on survolera avec le pointeur de la souris :

```
var marker = new google.maps.Marker({ position: myLatlng, map:
    map, title:"Hello World!" });
```

L'idéal serait d'agrémenter cela en ajoutant une fenêtre de type « popup » lorsque l'on clique sur la carte, qui permettrait d'ajouter un marqueur sur la carte. Nous utiliserons pour ce faire, le framework Dojo Toolkit, qui est un projet Open-Source développé par « The Dojo Foundation », et qui est concrètement un ensemble de bibliothèques JavaScript offrant diverses fonctionnalités pour enrichir et dynamiser le contenu de vos pages web (fenêtres pop-up, calendriers, listes déroulantes au contenu dynamique, onglets, etc). Comme tout projet libre, en plus de posséder une vaste communauté, on peut aussi y contribuer. Pour la parenthèse, on retrouve dans les noms de donateurs des grandes entreprises comme Google, AOL, IBM, et bien d'autres. Le projet a gravi des échelons, désormais disponible en version 1.5, et se voit compatible avec tous les navigateurs récents comme Opera, Safari et Mozilla Firefox. Une riche documentation est également disponible sur le site officiel du projet.

UTILISATION BASIQUE D'UNE FENÊTRE DE TYPE « POPUP » GRÂCE À DOJO TOOLKIT

Au préalable vous devrez obtenir une version du framework sur le site officiel de Dojo Toolkit (<http://dojotoolkit.org/>) qu'il vous faudra décompresser dans un répertoire de travail auquel vous pouvez accéder depuis une page de votre site web. Le fichier archivé disponible est structuré en trois parties distinctes : les composants « dojo » (éléments de base), « dijit » (formulaire et éditeurs) et « dojox » (pour Dojo eXtended). Il est recommandé de conserver cette structure lorsque vous décompressez les fichiers. Une fois que cela est fait, vous pouvez initialiser le script dans l'élément « head » de la page. Attention, les démonstrations se basent sur la version 1.2.

```
<script src="/VOTRE_CHEMIN/dojo/dojo.js" type="text/javascript">
</script>
```

Si télécharger le kit complet vous dérange, vous pouvez directement utiliser le script suivant, qui ne vous oblige pas à vous munir d'une version complète puisqu'on fait appel à un fichier distant. Le morceau de code est généreusement fourni par Google.

```
<script src="http://ajax.googleapis.com/ajax/libs/dojo/1.5/dojo/dojo.xd.js" type="text/javascript"></script>
```

Pour notre exemple, nous allons intégrer à la fenêtre popup une liste déroulante et un bouton pour confirmer l'ajout du marqueur sur la carte. Nous allons donc avoir besoin de l'élément « dialog », « FilteringSelect » et « Button ». Il existe une fonction dans Dojo pour charger ce dont on a réellement besoin : « require ». Il suffit de préciser l'objet que l'on souhaite utiliser (le nom complet de la classe). L'appel se fait bien sûr en JavaScript, donc nous avons besoin d'une balise « script ».

```
<script type="text/javascript">
  dojo.require("dijit.form.Button"); dojo.require("dijit.form.FilteringSelect");
  dojo.require("dijit.Dialog");
</script>
```

Ensuite, nous devons créer l'élément conteneur pour la boîte de dialogue, ainsi que pour les boutons « OK ». On créera aussi une liste déroulante. Chacun possède un attribut « dojoType » afin de préciser à quel genre de champ nous allons avoir à faire et ainsi pour permettre l'accès à des fonctions ou des attributs spécifiques à ce type de champ. Lorsque l'on souhaite accéder à un élément de notre page web, à partir du moment où celui-ci est identifié (attribut « id »), on peut le faire grâce à la fonction « dijit.byId(ID_CHAMP) ». C'est l'équivalent de la fonction bien connue « document.getElementById() ».

```
<div dojoType="dijit.Dialog" id="dialogmarker" title="Add a new marker for the institution"
  execute="if(dijit.byId('building').value) addGpsLocationToDatabase();">
  <select dojoType="dijit.form.FilteringSelect" id="building"
    name="buildingfilteringselect" autoComplete="true"
    invalidMessage="Nom du bâtiment incorrect.">
    <option value="0">Bâtiment principal</option><option value="1">Annexe</option>
  </select>
  <div dojoType="dijit.form.Button" type="submit">OK</div>
</div>
```

Telle que définie ci-dessus, la boîte de dialogue sera une sorte de formulaire. On y a créé un bouton OK dont le type est « submit » (le clic sur le bouton va générer l'envoi du formulaire), un bouton « Annuler » qui cache simplement la boîte de dialogue, ainsi qu'une liste déroulante proposant les options « Bâtiment principal » ou « Annexe ». Lorsque l'on saisit une valeur incorrecte dans celle-ci, elle affiche un message d'erreur (attribut « invalidMessage »). Enfin, on spécifie le code JavaScript qui sera appelé, dans l'attribut « execute » du conteneur de la boîte de dialogue. Dans notre cas, on appelle la fonction « addGpsLocationToDatabase() » si la valeur de la liste déroulante n'est pas vide.

Lors de l'initialisation de la carte, on doit prévoir l'événement qui sera appelé lors du clic sur la carte. La création d'événement en

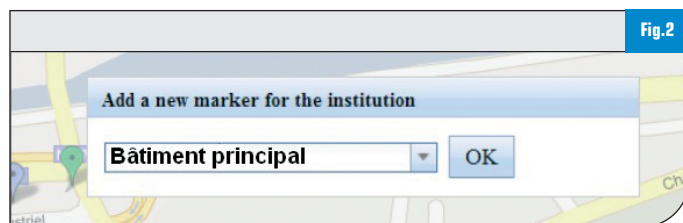
version 3 est similaire à celle de la version précédente. On n'utilise plus la classe « GEvent » mais « google.maps.event ». Voici un exemple de ce que nous pourrions écrire comme début d'événement « sur clic » :

```
GEvent.addListener(map,"click", function(overlay,latlng) { ... });
```

Nous l'écrirons plutôt de la manière suivante :

```
google.maps.event.addListener(marker, 'click', function(overlay, latlng) {
  if (latlng) {
    tempLatitude = latlng.y; tempLongitude = latlng.x;
    dijit.byId('dialogmarker')._onKey = function(){}; dijit.byId('dialogmarker').show();
  });
});
```

Lorsque l'on cliquera sur la carte, cela ouvrira la boîte de dialogue précédemment créée. Les deux variables « tempLatitude » et « tempLongitude » sont des variables globales qui seront réutilisées pour ajouter les marqueurs. Enfin, lorsque l'on valide la boîte de dialogue, on appelle donc la fonction « addGpsLocationToDatabase ». Sans rentrer dans les détails, celle-ci va tout d'abord ajouter les coordonnées « tempLatitude » et « tempLongitude » à la base de données, afin qu'elles puissent être réutilisées pour un affichage futur, ensuite on ajoute simplement les marqueurs sur la carte avec la méthode vue précédemment. Puisque l'on a utilisé une liste déroulante permettant de choisir le type de bâtiment à ajouter en annexe, il est possible avant l'ajout, de modifier l'icône par défaut du marqueur que l'on ajoute. La méthode est détaillée dans la documentation [Fig.2].



AU FINAL...

A partir du moment où votre site internet arrive à un stade avancé, le JavaScript devient un élément indispensable pour étendre les fonctionnalités et accroître le dynamisme global de tout l'ensemble. Ainsi, Google Maps et Dojo Toolkit s'avèrent être plus qu'intéressants puisqu'ils proposent tous les deux des fonctionnalités interactives. L'un propose de créer une carte avec divers contrôles, qu'on peut visualiser sous divers aspects, avec des marqueurs, des aides et info-bulles qui apparaissent lors d'un clic sur un élément, tandis que l'autre propose des éléments de formulaire qui disposent d'attributs amusants qui plairont aux utilisateurs pour leur convivialité. En conclusion, quel que soit le langage que vous utiliserez pour programmer votre site, vous en viendrez rapidement à manipuler le JavaScript, qui est un langage riche mais pas beaucoup plus compliqué qu'un autre.

■ Vincent Lecomte

Analyste-Développeur WinDev, <http://vincent-lecomte.blogspot.com>

DILIBEL – Hachette Livre Belgique

Economisez jusqu'à 50%



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

Pour votre développement personnel et professionnel, abonnez-vous à Programmez ! www.programmez.com

1 -25%

Abonnement 1 an

49€ au lieu de 65,45 € tarif au numéro - Tarif France métropolitaine

2 +0,8€ par mois

Abonnement Intégral : + archives

1 an au magazine + archives sur Internet et PDF

59€ Tarif France métropolitaine

3 jusqu'à -50%

Abonnement 2 ans + 1 livre numérique ENI

• **79€** au lieu de 130,90 (valeur de 22 numéros) Tarif France métropolitaine + un livre d'une valeur de 23,9 € à 31,9 €, soit un total de 154,8 € à 162,8 €

• **89€** 2 ans au magazine + archives sur Internet et PDF + 1 livre numérique ENI



OUI, je m'abonne

Vous pouvez vous abonner en ligne et trouver tous les tarifs www.programmez.com

- ☐ Abonnement 1 an au magazine : **49 €** (au lieu de 65,45 € tarif au numéro) Tarif France métropolitaine
- ☐ Abonnement Intégral : 1 an au magazine + archives : **59 €** Tarif France métropolitaine
- ☐ Abonnement 2 ans au magazine + livre numérique ENI : **79 €** Tarif France métropolitaine
- ☐ Abonnement 2 ans au magazine + livre numérique ENI + archives : **89 €** Tarif France métropolitaine

Livres à choisir : ☐ Visual Studio 2010 ☐ PHP5.3 ☐ Bing Maps ☐ MySQL 5, Administration et optimisation

☐ Java et Spring, Concevoir, construire et développer une application Java/J2EE avec Spring. Détails sur www.programmez.com/abonnement.php

☐ M. ☐ Mme ☐ Mlle Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

Adresse : _____

Code postal : _____ Ville : _____

Tél : _____ (Attention, e-mail indispensable)

E-mail : _____ @ _____

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

A remplir et retourner sous enveloppe affranchie à : Programmez ! - Service Abonnements - 22 rue René Boulanger - 75472 Paris Cedex 10. abonnements.programmez@groupe-gli.com

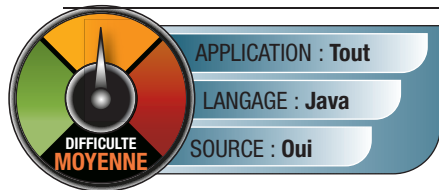
Offre limitée, valable jusqu'au 30 avril 2011

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.

Le magazine du développement
PROgrammez !

Introduction à la programmation par contraintes en Java

Fortement liée au monde de la recherche, la programmation par contraintes est un paradigme de programmation puissant et malheureusement méconnu du grand public. Dans cet article, nous allons tout d'abord nous intéresser aux fondements théoriques sur lesquels est bâtie la programmation par contraintes. Nous mettrons ensuite en pratique ces aspects théoriques pour résoudre le problème bien connu du jeu de Kakuro.



Au même titre que la programmation objet ou que la programmation impérative, la programmation par contraintes peut être considérée comme un paradigme

de programmation à part entière. Là où les deux paradigmes précédemment cités s'attachent à spécifier un ensemble d'étapes à exécuter pour résoudre un problème, la programmation par contraintes se démarque en décrivant la solution d'un problème en termes de propriétés et de contraintes qui doivent être vérifiées. En ce sens, elle peut être considérée comme une forme de programmation déclarative. Pour rappel, cette dernière est un paradigme de programmation qui s'attache à décrire ce que doit faire un programme plutôt que de spécifier comment il doit le faire.

Discipline au carrefour d'un certain nombre de domaines de recherche tels que l'intelligence artificielle, la recherche opérationnelle ou l'analyse numérique, la programmation par contraintes connaît un succès grandissant dans le monde industriel depuis quelques années. Ses domaines de prédilection sont assez divers et variés mais on peut citer les quelques exemples suivants :

- Problèmes de planification et d'ordonnancement : gestion d'un trafic ferroviaire, planification d'une production, ...
- Problèmes de gestion de ressources : mise au point d'emplois du temps, affectation de tâches à des ressources, affectation de marchandises dans des entrepôts suivant diverses contraintes, ...
- Problèmes d'optimisation : optimisation du routage dans les réseaux de télécommunication, ...

Loin d'être exhaustive, cette liste permet de prendre conscience du large champ d'application de la programmation par contraintes. Plus généralement, on désigne par CSP (Constraint Satisfaction Problem) un problème modélisé pour être résolu via la programmation par contraintes.

UN PEU DE THÉORIE ...

Avant tout chose, il me paraît important de poser les bases théoriques permettant de comprendre ce qu'est un CSP. Un CSP est ainsi défini via un triplet (X, D, C) où :

- $X = \{x_1, x_2, x_3, \dots, x_i, \dots, x_n\}$ correspond à l'ensemble des variables du problème.
- $D = \{\text{dom}(x_i) \mid x_i \in X\}$ correspond à l'ensemble des domaines associés aux variables du problème, c'est-à-dire l'ensemble des valeurs possibles pour chaque variable.
- $C = \{c_1, c_2, c_3, \dots, c_j, \dots, c_m\}$ correspond à l'ensemble des

contraintes appliquées aux variables du problème. Ces contraintes pouvant concerner de 1 à n variables.

Une solution d'un CSP correspond à une instanciation complète des variables du problème (avec une valeur de leurs domaines respectifs) respectant l'ensemble des contraintes définies. Ainsi, il est de coutume de qualifier un CSP ayant une solution de *consistant* alors qu'un CSP qui n'en a pas est considéré comme *inconsistant*.

La résolution d'un CSP peut tendre vers 3 objectifs différents :

- Existence d'une solution
- Recherche de l'ensemble des solutions
- Recherche de la solution optimale suivant certains critères prédéfinis

Fortement combinatoires, les CSP nécessitent bien souvent un parcours plus ou moins systématique de l'espace de recherche afin de trouver une solution. Suivant le type d'algorithme utilisé, la taille de l'espace de recherche parcouru sera plus ou moins importante.

Vaste domaine de recherche, la résolution de CSP peut se faire via un grand nombre d'algorithmes différents ayant chacun leurs avantages et leurs inconvénients. L'étude détaillée de ces algorithmes n'étant pas le but poursuivi par cet article, il sera juste fait mention dans les lignes qui suivent du nom des algorithmes de base permettant de résoudre des CSP.

Méthode de résolution énumérative, l'algorithme Backtrack a pour principe simple de tenter d'étendre progressivement une affectation consistante des variables du CSP. En cas d'échec, on change de valeur pour la variable en cours d'affectation jusqu'à ce qu'on ait essayé l'ensemble des valeurs de son domaine. Si aucune de ces valeurs ne permet de poursuivre l'affectation, on revient sur la variable précédente et on réitère ce procédé jusqu'à ce que l'on ait trouvé une solution au CSP ou bien que l'on ait essayé l'ensemble des possibilités. Il est bien évident que suivant la taille du CSP à résoudre, le temps d'exécution de cet algorithme peut se révéler très très long ! Toujours dans la famille des méthodes de résolution énumérative, l'algorithme Forward Checking se révèle bien souvent meilleur, profitant de la propagation locale des contraintes qui est effectuée sur les domaines des variables restant à instancier après chaque affectation.

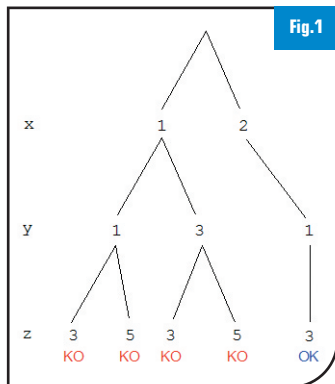
En pratique, après chaque affectation de variable on va filtrer les domaines des variables liées par une contrainte à cette variable en retirant les valeurs violant ces contraintes. Lorsqu'un domaine devient vide, on essaie une nouvelle valeur pour la variable qui était en cours d'affectation et ceci jusqu'à ce que toutes les valeurs de son domaine aient été essayées. Dans ce cas-là, on revient sur la variable précédente et on réitère ce procédé jusqu'à l'obtention d'une solution ou bien que l'on ait parcouru l'ensemble de l'espace

de recherche. Le lecteur curieux et intéressé d'en savoir plus sur ces algorithmes pourra aisément trouver de la documentation de qualité sur internet.

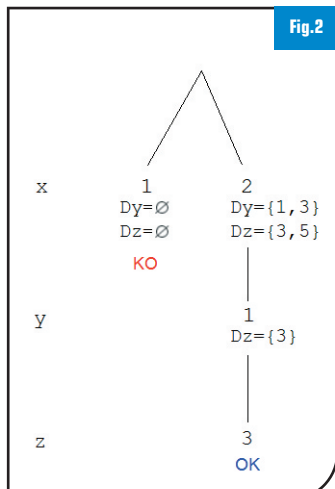
EXEMPLE DE RÉSOLUTION D'UN CSP

Afin de mettre en pratique ces quelques fondements théoriques concernant les CSP, nous allons considérer le problème CSP suivant :

- $X = \{x, y, z\}$
- $D = \{D_x = \{1, 2\}, D_y = \{1, 2, 3\}, D_z = \{3, 4, 5\}\}$
- $C = \{x + y = z\}$



Résolution via le Backtrack



Résolution via le Forward Checking

Le problème considéré possède ainsi 3 variables x , y et z ayant chacun leurs domaines respectifs D_x , D_y et D_z . Enfin, ce CSP possède une seule contrainte qui implique que la somme des variables x et y soit égale à la variable z .

La résolution de ce problème via l'algorithme du Backtrack décrit précédemment est présentée dans le diagramme de la [Fig.1]. On s'aperçoit tout de suite que l'utilisation d'un tel algorithme n'est pas forcément la meilleure solution. En effet, en cherchant à trouver une instantiation consistante du CSP par énumération des valeurs contenues dans les domaines des variables, on risque fort au pire des cas de parcourir l'ensemble de l'espace de recherche pour trouver une solution ou une inconsistance. C'est le cas pour notre CSP, puisqu'une solution consistante est trouvée sur la dernière branche de l'arbre de l'espace de recherche.

Présentée dans le diagramme de la [Fig.2], la résolution du CSP

via l'algorithme Forward Checking montre un espace de recherche plus restreint grâce à la propagation des contraintes réalisée par cet algorithme.

CHOCO : METTEZ DES CONTRAINTES DANS VOTRE JAVA ...

Les fondements théoriques de la programmation par contraintes mis en place, nous allons maintenant passer au côté pratique de ce paradigme de programmation. Pour cela, nous nous intéressons à la bibliothèque de programmation par contraintes Choco. Issue d'un projet de recherche français soutenu par le monde industriel, la bibliothèque Choco fut initialement créée en 1999 mais c'est réellement en 2003 que le projet connaît un premier essor avec sa première implémentation via le langage Java. La bibliothèque est déjà pleinement opérationnelle mais elle n'est pas réellement optimisée en termes de performances. En outre, son architecture laisse clairement à désirer avec un découpage objet plutôt moyen et une API

pas vraiment conviviale. Afin de corriger ces défauts et de tirer parti des différents retours des utilisateurs, la version 2 est repensée en profondeur. Sortie en 2008, elle vient avec une architecture soignée offrant une séparation nette entre la partie modélisation et la partie résolution des problèmes de satisfaction de contraintes tout en étant facilement extensible. Les performances du solveur de Choco se voient ainsi grandement améliorées et l'API de programmation plus orientée objet devient maintenant plus conviviale. Avec cette nouvelle mouture, le monde de la programmation par contraintes est à la portée de tous en Java.

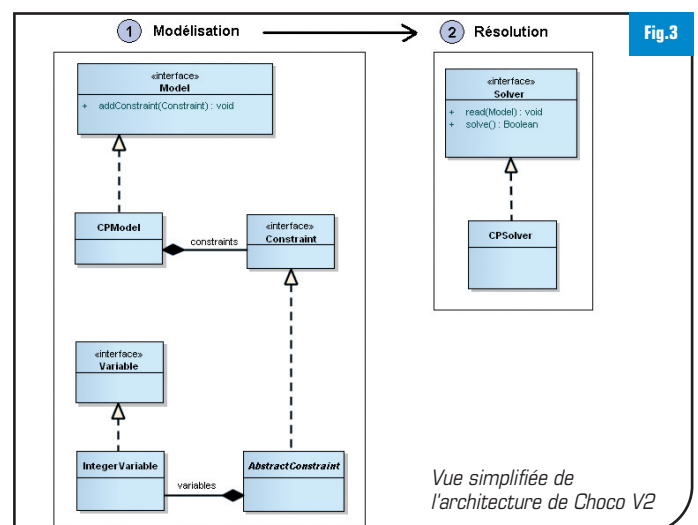
ARCHITECTURE DE CHOCO

Actuellement en version 2.1 depuis la fin de l'année 2010, la bibliothèque Choco compte différents projets parmi lesquels on retrouve le projet de base Choco-Solver qui représente le solveur de problèmes CSP. L'architecture de ce solveur [Fig.3] est basée sur un modèle objet soigné au sein duquel on distinguera 2 grandes parties :

1. La partie modélisation qui va permettre de représenter le problème CSP que l'on souhaite résoudre
2. La partie solveur qui à partir d'une modélisation Choco d'un problème CSP va réaliser sa résolution

Une modélisation d'un problème avec Choco est une implémentation de l'interface `Model`. Choco fournit une implémentation de cette interface via la classe `CPModel` qui est une représentation via le paradigme de programmation par contraintes d'un problème. Ce modèle est composé d'un ensemble de contraintes modélisées via l'interface `Constraint`. Les contraintes portent sur des variables qui implémentent l'interface `Variable`. Plusieurs implémentations de variables sont bien entendues proposées par Choco. L'ajout de contraintes au modèle du CSP se fait via la méthode `addConstraint` qui est contractualisée dans l'interface `Model`.

Une fois l'étape de modélisation terminée, on passe à la résolution du problème modélisé. Pour cela, Choco utilise la notion de solveur. Représenté par l'interface `Solver`, un solveur propose une méthode de lecture du modèle d'un problème ainsi qu'une méthode permettant sa résolution. Cette dernière renvoyant un booléen qui indique si le problème est consistant ou non. Ici encore, Choco propose une implémentation de solveur, la classe `CPSolver`, utilisant la programmation par contraintes pour résoudre le problème associé au modèle qu'il va lire. Si le problème est consistant, l'affichage des solutions se fera via le solveur en parcourant les instantiations valides des



Vue simplifiée de l'architecture de Choco V2

variables du problème CSP. Le solveur implémenté dans la classe CPSolver est basé sur un algorithme de résolution énumérative qui est optimisé grâce à divers algorithmes de propagation de contraintes (cf. algorithme du Forward-Checking présenté précédemment). En outre, le solveur de Choco est extensible et il est possible de le configurer en lui appliquant différentes stratégies qui seront ainsi utilisées durant la résolution du problème de contraintes.

LE PROBLÈME DU KAKURO

Jeu logique d'origine japonaise, le Kakuro est un exemple parfait de problème de satisfaction de contraintes. Le jeu est constitué d'une grille comportant des cases remplies (noires) ou vides (blanches). Les cases noires peuvent être divisées en deux par une diagonale allant du coin haut à gauche au coin bas à droite de telle sorte que la demi-case située à droite indique un nombre à réaliser de manière horizontale et que celle située à gauche indique un nombre à réaliser de manière verticale. L'objectif du Kakuro est de remplir les cases blanches avec des chiffres allant de 1 à 9 de sorte que la somme de tous les chiffres associés à une demi-case noire soit égale au nombre défini dans celle-ci. En outre, un nombre ne peut contenir deux fois le même chiffre dans sa somme. Avec cette dernière règle, il devient ainsi possible de garantir l'unicité de la solution d'une grille de Kakuro créée.

MODÉLISATION DU KAKURO

La décomposition en objets du jeu de Kakuro nous amène à considérer la grille de jeu. Cette dernière est composée de cases qui peuvent être noires ou blanches, ce que l'on traduira de la manière suivante :

```
// classe abstraite modélisant une case
public abstract class Case {
    // enum représentant le type d'une case
    public enum TypeCase { BLACK, WHITE};

    ...

    // par défaut renvoie la somme horizontale d'une case
    public int getSumHor() {
        return -1;
    }

    // par défaut renvoie la somme verticale d'une case
    public int getSumVer() {
        return -1;
    }

    // par défaut renvoie la variable associée à la case
    public IntegerVariable getVariable() {
        return null;
    }

    ...
}

// classe modélisant une case blanche
public class WCase extends Case {
```

```
// variable Choco associée à la case
private IntegerVariable variable;

...

@Override
public IntegerVariable getVariable() {
    return variable;
}

...
}

// classe modélisant une case noire
public class BCase extends Case {
    // somme horizontale
    private int sumHor;
    // somme verticale
    private int sumVer;

    ...

    @Override
    public int getSumHor() {
        return sumHor;
    }

    @Override
    public int getSumVer() {
        return sumVer;
    }

    ...
}
```

Ici, on constate que les cases noires se voient associées des sommes horizontales et verticales alors que les cases blanches se voient associées une variable CSP de type IntegerVariable. A la création du problème de Kakuro, chaque variable associée à une case blanche verra son domaine initialisé à l'ensemble de valeurs allant de 1 à 9 comme ceci :

```
Choco.makeIntVar("W" + line + "," + col, 1, 9, Options.V_ENUM);
```

Nous allons ensuite considérer les sommes à obtenir comme des vecteurs composés de cases blanches et ayant une somme à satisfaire (celle correspondant au nombre de la demi-case noire associée au vecteur). Ces vecteurs pouvant être de type horizontaux ou verticaux. Dans le programme complet associé à cet article, nous utilisons pour créer ces vecteurs une classe de parsing nommée ParserKakuro. A partir d'une grille de Kakuro représentée dans un fichier texte, cette classe va charger en mémoire les vecteurs correspondants ainsi que la représentation de la grille avec ses cases noires et blanches.

Une fois le jeu de Kakuro représenté en mémoire, nous pouvons passer à la représentation CSP du problème via la bibliothèque Choco. Pour cela, nous définissons tout d'abord un modèle de type Model qui va recevoir les différentes contraintes de notre problème.

Ces dernières s'appuient sur les vecteurs créés précédemment. Ainsi, pour chaque vecteur, la somme des cases blanches le composant devra être égale au nombre du vecteur. En outre, toutes les valeurs instanciées dans les cases blanches d'un même vecteur devront être différentes. L'expression de ces contraintes est présentée dans le code suivant :

```
// création du modèle Choco
Model model = new CPModel();
// parsing de la grille de Kakuro
ParserKakuro parser = new ParserKakuro(filePath);
parser.process(model);
// on récupère la grille
HashMap<Integer, Case[]> hMap = parser.getGrid().toHashMapOf
Array(new Case[0]);
// parcours des vecteurs horizontaux
for(Point coupleHor : parser.getPositionsVecteursHor()){
    Case[] hor = hMap.get(coupleHor.x);
    // on récupère la somme associée
    if(hor != null){
        int sumHor = hor[coupleHor.y].getSumHor();
        List<IntegerVariable> listDomainVarHor = new ArrayList<
IntegerVariable>();
        int j = coupleHor.y + 1;

        while(j < hor.length && TypeCase.WHITE.equals(hor[j]
.getType())){
            listDomainVarHor.add(hor[j].getVariable());
            j++;
        }

        // création des contraintes CSP pour ce vecteur
        IntegerVariable[] tmp = listDomainVarHor.toArray(new Inte
gerVariable[0]);
        // variables d'un vecteur sont toutes différentes ...
        model.addConstraint(Choco.allDifferent(tmp));
        // ... et la somme doit être égale au nombre du vecteur
        model.addConstraint(Choco.eq(Choco.sum(tmp), sumHor));
    }
}

// parcours des vecteurs verticaux
for(Point coupleVer : parser.getPositionsVecteursVer()){
    // on récupère la somme associée
    int sumVer = hMap.get(coupleVer.x)[coupleVer.y].getSumVer();
    List<IntegerVariable> listDomainVarVer = new ArrayList<Inte
gerVariable>();
    int x = coupleVer.x + 1;

    while(x < hMap.size() && TypeCase.WHITE.equals(hMap.get
(x)[coupleVer.y].getType())){
        listDomainVarVer.add(hMap.get(x)[coupleVer.y].getVariable());
        x++;
    }

    // création des contraintes CSP pour ce vecteur
    IntegerVariable[] tmp = listDomainVarVer.toArray(new Inte
gerVariable[0]);
```

```
// variables d'un vecteur sont toutes différentes ...
model.addConstraint(Choco.allDifferent(tmp));
// ... et la somme doit être égale au nombre du vecteur
model.addConstraint(Choco.eq(Choco.sum(tmp), sumVer));
}
```

RÉSOLUTION DU KAKURO

La modélisation du Kakuro en problème CSP faite via Choco, il reste à réaliser sa résolution afin de trouver une solution à notre problème. Pour cela, nous utilisons un objet de type Solver auquel nous passons le modèle Choco qui vient d'être créé. La résolution consiste ensuite à appeler la méthode solve de notre objet Solver instancié :

```
// création du solveur
Solver solver = new CPSolver();
// lecture du modèle
solver.read(model);
// résolution
solver.solve();
```

Comme expliqué brièvement lors de la présentation de Choco, il est possible de configurer le solveur Choco afin de lui appliquer des options de résolution spécifiques mais dans le cas de notre problème, cela ne s'avère pas vraiment nécessaire. L'affichage des solutions, dans le cas où il y en a, se fait également de manière aisée :

```
// on vérifie que le problème est consistant
if (solver.checkSolution()) {
    // on parcourt les variables associées aux cases blanches
    for(Case wcase : wcases) {
        // Un Kakuro ayant une solution unique on sait que si
        solution il y a,
        // la case n'a qu'une seule valeur possible
        int value = solver.getVar(wcase.getVariable()).getVal();
        ...
    }
}
```

CONCLUSION

La programmation par contraintes est un domaine passionnant qui dépasse le simple cadre de la recherche. Cet article d'introduction, loin d'être exhaustif, aura permis de mettre en avant la puissance de ce paradigme de programmation encore trop méconnu des développeurs Java et du monde industriel. Et ceci de manière ludique via la résolution du problème du Kakuro. Le succès grandissant que connaît le mouvement NoSQL le montre, le pragmatisme semble faire son retour dans le monde informatique. Figure de proue de ce retour, NoSQL remet au goût du jour une évidence bien souvent oubliée par les développeurs travaillant dans le monde industriel : il est primordial d'utiliser l'outil le mieux adapté à son problème. A ce titre, la programmation par contraintes devrait être de nouveau considérée dans le futur pour résoudre des problèmes visant à satisfaire un certain nombre de contraintes et pour lesquels son paradigme de programmation fait merveille.

■ Sylvain Saurel

Ingénieur d'Etudes Java / JEE

sylvain.saurel@gmail.com

Instrumentalisez vos Linux

L'instrumentation des systèmes informatiques est spécifiée dans deux standards: CIM et WBEM. Nous mettons en œuvre deux implémentations de ces standards sous Linux.



L'instrumentation des systèmes est une branche de l'ingénierie qui applique des techniques pour la création d'un système capable d'acquisition de données d'une

part et de commandes d'autre part. De nos jours l'instrumentation est omniprésente, par exemple dans une Formule 1 ou même dans la voiture de monsieur tout le monde. L'informatique est bien entendu un terrain privilégié car les besoins sont très grands. Souvent il ne s'agit pas de pouvoir surveiller une seule machine, mais tout un parc, et dans chaque machine, on voudra tout savoir par exemple des processus, des cartes réseau, de la configuration du pare-feu, etc. Il existe de nombreuses solutions à ce problème. Nous nous intéressons aujourd'hui à deux standards ouverts : CIM et WBEM, qui sont destinés à être utilisés de concert. Ils sont spécifiés par un consortium d'industriels: la Distributed Management Task Force. Nous avons déjà rencontré ces deux standards dans l'article « *Observez la charge de votre processeur multi-cœur sous Windows* » de Programmez 110. Nous avons alors utilisé l'implémentation WMI que Microsoft fournit avec ses systèmes d'exploitation. En revanche les implémentations avec lesquelles nous travaillons aujourd'hui sont multi-plates-formes.

1 APERÇU DE CIM

CIM signifie Common Information Model. Son but est de représenter tous les éléments d'un environnement informatique. Pour cela, des classes (autrement dit des structures de données) sont définies, et les membres des classes représentent les détails d'un élément. Ainsi par exemple, si nous considérons une classe CIM_Process qui décrira un processus, les membres, parmi d'autres, Handle, Name et Priority, préciseront le PID (identifiant), le nom et la priorité du processus. En plus des membres, une classe peut comporter des méthodes. Ces méthodes peuvent théoriquement être invoquées, mais les implémentations supportent rarement cette fonctionnalité.

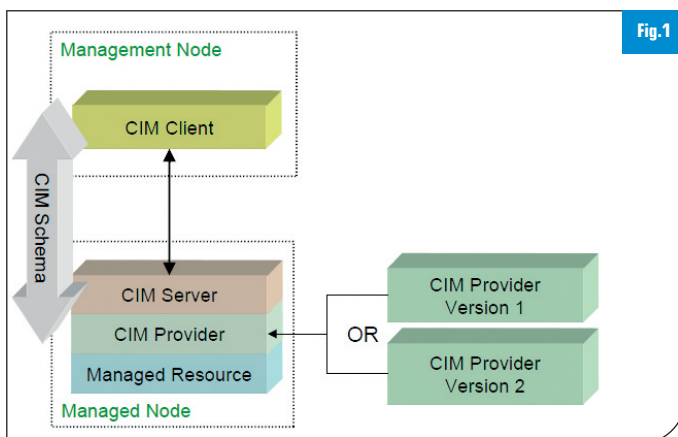


Fig.1

Un client CIM interroge des providers à travers un protocole WBEM et un serveur CIM, pour obtenir des informations relatives à une ressource gérée.

Ces classes présentent une très grande analogie avec la programmation objet, et comme avec celle-ci, les classes sont instanciées par l'implémentation. Les instances sont accessibles à travers des providers (fournisseurs) chapeautés par un serveur. Toujours à propos de notre exemple, une implémentation sera capable de fournir la classe CIM_Process d'une part, et toutes les instances de celle-ci, une par processus, d'autre part. Les classes sont rangées dans des espaces de noms. L'espace de nom par défaut est root/cimv2 et il existe un mécanisme d'héritage. Ainsi selon les implémentations, il est possible de rencontrer des classes Linux_UnixProcess ou PG_UnixProcess qui dériveront chacune de CIM_Process. Mentionnons encore l'existence des associations qui présente une analogie avec l'agrégation en programmation objet

2 APERÇU DE WBEM

WBEM, ou Web Based Enterprise Management, est la partie complémentaire de CIM. A notre niveau il nous suffit de savoir que le protocole de communication est basé sur HTTP et qu'il utilise le port 5988 par défaut. Un client CIM interroge un serveur CIM via ce protocole et reçoit en réponse des données au format XML [Fig.1]. Ce principe de communication permet facilement d'interroger des serveurs CIM résidant sur des machines distantes.

3 POUR SE FAIRE LA MAIN

Pour faire connaissance de façon pratique avec tout cela, manipulons une implémentation légère : SBLIM. Nous travaillons avec une distribution Ubuntu. Avec celle-ci il suffit d'installer les paquets wbemcli, sfcv et sblim-cmpi-base depuis le gestionnaire de paquets pour être opérationnel. Ubuntu lance automatiquement les démons lors de l'installation. Si vous utilisez une autre distribution, vous devrez d'abord trouver les paquets équivalents, puis, bien entendu, lancer le démon avec les droits de l'administrateur. Les essais avec un système Ubuntu virtualisé sous VirtualBox ne sont pas toujours concluants, ni ici, ni pour la suite de cet article. Dès que tout est prêt nous pouvons nous faire la main avec le client wbemcli, par exemple en lui demandant des classes CIM disponibles avec cette implémentation.

```
fred@mars-linux:~$ wbemcli -nl ecn 'http://localhost:5988/root/cimv2'
localhost:5988/root/cimv2:Linux_OSProcess
localhost:5988/root/cimv2:CIM_Processor
localhost:5988/root/cimv2:Linux_BaseBoard
# etc, etc.
```

Nous pouvons alors énumérer les instances de la classe Linux_BaseBoard, qui décrit la carte mère du système. Il n'y a pour un PC de bureau qu'une seule instance a priori :-].

```
fred@mars-linux:~$ wbemcli -nl ei 'http://localhost:5988/root/cimv2:Linux_BaseBoard'
```

```
localhost:5988/root/cimv2:Linux_BaseBoard.CreationClassName=»
Linux_BaseBoard»,Tag=»Hewlett-Packard::CNF81926ZS»
-UUID=
-VendorCompatibilityStrings=
# etc, etc.
```

SBLIM est une implémentation légère de CIM et WBEM. Le nombre de classes disponibles est relativement faible. Toutefois c'est une implémentation standard et il sera possible d'utiliser wbemcli pour les premiers essais avec une implémentation beaucoup plus complète à laquelle nous nous intéressons maintenant : OpenPegasus

4 OPENPEGASUS

OpenPegasus (<http://www.openpegasus.org>) est sans doute la meilleure implémentation de CIM et WBEM au moment de la rédaction de cet article. Elle est disponible pour Windows, où elle peut y remplacer WMI l'implémentation officielle de Microsoft et ses clients COM fâcheux, pour un grand nombre de système UNIX et bien sûr pour Linux. Par contre il est probable que vous devrez compiler OpenPegasus vous-même à partir des sources. Pour ce faire, la séquence de commandes classiques ./configure, make, make install ne fonctionnera pas. Voici comment procéder :

- désarchiver les sources, par exemple dans un répertoire /home/fred/src/pegasus
- choisir un répertoire d'installation pour OpenPegasus. Par exemple /home/fred/pegasus.
- configurer alors des variables d'environnement ainsi, AVANT toute opération, y compris la compilation :

```
PEGASUS_ROOT=/home/fred/src
PEGASUS_HOME=/home/fred/Pegasus
PATH=$PEGASUS_HOME/bin:$PATH
LD_LIBRARY_PATH=$PEGASUS_HOME/lib:$LD_LIBRARY_PATH
PEGASUS_PLATFORM=LINUX_IX86_GNU
```

La dernière variable définit pour quelle plate-forme cible l'implémentation doit être compilée. Les noms des plates-formes cibles disponibles se déduisent des noms des fichiers en-tête que l'on trouvera dans le répertoire /home/fred/src/pegasus/src/Pegasus/Common. Ainsi, du fichier en-tête Platform_LINUX_IX86_GNU.h on déduit la variable d'environnement LINUX_IX86_GNU. Le suffixe, ici GNU, est important car il définit le compilateur utilisé, dans notre cas g++. Ensuite il suffit de lancer la compilation, qui sera suivie automatiquement par une grande phase de test, avec un simple 'make' à la racine des sources. Pour que tous les tests se déroulent convenablement, vous devez lancer la compilation avec les droits du super-utilisateur. Attention à l'environnement si vous passez par sudo sous Ubuntu. Enfin sachez qu'une partie des tests échouera inévitablement si vous travaillez avec un Ubuntu virtualisé sous VirtualBox. Toutefois, ceci ne vous empêchera pas de compiler l'implémentation complètement ni de compiler et essayer les exemples qui vont suivre. Enfin le démon d'OpenPegasus doit être lancé avec les droits du super-utilisateur et les variables d'environnements données plus haut. Il peut être nécessaire de stopper celui de SBLIM en premier lieu :

```
/etc/init.d/sfcb stop
cimserver -D $PEGASUS_HOME # cimserver -s pour arrêter
```

On pourra alors tester que tout fonctionne avec wbemcli dans un premier temps, puis avec cimcli, le client de l'implémentation.

5 HELLOWORLD !

Ici un helloworld consistera en une simple connexion au serveur. Nous utilisons l'API C++, très facile à mettre en œuvre, qui vient avec l'implémentation. On utilisera la documentation en ligne sur le site officiel d'OpenPegasus, plutôt que celle qui vient avec les sources, car conçue pour doc++ qui ne sera peut-être pas disponible sur votre Linux. Voici donc notre premier exemple :

```
#include <Pegasus/Client/CIMClient.h>

PEGASUS_USING_PEGASUS;
PEGASUS_USING_STD;

int main(int argc, char** argv)
{
    try
    {
        CIMClient client;
        client.connectLocal();
        cout << «Hello Pegasus ! » << endl;
        client.disconnect();
    }
    catch(Exception& e)
    {
        cerr << «Error: » << e.getMessage() << endl;
        exit(1);
    }
    return 0;
}
```

Nous nous connectons en local, d'où l'emploi de la méthode *connectLocal* de la classe du client. Sinon la méthode *connect* permettrait de toucher un serveur distant. On compilera ce code ainsi :

```
g++ -D PEGASUS_PLATFORM_LINUX_IX86_GNU -I$PEGASUS_HOME -L$PEGASUS_HOME/lib -lpegcommon -lpegclient -lpthread -lrt -o helloworld helloworld.cpp
```

6 LISTER LES PROCESSUS ACTIFS

Voici maintenant un exemple (ei_demo.cpp) de client qui liste toutes les instances de la classe CIM_Process, donc finalement tous les processus actifs

```
#include <Pegasus/Client/CIMClient.h>

PEGASUS_USING_PEGASUS;
PEGASUS_USING_STD;

int main(int argc, char** argv)
{
    CIMNamespaceName nspace(«root/cimv2»);
    CIMName nclass(«CIM_Process»);
    try
```



```

{
    Boolean    deepInheritance = true;
    Boolean    localOnly = true;
    Boolean    includeQualifiers = false;
    Boolean    includeClassOrigin = false;
    Array<CIMInstance> cimInstances;
    CIMClient client;

    client.connectLocal();
    cimInstances = client.enumerateInstances(
        nspace,
        nclass,
        deepInheritance,
        localOnly,
        includeQualifiers,
        includeClassOrigin);
    client.disconnect();
    cout << «Nombre de processus: » << cimInstances.size() << endl;

    for(UInt32 i =0; i<cimInstances.size();i++)
    {
        CIMName name(«Name»);
        CIMName handle(«Handle»);
        UInt32 index = cimInstances[i].findProperty(name);
        CIMProperty property = cimInstances[i].getProperty(index);
        cout << «Nom: » << property.getValue().toString() << « »;
        index = cimInstances[i].findProperty(handle);
        property = cimInstances[i].getProperty(index);
        cout << «Handle (pid): » << property.getValue().toString()
    } << endl;

    }
}
catch(Exception& e)
{
    cerr << «Error: »<< e.getMessage() << endl;
    exit(1);
}
}

```

Nous voyons que nous pouvons nous déconnecter du serveur dès que les instances sont obtenues. Ensuite nous affichons quelques propriétés, nom et pid, de ces instances. Pour obtenir une valeur de propriété, il convient de procéder en trois temps. Créer un objet CIMName, à partir du nom de la propriété, obtenir l'index de celle-ci, puis à partir de l'index, obtenir finalement sa valeur. L'exemple suivant (eip_demo.cpp) est une variante du précédent. Il affiche pour chaque instance un 'ObjectPath', qui est ni plus ni moins qu'un dictionnaire de toutes les propriétés associées à leur valeurs :

```
#include <Pegasus/Client/CIMClient.h>
```

```

PEGASUS_USING_PEGASUS;
PEGASUS_USING_STD;

int main(int argc, char** argv)
{

```

```

CIMNamespaceName nspace(«root/cimv2»);
CIMName nclass(«CIM_Process»);
try
{
    Boolean    deepInheritance = true;
    Boolean    localOnly = true;
    Boolean    includeQualifiers = false;
    Boolean    includeClassOrigin = false;
    Array<CIMObjectPath> cimInstanceNames;
    CIMClient client;

    client.connectLocal();
    cimInstanceNames = client.enumerateInstanceNames(
        nspace,
        nclass);
    client.disconnect();
    cout << «Nombre de processus: » <<
        cimInstanceNames.size() << endl;

    for(UInt32 i =0; i<cimInstanceNames.size();i++)
    {
        cout << cimInstanceNames[i].toString() << endl;
    }
}
catch(Exception& e)
{
    cerr << «Error: »<< e.getMessage() << endl;
    exit(1);
}
}

```

7 OBTENIR UNE INSTANCE PARTICULIÈRE

Nous nous intéressons maintenant à la question de savoir comment obtenir des valeurs de propriétés d'une instance particulière. Pour l'exemple nous voulons en savoir plus sur le processus de pid=1, dont on est sûr qu'il existe sous Linux car il s'agit du processus init, père de tous les autres. Voici le code (gi_demo.cpp) :

```
#include <Pegasus/Client/CIMClient.h>
```

```

PEGASUS_USING_PEGASUS;
PEGASUS_USING_STD;

int main(int argc, char** argv)
{
    CIMNamespaceName nspace(«root/cimv2»);
    CIMName nclass(«PG_UNIXProcess»);
    try
    {
        Boolean    deepInheritance = true;
        Boolean    localOnly = false;
        Boolean    includeQualifiers = false;
        Boolean    includeClassOrigin = false;
        Array<CIMKeyBinding> keyBindings;
        CIMName cscnProperty(«CSCreationClassName»);
        CIMName csnProperty(«CSName»);
        CIMName oscnProperty(«OSCreationClassName»);

```

```

CIMName osnProperty(«OSName»);
CIMName ccnProperty(«CreationClassName»);
CIMName handleProperty(«Handle»);
CIMClient client;

client.connectLocal();
/*
keyBindings.append(CIMKeyBinding(csccnProperty,
    «CIM_UnitaryComputerSystem», CIMKeyBinding::STRING));
keyBindings.append(CIMKeyBinding(csnProperty,
    «mars», CIMKeyBinding::STRING));
keyBindings.append(CIMKeyBinding(osccnProperty,
    «CIM_OperatingSystem», CIMKeyBinding::STRING));
keyBindings.append(CIMKeyBinding(osnProperty,
    «Debian GNU/Linux Distribution», CIMKeyBinding::STRING));
*/

keyBindings.append(CIMKeyBinding(csccnProperty,
    String::EMPTY, CIMKeyBinding::STRING));
keyBindings.append(CIMKeyBinding(csnProperty,
    String::EMPTY, CIMKeyBinding::STRING));
keyBindings.append(CIMKeyBinding(osccnProperty,
    String::EMPTY, CIMKeyBinding::STRING));
keyBindings.append(CIMKeyBinding(osnProperty,
    String::EMPTY, CIMKeyBinding::STRING));

keyBindings.append(CIMKeyBinding(ccnProperty,
    «PG_UnixProcess», CIMKeyBinding::STRING)); //
keyBindings.append(CIMKeyBinding(handleProperty,
    «1», CIMKeyBinding::STRING));
CIMObjectPath instanceName = CIMObjectPath(String::EMPTY,
    nspace, nclass,
    keyBindings);

CIMInstance cimInstance = client.getInstance(
    nspace,
    instanceName,
    localOnly,
    includeQualifiers,
    includeClassOrigin);

client.disconnect();

// Nom du process
CIMName name(«Name»);
Uint32 index = cimInstance.findProperty(name);
CIMProperty property = cimInstance.getProperty(index);
cout << «Nom: » << property.getValue().toString() << endl;

// Priorité
CIMName priority(«Priority»);
index = cimInstance.findProperty(priority);
property = cimInstance.getProperty(index);
cout << «Priorité: » << property.getValue().toString() << endl;

```

```

// Description
CIMName description(«Description»);
index = cimInstance.findProperty(description);
property = cimInstance.getProperty(index);
cout << «Description: » << property.getValue().toString() << endl;

// caption
CIMName caption(«Caption»);
index = cimInstance.findProperty(caption);
property = cimInstance.getProperty(index);
cout << «Caption: » << property.getValue().toString() << endl;

// Execution state
CIMName executionstate(«ExecutionState»);
index = cimInstance.findProperty(executionstate);
property = cimInstance.getProperty(index);
cout << «Etat d'exécution: » << property.getValue().toString() << endl;
}
catch(Exception& e)
{
    cerr << «Error: » << e.getMessage() << endl;
    exit(1);
}
}

```

Le cheminement est le suivant. Nous devons d'abord définir un `ObjectPath` qui servira à l'implémentation pour rechercher l'instance correspondante. Et pour définir cet `ObjectPath`, nous devons commencer par fournir un dictionnaire de clés. Selon la requête et la classe visées, il faut «un certain nombre» de clés. Ce nombre n'étant en général pas documenté, il faut expérimenter. La plupart des clés peuvent avoir des valeurs vides, mais il faut bien entendu aussi des valeurs concrètes. Pour notre exemple la classe `(PG_UnixProcess)` et le PID, ou `Handle`, sont suffisants. Après quoi nous obtenons l'instance et nous en affichons quelques propriétés. Nous constatons alors avec satisfaction que le nom du processus est bien 'init' :-)

8 INVOQUER UNE MÉTHODE

Certaines classes comportent des méthodes qu'il est théoriquement possible d'invoquer. Ainsi `CIM_OperatingSystem` propose les méthodes *Reboot* et *Shutdown*. Essayons de redémarrer le système en invoquant la méthode *Reboot*. Le code est disponible sur notre site dans le fichier `im_demo.cpp`. Comme nous considérons que l'instance de l'OS est unique, nous allons au plus simple, en les énumérant toutes et en ne considérant que la première trouvée. Et... cela ne fonctionne pas

```

Error: CIM_ERR_NOT_SUPPORTED: OperatingSystemProvider does not
support invokeMethod

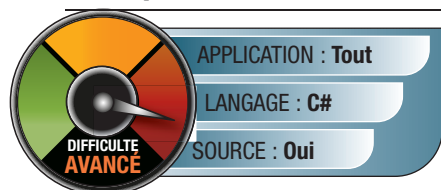
```

Comme bien souvent avec les implémentations CIM, l'invocation de méthodes n'est pas supportée. On rencontre ce souci avec la WMI de Windows également. Il nous faudrait donc réécrire le provider de la classe pour supporter la fonctionnalité. Pour une prochaine fois :-)

■ Frédéric Mazué - fmazue@programmez.com

Async : une nouvelle approche de la programmation Asynchrone sous .NET

Async est une nouvelle manière très séduisante d'aborder la programmation Asynchrone sous .NET. Nous la découvrons aujourd'hui, en attendant qu'elle fasse partie intégrante d'un prochain C# 5.0



Classiquement, la programmation asynchrone est basée sur un mécanisme par fonction de rappel, ou callback. On la trouve chaque fois que l'on n'est pas

complètement maître du déroulement d'un programme. Par exemple lorsque celui-ci est tributaire d'actions de l'utilisateur, qui seront effectuées à un moment imprévisible, ou encore lorsque le programme doit interagir avec un autre programme situé sur une machine distante. Une simple opération de téléchargement pouvant se dérouler bien, médiocrement ... ou pas du tout, la partie d'une application concernée doit travailler de manière asynchrone afin de ne pas bloquer tout le reste de l'application. En fait dans la programmation moderne, la programmation asynchrone est partout. Ainsi en programmation Win32, la procédure de fenêtre qui s'occupe de tout, réagir à un clic de l'utilisateur, rafraîchir l'affichage, n'est rien d'autre qu'une fonction de rappel que le système invoque de manière asynchrone. En programmation .NET, pour reprendre l'exemple du téléchargement, on commence par spécifier une fonction de rappel, puis on lance le téléchargement asynchrone. Lorsque celui-ci se termine, le système invoque la fonction de rappel afin que l'application puisse être avertie du résultat.

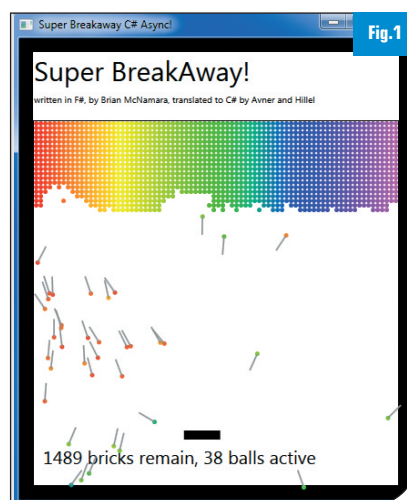
1 APERÇU DES PROBLÈMES

Les problèmes sont nombreux. Par exemple en programmation Win32, lorsque la procédure de fenêtre lance une opération longue, comme un calcul lourd, l'interface n'est plus réactive aux actions de l'utilisateur. Le même problème se rencontre bien sûr en programmation .NET lorsqu'un gestionnaire d'événement lance une opération longue. Une première solution est alors de faire en sorte de lancer l'opération longue dans un thread séparé. Mais très vite la logique de l'application va se compliquer : quand le thread se terminera-t-il ? Comment réagir si une exception est levée dans le thread ? Si l'on reprend le cas du téléchargement, on voit vite que si tout est relativement simple pour un téléchargement, tout se compliquera très vite dans le cas de plusieurs téléchargements simultanés, surtout si l'exploitation des résultats doit être faite dans un ordre déterminé. L'invocation erratique des fonctions de rappel par le runtime deviendra vite un vrai souci. Au final le code d'une application asynchrone devient du code spaghetti, difficile à écrire, et encore plus difficile à relire et à maintenir. C'est ici qu'intervient Async. Celui-ci permet d'écrire du code asynchrone avec un look et même une démarche (ou quasiment) de programmation synchrone. Autrement dit, Async gomme les problèmes évoqués précédemment pour le programmeur qui écrira du code asynchrone quasiment sans le

savoir. Async, en version CTP (Community Technology Preview) mérite vraiment d'être découvert, et il est très probable qu'il fasse partie intégrante de la prochaine mouture de .NET.

2 MICROSOFT ASYNC CTP

Le framework Async est conçu pour s'intégrer à Visual Studio, mais, en ce qui concerne la version CTP qui sert de base à cet article, seulement pour Visual Studio Professional, Premium, ou Ultimate en anglais. Ce framework consiste d'une part en un patch du compilateur .NET et d'autre part en deux assemblies, un pour les applications de bureau et un pour les applications Silverlight. Lors de l'installation, le compilateur C# se voit appliquer le patch. Après quoi



Un casse-briques hallucinant, écrit avec Async

il connaît deux nouveaux mots-clés, *async* et *await*. Si l'on désinstalle Async, on retrouve le compilateur dans son état précédent. Dans cet article nous ne parlons que de C#, mais le compilateur VB.NET est lui aussi modifié et le langage également enrichi des mots-clés Async et Await. Les exemples qui accompagnent cet article sont donc tous transposables en VB.NET. Hormis le patch, Async s'installe

par défaut dans un répertoire baptisé Microsoft Visual Studio Async CTP, sous votre répertoire 'Mes documents'. C'est dans le sous-répertoire 'Exemples' que l'on trouvera ... des exemples, mais aussi les assemblies mentionnés plus haut. Certains exemples sont assez impressionnants, tel ce casse-briques hallucinant [Fig.1] ou encore cette animation d'une myriade d'avions en papier [Fig.2]. On trouvera encore un sous-répertoire contenant de la documentation et les spécifications d'Async.

3 HELLO WORLD

Les interfaces utilisateur sont un des champs d'application privilégiés d'Async. Toutefois, pour bien comprendre cet outil nous allons commencer par travailler avec une application console toute simple, de type Hello World. En voici le code, qui pour l'instant est totalement synchrone :


```

using System;
using System.Threading;

namespace SimpleSync
{
    class Program
    {
        string texte = «Programmez!»;
        private void BigWork()
        {
            Console.WriteLine(«Debut BigWork»);
            Thread.Sleep(2000);
            texte += « Abonnez vous! :-»»;
            Console.WriteLine(«Fin BigWork»);
        }

        private void Go()
        {
            BigWork();
            for (int i = 1; i < 10; i++)
            {
                Console.WriteLine(texte);
                Thread.Sleep(500);
            }
        }

        public static void Main(string[] args)
        {
            Program p = new Program();
            p.Go();
        }
    }
}

```

Cet exemple, comme tous ceux accompagnant cet article, est disponible sur notre site. Dans le cas présent, la méthode Go veut matérialiser le thread principal d'une application fenêtrée. Le code affiche une chaîne de caractères à intervalles réguliers. La méthode BigWork veut matérialiser un calcul lourd, et donc long, à l'issue duquel la chaîne à afficher par la partie principale de l'application est modifiée. Cette méthode est invoquée au tout début du code de la

méthode Go, et, pendant tout le temps de son exécution, la méthode Go ne peut pas travailler. Si notre méthode Go était vraiment le thread principal d'une application fenêtrée, celle-ci serait alors totalement bloquée, incapable de répondre aux sollicitations de l'utilisateur, pendant tout le temps d'exécution de BigWork, ce qui n'est pas acceptable.

4 PREMIERS PAS AVEC ASYNC

Nous ne parlons pas de la solution jusqu'ici classique consistant à placer BigWork dans un thread. A la place nous mettons en œuvre Async comme ceci :

```

using System;
using System.Threading;
using System.Threading.Tasks;

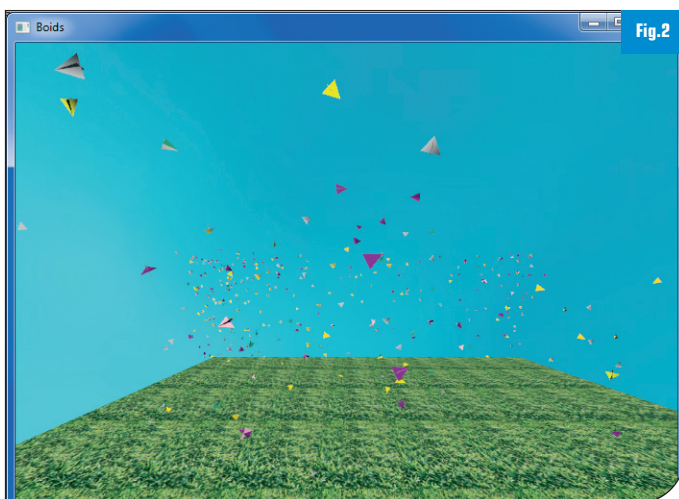
namespace SimpleAsync
{
    class Program
    {
        string texte = «Programmez!»;

        private async void BigWork()
        {
            Console.WriteLine(«Debut BigWork»);
            await TaskEx.Delay(2000);
            Console.WriteLine(«Fin BigWork»);
            texte += « Abonnez vous! :-»»;
        }

        private void Go()
        {
            BigWork();
            for (int i = 1; i < 10; i++)
            {
                Console.WriteLine(texte);
                Thread.Sleep(500);
            }
        }

        public static void Main(string[] args)
        {
            Program p = new Program();
            p.Go();
        }
    }
}

```



Une myriade d'avions en papier animés avec Async

Pour compiler ce code, il est nécessaire d'ajouter AsynCtpLibrary.dll dans les références du projet. La première chose qui frappe avec ce code est que sa structure est identique au code synchrone précédent. Le code lui-même est quasiment identique. Examinons la méthode BigWork de près. Elle est déclarée async, ce qui ne veut pas dire qu'elle devient asynchrone en tant que telle. Il est ici important de comprendre qu'en dépit de l'appellation async, une telle méthode n'est jamais lancée dans son entier dans un thread séparé. Mais cette déclaration est indispensable pour que l'usage du

mot-clé `await` devienne légal dans le corps de la méthode et que le compilateur puisse la «travailler» comme nous allons le voir. Nous avons donc :

```
await TaskEx.Delay(2000);
```

Nous voyons que le `Thread.Sleep` du code synchrone est remplacé par `TaskEx.Delay` car ce dernier retourne un objet `Task`. En effet toute méthode invoquée via `await` doit retourner un objet de type `Task<T>`. Ce type permet soit de s'enquérir si un travail est terminé, ce dont le système va se charger, soit de récupérer le résultat de type `T` du côté du programmeur, possibilité que cet exemple n'exploite pas. Il est maintenant important de bien comprendre ce qui se passe au niveau du flux d'exécution. Quand le flux entre dans le code de la méthode, nous sommes toujours en mode synchrone. Puis, lorsque la ligne comportant `await` est rencontrée, ce qui la suit est effectué en mode asynchrone, peut-être dans un thread, mais le programmeur n'a pas du tout à se préoccuper de cette question. *Await*, comme le nom l'indique signifie que le système va attendre la fin du travail lancé. Mais *await*, **comme son nom ne l'indique pas**, signifie également que notre méthode `BigWork` rend la main immédiatement. Ce qui concrètement signifie pour notre exemple que les lignes de code qui la suivent **ne sont pas exécutées**, et que le flux d'exécution retourne dans l'appelant, ici notre méthode `Go`. Nous savons que celle-ci symbolise l'interface utilisateur, qui revient donc à la vie, comme on le voit sur l'illustration [Fig.3].

5 LE TRAVAIL DU COMPILATEUR

Le comportement inattendu que nous venons de décrire est dû à tout le travail que le compilateur a effectué sur notre code. Ainsi il a ajouté d'un côté la gestion de l'attente du travail asynchrone lancée par `await`, et d'un autre côté, il a fabriqué une fonction de rappel à partir des lignes de codes qui suivent `await`. Lorsque `await` a terminé son attente, le système invoque automatiquement la fonction de rappel générée. Autrement dit, si nous avons bien une fonction de rappel comme il se doit en programmation asynchrone, nous n'avons pas eu besoin de nous soucier de l'écrire. Le compilateur s'en est chargé à partir de notre code écrit à la manière synchrone! C'est puissant. Bien sûr, une méthode peut comporter plusieurs appels à `await` dans son corps. Dans ce cas, le compilateur éclatera le flux d'exécution autant de fois que c'est nécessaire, suivant le principe expliqué.

6 CE QUE ASYNC N'EST PAS

Async, nous l'avons vu, permet d'obtenir un code asynchrone sans se préoccuper de l'écrire, mais il y a aussi un problème potentiel. Les habitués de la programmation concurrente auront tout de suite remarqué l'accès concurrent au membre 'texte' de la classe `SimpleAsync`. Non seulement le compilateur ne gère pas ce point, mais il interdit même de poser un verrou sur le membre concerné. Ainsi ce code est **illégal** et en tant que tel, il sera refusé par le compilateur :

```
Object mylock = new Object();
string texte = «Programmez!»;

public async void BigWork()
{
```

```
    lock (mylock);
    await TaskEx.Delay(2000);
}
```

Async permet donc de faire de la programmation asynchrone, mais pas de la programmation concurrente qui est une notion différente. Du moins ne le permet-il pas seul, car dans les exemples d'Async on trouve une solution au problème du dîner des philosophes qui est un classique des classiques de la programmation concurrente. Cet exemple est écrit avec Async certes, mais aussi avec TPL Dataflow (TDF) qui est une librairie dédiée à la construction d'applications concurrentes et qui sort du champ de cet article.

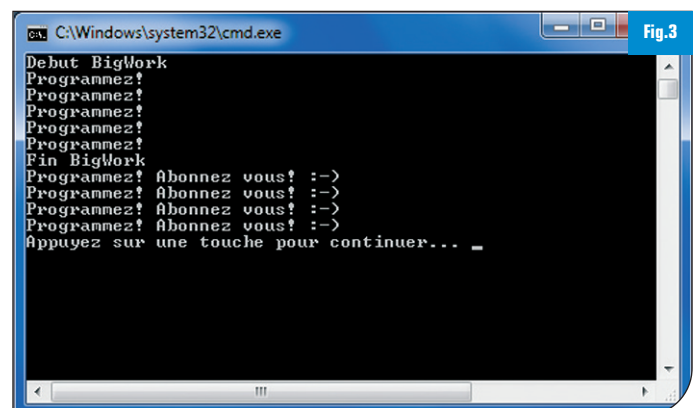
7 ASYNC ET LA PROGRAMMATION PARALLÈLE

Il est malgré tout possible de faire de la programmation parallèle avec Async. Voici un nouvel exemple dérivé du précédent. Nous lançons plusieurs travaux `BigWork` simultanément.

```
using System;
using System.Collections.Generic;
using System.Threading;
using System.Threading.Tasks;

namespace ParallelAsync
{
    class Program
    {
        private async Task<int> BigWork(int delay)
        {
            Console.WriteLine(«Debut BigWork {0}», delay);
            await TaskEx.Delay(2000);
            Console.WriteLine(«Fin BigWork {0}», delay);
            return delay*2;
        }

        private async void GoParallel()
        {
            Console.WriteLine(«Debut travail parallele»);
            List<Task<int>> asynclist = new List<Task<int>>();
            asynclist.Add(BigWork(2000));
            asynclist.Add(BigWork(3000));
            asynclist.Add(BigWork(4000));
            // Ici results est tout simplement de type Array
        }
    }
}
```



Notre premier exemple asynchrone en action.

```

var results = await TaskEx.WhenAll(asyncList);
Console.WriteLine("Fin travail parallele");
// Collecte les resultats
int somme = 0;
foreach (int result in results)
    somme += result;
Console.WriteLine(
    "Collection des resultats - Resultat: {0}", somme);
}

private void Go()
{
    GoParallel();
    for (int i = 1; i < 15; i++)
    {
        Console.WriteLine("Programmez!");
        Thread.Sleep(500);
    }
}

public static void Main(string[] args)
{
    Program p = new Program();
    p.Go();
}
}

```

Notre méthode `BigWork` est légèrement modifiée. Cette fois, elle retourne une valeur entière encapsulée dans un objet `Task`. Le type de retour est donc `Task<int>`. Ensuite, la méthode `GoParallel` place toutes ces `Task<int>` issues de `BigWork` dans une liste, puis un appel à `TaskEx.WhenAll` permet d'attendre la terminaison du total, sachant que, tout comme dans l'exemple précédent, `GoParallel` rend immédiatement la main à son appelant, ici `Go`, qui poursuit son exécution. Le restant de `GoParallel`, là encore, est transformé en une fonction de rappel qui sera automatiquement invoquée lorsque l'attente `TaskEx.WhenAll` sera terminée. L'illustration montre cet exemple en action [Fig.4]. Le lecteur trouvera sur notre site l'exemple complet `ParallelAsyncBis` qui est une variable du précédent. En voici un extrait :

```

private int BigWork(int delay)
{
    Console.WriteLine("Debut BigWork {0}", delay);
    Thread.Sleep(2000);
    Console.WriteLine("Fin BigWork {0}", delay);
    return delay * 2;
}

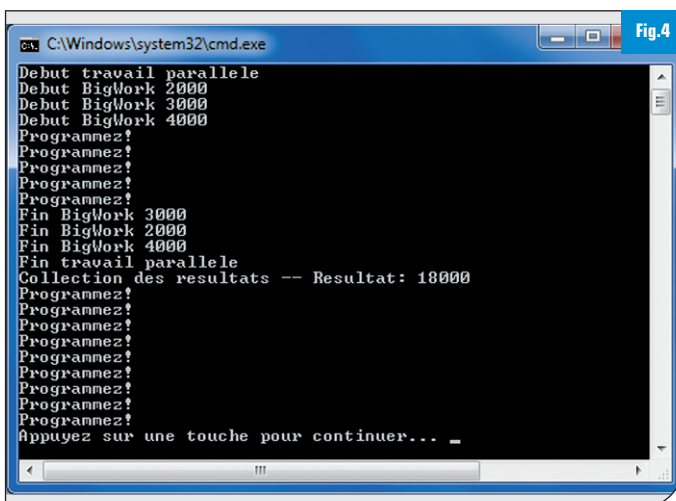
private async void GoParallel()
{
    Console.WriteLine("Debut travail parallele");
    List<Task<int>> asyncList = new List<Task<int>>();
    asyncList.Add(TaskEx.Run(() => BigWork(2000)));
    asyncList.Add(TaskEx.Run(() => BigWork(3000)));
    asyncList.Add(TaskEx.Run(() => BigWork(4000)));
    // Ici results est tout simplement de type Array
    var results = await TaskEx.WhenAll(asyncList);
    Console.WriteLine("Fin travail parallele");
    // Collect les resultats
    int somme = 0;
    foreach (int result in results)
        somme += result;
    Console.WriteLine(
        "Collection des resultats - Resultat: {0}", somme);
}

```

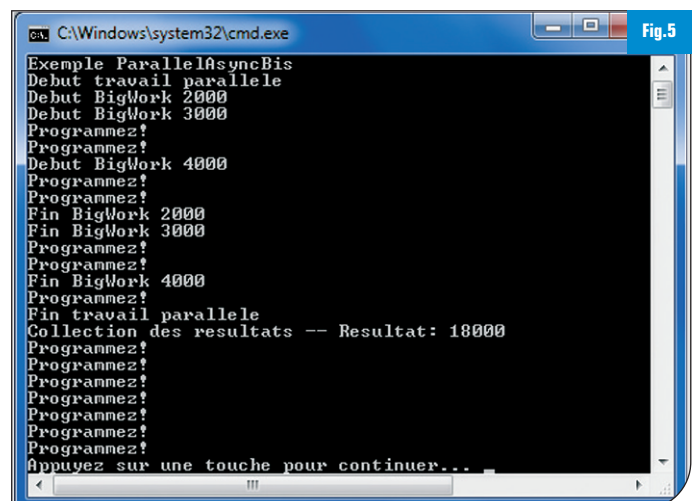
Ici nous avons écrit `BigWork` sans passer par `TaskEx.Delay`, mais par `Thread.Delay` comme il est plus instinctif de le faire. Nous savons que le problème avec `Thread.Sleep` est l'absence d'un type de retour adapté. Nous réglons ce problème avec `Task.Run` et une fonction anonyme :

```
asyncList.Add(TaskEx.Run(() => BigWork(2000)));
```

Ce moyen permet, si besoin, de faciliter la modification d'un ancien code synchrone en code asynchrone. L'ordre d'exécution peut s'avérer toutefois différent de ce que l'on a obtenu avec l'exemple précédent, ainsi que l'illustre la [Fig.5], ce qui toutefois n'est pas gênant (ou ne doit pas l'être), car, par définition, un ordre d'exécution ne peut être garanti en programmation asynchrone ou parallèle.



Async est capable de parallélisme



Qui dit parallélisme et asynchronicité dit ordre d'exécution non garanti :-)

8 ASYNC ET LES EXCEPTIONS

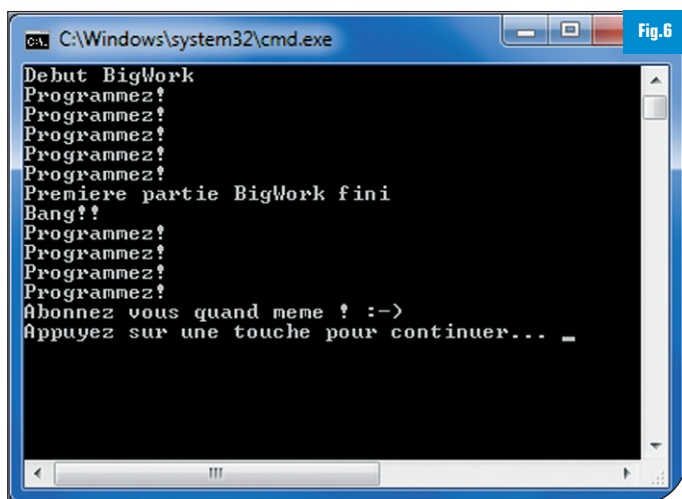
Async permet le travail avec les exceptions et normalement propage vers l'appelant les éventuelles exceptions survenues dans une méthode async et non capturées au sein de celles-ci. Les spécifications, sauf incompréhension de notre part, sont claires à ce sujet : *«If the return type of the asynchronous function is void, nothing is returned to the caller. If execution of the asynchronous method completes successfully, nothing happens. If an exception occurs and is not caught within the asynchronous method, the exception is thrown to whichever method last invoked or resumed it.»* Cependant l'exemple ci-dessous et la [Fig.6] montrent que ceci ne fonctionne pas comme annoncé, du moins avec notre version CTP d'Async :

```
using System;
using System.IO;
using System.Threading;
using System.Threading.Tasks;

namespace AsyncException
{
    class Program
    {
        string texte = «Programmez!»;

        private async void BigWork()
        {
            Console.WriteLine(«Debut BigWork»);
            await TaskEx.Delay(2000);
            Console.WriteLine(«Premiere partie BigWork fini»);
            Console.WriteLine(«Bang!!»);
            await TaskEx.Run(() => (File.OpenRead(null)));
            texte += « Abonnez vous ! :-»);
        }

        private void Go()
        {
            try
            {
                BigWork();
                for (int i = 1; i < 10; i++)
            }
        }
    }
}
```



Oups ! Async CTP ne se comporte pas comme annoncé

```
{
    Console.WriteLine(texte);
    Thread.Sleep(500);
}
}
catch(Exception ex)
{
    Console.WriteLine(«Exception: {0}», ex.Message);
}
finally
{
    Console.WriteLine(«Abonnez vous quand meme ! :-»);
}
}

public static void Main(string[] args)
{
    Program p = new Program();
    p.Go();
}
}
```

Dans cet exemple, l'exception levée lors de la tentative d'ouverture d'un fichier ne remonte pas à l'appelant Go qui continue son exécution normalement. Cette exécution se termine comme il se doit par le bloc finally, mais le gestionnaire d'exception catch n'est jamais exécuté. En clair, l'exception est passée à la trappe! Par contre si une exception est interceptée dans le corps d'une méthode async, tout fonctionne comme prévu. Nous renvoyons le lecteur à l'exemple AsyncExceptionBis sur notre site, et dont voici l'extrait intéressant :

```
private async void BigWork()
{
    try
    {
        Console.WriteLine(«Debut BigWork»);
        await TaskEx.Delay(2000);
        Console.WriteLine(«Premiere partie BigWork fini»);
        Console.WriteLine(«Bang!!»);
        await TaskEx.Run(() => (File.OpenRead(null)));
        texte += « Abonnez vous ! :-»);
    }
}
catch (Exception ex)
{
    Console.WriteLine(«Exception: {0}», ex.Message);
    Console.WriteLine(«Abonnez vous quand meme ! :-»);
}
}
```

9 UNE APPLICATION FENÊTRÉE QUI RESTE RÉACTIVE

Construisons pour le plaisir, une application fenêtrée qui restera réactive lors du téléchargement d'un minimag sur le site de Programmez. Notre fenêtre comporte un slider que nous voulons pouvoir actionner à tout moment. Commençons par écrire le code de façon synchrone (Cet exemple BadUI est complet sur notre site :

```
private void DownloadMinimag()
{
    label1.Content = «Début téléchargement»;
    button1.IsEnabled = false;
    WebClient wc = new WebClient();
    // ceci est horrible
    wc.DownloadData(«http://www.programmez.com/demoasync/mini_98.pdf»);
    // ceci est pire :-
    // wc.DownloadDataAsync(new Uri(«http://www.programmez.com/demoasync/mini_98.pdf»));
    label1.Content = «Téléchargement terminé»;
    button1.IsEnabled = true;
}

private void button1_Click(object sender, RoutedEventArgs e)
{
    DownloadMinimag();
}
```

DownloadData étant synchrone, notre interface reste bloquée tout le temps de son action. Modifions maintenant notre code (GoodAsyncUI est disponible au complet sur notre site)

```
byte[] minimag;

private async void button1_Click(object sender, RoutedEventArgs e)
{
    minimag = null;
    minimag = await DownloadMinimag();
}

private async Task<byte[]> DownloadMinimag()
{
    label1.Content = «Début téléchargement»;
    button1.IsEnabled = false;
    button2.IsEnabled = false;
    WebClient wc = new WebClient();
    var data = await wc.DownloadDataTaskAsync(
        new Uri(«http://www.programmez.com/demoasync/mini_98.pdf»));
    label1.Content = «Téléchargement terminé»;
    button1.IsEnabled = true;
    button2.IsEnabled = true;
    return data;
}
```

```
private void button2_Click(object sender, RoutedEventArgs e)
{
    if (minimag != null)
        label1.Content = «Abonnez-vous! :-»);
}
```

La méthode DownloadMinimag est maintenant déclarée async et son type de retour est Task<byte[]>. Autrement dit, à l'issue du téléchargement le minimag sera placé dans un tableau de byte. Ensuite, bien remarquer l'invocation de la méthode DownloadDataTaskAsync de WebClient et ne pas la confondre avec la méthode habituelle DownloadDataAsync qui ne peut être utilisée car elle ne retourne pas de type Task<T>. DownloadDataTaskAsync a été ajoutée lors de l'installation de Async. Mais nous aurions pu aussi tout simplement utiliser DownloadData conjointement avec TaskEx.Run, comme ceci :

```
var data = await TaskEx.Run(
    () => wc.DownloadData(
        new Uri(«http://www.programmez.com/demoasync/mini_98.pdf»)));
```

On utilisera d'ailleurs cette approche, chaque fois qu'Async ne fournit pas de méthode adaptée. Enfin il est important de remarquer que le gestionnaire d'événement du bouton qui lance le téléchargement est lui aussi déclaré async, bien que rien d'asynchrone n'y soit fait. Simplement le compilateur exige que toute méthode qui invoque await soit déclarée async. Le gestionnaire du bouton 2 n'est là que pour vous permettre de placer un point d'arrêt afin de constater sous débogueur que le document a bien été téléchargé. En effet, pour l'instant il n'est pas possible de déboguer les méthodes async.

10 CONCLUSION

Async se présente comme une nouvelle façon très séduisante d'écrire du code asynchrone. Cette version CTP n'est pas exempte de problèmes qui seront sans doute, au moins pour certains d'entre eux, corrigés par la suite. Nous avons vu que la propagation des exceptions nous réserve des surprises. Par ailleurs, il n'y a pas de résolution de surcharge pour les méthodes async et il convient de faire attention à l'ordre d'évaluation des fonctions. Ainsi :

```
f() + await g()
```

verra g() évalué avant f(). Malgré ces petits soucis, le confort de programmation qu'Async apporte est indéniable, et il fera très probablement partie intégrante de .NET 5. C'est du moins ce que projette Microsoft au moment de la rédaction de cet article.

■ Frédéric Mazué - fmazue@programmez.com

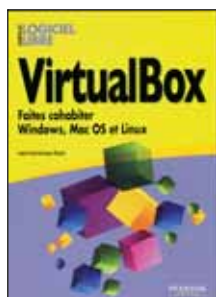
L'information permanente

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

www.programmez.com

VIRTUALISATION

VirtualBox



Difficulté : *
 Editeur : Pearson
 Auteur :
 Henri-Dominique Rapin
 Prix : 20 €

Très accessible, cet ouvrage met à la portée de tous VirtualBox, l'une des solutions de virtualisation les plus populaires du moment, car conviviale, performante... Il vous explique pas à pas comment virtualiser d'anciens systèmes d'exploitation, faire tourner des logiciels incompatibles, tester des programmes ou des utilitaires sans compromettre votre ordinateur, ou encore archiver vos applications et protéger vos données dans des machines virtuelles. Vous apprendrez à maîtriser les principales fonctionnalités de VirtualBox, et deviendrez rapidement un incontournable de la virtualisation. Cet outil de virtualisation propose de puissantes fonctionnalités et c'est gratuit !



WEB

ASP.Net 4 avec C#

Difficulté : **
 Editeur : Editions Eni
 Auteur :
 Brice-Arnaud Guerin
 Prix : 39 €

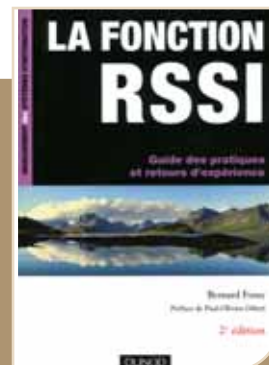
ASP.Net 4 offre de nouvelles possibilités dans le développement web et couplé à C# et surtout Visual Studio 2010, le développeur bénéficie de toutes les nouveautés et fonctions de la plate-forme .Net. Le lecteur commencera par Visual Studio et ses outils (refactoring, tests unitaires, tests d'interface graphique,...) et par les évolutions du langage C#. Le deuxième chapitre décrit le fonctionnement des applications IIS. L'ouvrage étudie en détail les Web forms, AJAX, les sites MVC et propose des composants personnalisés pour créer des graphiques. Les chapitres suivants élaborent des solutions pour allier rapidité de développement et performances dans l'accès aux bases de données ADO.NET.

LIVRE DU MOIS

La fonction RSSI 2e édition

Difficulté : *** - Editeur : Dunod
 Auteur : Bernad Foray - Prix : 37 €

La sécurité informatique ne s'invente pas, ne s'improvise pas ! Ce n'est pas un hasard si la fonction du responsable de la sécurité des systèmes d'information (RSSI) devient centrale dans certaines entreprises. Mais le RSSI n'est pas une fin en soi. Car au-delà du poste, il y a toute une politique, une stratégie, une sensibilisation de la sécurité, des procédures de sécurité à auditer, à adopter par tous les utilisateurs, les développeurs. Malheureusement, aujourd'hui, encore trop peu de logiciels utilisent un code sécurisé ! Cet ouvrage aborde longuement les problématiques RSSI autour de 4 thèmes : préparation, fondamentaux, activités opérationnelles et moyens de contrôles. « Qui veut la paix, prépare la guerre » ! Si vous vous préoccupez de la sécurité après une attaque, il est déjà trop tard. Cette deuxième édition s'enrichit de mises à jour sur la continuité de service, la sensibilisation à la sécurité ou encore la conformité informatique et libertés. Ces nouveaux chapitres s'attachent plus particulièrement aux aspects organisationnels. A lire absolument !



FUTUR

Préserver son patrimoine numérique



Difficulté : *
 Editeur : Eyrolles
 Auteur : Claude Huc
 Prix : 19,90 €

Aujourd'hui, et ce depuis 15 ans, nous stockons à tout va de multiples données et la volumétrie ne cesse d'exploser. Mais beaucoup d'utilisateurs ne se soucient pas de ce patrimoine qui tombe parfois rapidement en désuétude et devient même inaccessible : format inconnu ou trop vieux, fichiers corrompus, incompatibilités des systèmes, etc. Imaginez des codes sources vieux de 10 ans en Java, ils ne vont plus fonctionner sans modification, ni des fichiers comptables de 1994 ! La préservation du patrimoine numérique devient donc une véritable question ! La question du format est importante mais pas l'unique problème car finalement le format n'est que l'arbre qui cache la forêt. Comment préserver l'archivage de ses mails ? Nous avons un historique de 20 Go de mails représentant plus de 600 000 messages sur 7 à 8 ans ! Mais demain, que faire si on change de système ? Un ouvrage instructif qui a le

mérite de poser les questions, mais c'est à l'utilisateur de faire les démarches et d'être pro-actif !

WEB

HTML, XHTML et CSS pour les Nuls 2e édition



Difficulté : *
 Editeur : First Edition
 Auteur : collectif
 Prix : 24,90 €

Nul besoin d'être un maître de la programmation pour créer de belles pages Web : ce livre vous montre à quoi HTML peut servir et comment utiliser XHTML pour créer des pages esthétiques. Découvrez ensuite comment CSS peut vous aider à manipuler les couleurs, les polices et plus encore. Une belle manière de débiter dans le développement web si vous n'avez pas encore tâté la bête ! Vous y trouverez tous les fondamentaux pour démarrer rapidement un site et coder les principales fonctions. Pour cette 2e édition, les auteurs abordent mieux le web mobile et surtout on dispose de deux chapitres pour HTML 5 (personne n'y échappe) et pour CSS3 ! Ces nouveautés permettent d'être à jour dans les technologies et ses connaissances.



ABONNEMENT PDF

soit 2,73 € le numéro

30 € par an

www.programmez.com

solutions **LiNux** **Open Source**



Le salon européen dédié à Linux
et aux logiciels libres

Toutes les solutions et nouveautés en Open Source...
Pour encore plus de libre au service de l'entreprise !

**NOUVEAU
LIEU
NOUVELLES
DATES**

**10-11-12
MAI
2011**

CNIT - Paris La Défense



EXTRAIT DU PROGRAMME DES CONFÉRENCES :

- ERP, CRM, BI : les solutions open source à déployer en entreprise ?
- Dans les nuages et au-delà : un ciel dégagé pour les logiciels libres ?
- Gouvernance de SI et Logiciel Libre : méthodes, outils, retours d'expérience
- Android, succès, limite et avenir du système d'exploitation mobile
- Quelle place pour l'Open Source dans l'e-Commerce de demain ?
- Quelle politique publique en matière de logiciel libre ?
- 2010 l'année des forks : et l'avenir ? Quelles sont les meilleures pratiques pour pérenniser une communauté ?
- Open source et réseaux sociaux d'entreprise: convergences et enjeux ?
- HTML5, Guerre des navigateurs, convergence Web/mobile : quel affichage pour le web de demain ?

Pour toute demande d'information : info@solutionslinux.fr
Demandez votre badge d'accès gratuit sur www.solutionslinux.fr

Un événement
Tarsus
un événement

Partenaire officiel
**monannuale
pro.com**

.NET, PHP, Java...

Nous avons chacun notre langage, mais pensons tous Windows Azure.

.Net, PHP, Java... Grâce à la plateforme applicative Windows Azure, vous déployez et exécutez vos applications dans le Cloud quel que soit le langage ou l'outil de développement. Et disposez d'un environnement de développement et de production à la demande, en quelques minutes.

Avec Windows Azure, le code vous appartient, pour le reste comptez sur nous.

C'est ça la puissance du Cloud.

**Pensez Windows Azure.
Essayez la plateforme Windows Azure gratuitement⁽¹⁾
sur www.windowsazure.fr**

⁽¹⁾1 mois d'utilisation offert. Offre valable jusqu'au 13 mai 2011 minuit.

