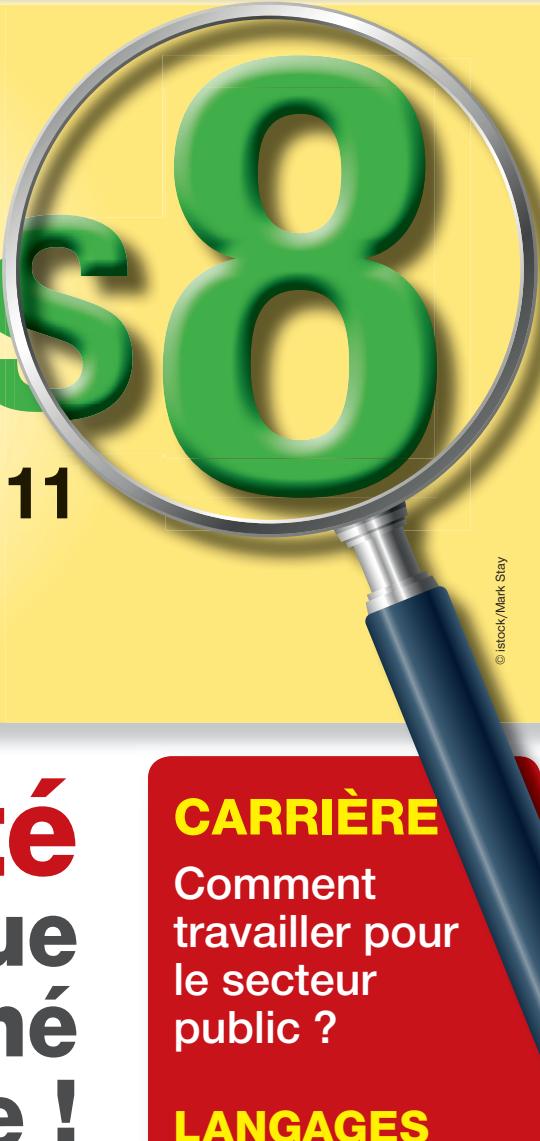




Inside Windows 8

WinRT, Metro, Visual Studio 11
Ma 1^{re} application Win8 ...

Tout ce qui va changer



© istock/DutchIcon

La sécurité informatique n'a pas gagné la guerre !

Botnet, forensique, ANSSI, malwares :
les menaces légales et illégales se multiplient



AR. Drone

Un OVNI programmable !



CARRIÈRE

Comment travailler pour le secteur public ?

LANGAGES

Redécouvrir la programmation fonctionnelle

GAMING

XNA : le dev tool à tout faire

M 04319 - 146 - 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

OPÉRATION 1 PC POUR 1 EURO DE PLUS

ACHETEZ WINDEV 17 ET RECEVEZ
UN PC DELL OU DEUX TABLETTES
SAMSUNG POUR 1 EURO DE PLUS

RENDEZ-VOUS SUR PCSOFT.FR

Offre réservée aux entreprises, administrations, collectivités, indépendants, GIE, associations,... en France métropolitaine. Chaque élément de l'offre peut être acheté séparément. Tous les détails sont sur www.pcsoft.fr.

DELL



OU

DELL



x2

Samsung GALAXY Tab 10.1



OU

CALENDRIER

Montpellier	8 Nov
Toulouse	15 Nov
Bordeaux	16 Nov
Nantes	17 Nov
Paris	22 Nov
Lille	23 Nov
Bruxelles	24 Nov
Strasbourg	29 Nov
Genève	30 Nov
Lyon	1 Déc
Marseille	6 Déc

TOUR DE FRANCE



GRATUIT

inscrivez-vous sur www.pcsoft.fr

Attention: 10.000 places seulement

TOUR DE FRANCE
VENEZ DÉCOUVRIR
WINDEV 17
& **WEBDEV 17**
PRÈS DE CHEZ VOUS

Fournisseur Officiel de la
Préparation Olympique

www.pcsoft.fr

JavaOne 2011 : Java 8, pas avant 2013

Du 2 au 6 octobre dernier, Oracle organisait la traditionnelle JavaOne, la grande conférence des développeurs Java. Contrairement aux années précédentes, l'édition 2011 fut un bon cru avec (enfin) une vision claire de la stratégie sur et autour de Java, et des grandes évolutions d'ici 2013.

2011 clôt les nombreuses polémiques qui avaient cours depuis le rachat de Sun et les attitudes changeantes envers la communauté. Au moins, nous ne pourrons pas dire qu'Oracle n'essaie pas de faire avancer Java et la JCP.

Un des points importants de cette JavaOne 2011 a été l'agenda 2011 - 2013 autour de Java et de NetBeans.

Un agenda 2011-2013 plus clair

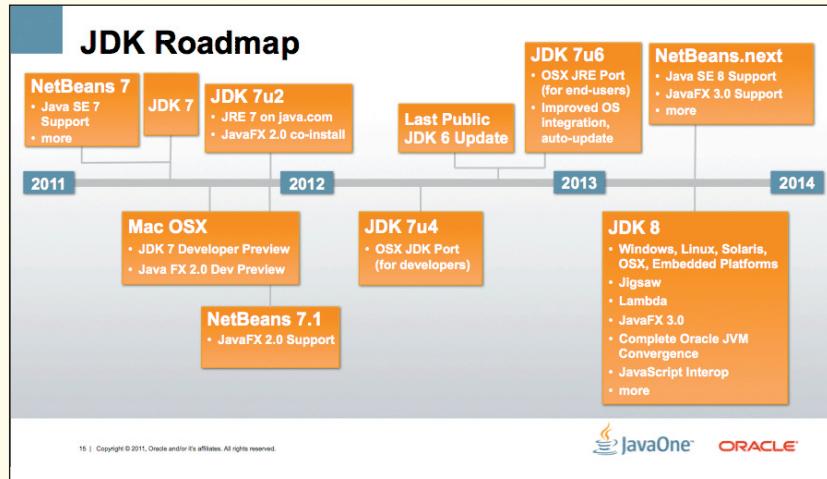
La roadmap révèle que Java 8 ne sera pas disponible avant l'été 2013, initialement, la v8 devait sortir au 2e semestre 2012 mais des retards dans le développement et dans plusieurs modules annexes obligent à décaler de 6-7 mois, minimum. Ce n'est pas une mauvaise nouvelle.

Un rythme de 18-24 mois entre chaque version majeure devrait être la norme désormais, laissant le temps à la communauté d'adopter chaque version.

Car il ne faut pas oublier que l'adoption d'une version majeure d'un langage prend du temps.

Entre temps, la JDK 7 aura droit à plusieurs mises à jour (u2 à u6). Surtout, Oracle annonce une version OS X de la JDK 7 lors de la sortie de l'u4 vers le printemps ou l'été 2012.

NetBeans ne sera pas abandonné, loin de là ! L'IDE évolue régulièrement et chaque année, une version majeure est annoncée. La prochaine sera la v7.1 incluant le support de JavaFX 2.0. Pour 2012, aucune annonce n'a pour le moment été faite. Une version 7.x devrait voir le jour, seule une NetBeans.next est planifiée pour 2013 avec un support de Java 8, JavaFX 3.



Java 8 : une évolution en douceur

La modularité demeure un des axes centraux de Java 8. Il s'agit de rendre la JDK plus flexible et plus performante dans la montée en charge et le démarrage. Deux éléments renforcent cette approche : les projets Coin et Lambda. Si la v7 supportait partiellement la modularité, Oracle veut aller plus loin. Autre projet à suivre : Jigsaw. Pour rappel, ce projet doit apporter une modularité de la plateforme, assurer la montée en charge, le chargement incrémental, la suppression du classpath, une approche multi-module, etc. Oracle va aussi en profiter pour unifier le modèle de JVM en rapprochant la JVM Java et JRockit (issu du rachat de BEA). La prochaine JVM aura des nouveautés importantes comme l'incorporation du JavaScript (projet Nashorn). A cela se rajoutent : JavaFX 3, le support du tactile, des terminaux mobiles, des annotations améliorées, de nouvelles API.

Une arrivée sur OS X, plus de mobilité !

Oracle annonce l'arrivée pour 2012 de la JDK 7 sur OS X, une préversion est disponible officiellement. Jusqu'à présent, la plateforme Mac était en retard sur Java mais depuis l'arrivée d'Apple dans la communauté, les choses bougent assez rapidement. Surtout, Java veut être présent sur les terminaux mobiles de type smartphone. Pour

le moment, Java est très en retard sur ces matériels. Il s'agira tout d'abord de mieux faire travailler Java et HTML 5 et d'être présent sur iOS, Android. Le projet Avatar doit permettre de créer ces applications hybrides Java / HTML 5. A terme, on peut penser que Java SE et ME vont ne faire plus qu'un. Ce qui serait une bonne nouvelle et simplifiera le travail du développeur.

Java EE n'a pas été oublié. L'idée serait de poursuivre les fondations posées par JEE 7, notamment pour pousser JEE sur le cloud et particulièrement sur le PaaS (partie plate forme). JEE 7 subira courant 2012 plusieurs changements : gestion de cache entièrement nouvelle (JSR 107), support de HTML 5, JSON, au-delà, ce sera l'arrivée de l'architecture NoSQL.

D'autre part, JavaFX 2.0 est annoncé en open source. Pour rappel, JavaFX 2.0 introduit les API Java, un langage dédié à l'interface, l'accélération matérielle, un conteneur web intégré. Pour en savoir plus sur FX 2 : http://blogs.oracle.com/javaone/entry/javafx_2_0_arives_and

Cette JavaOne a été aussi l'occasion de parler des évolutions de la gouvernance de Java via la JCP. Un des objectifs est d'être plus transparent. Le recrutement des experts sera public. Le processus de tests TCK sera lui aussi plus transparent et accessible au public. Par le passé, l'opacité et la main mise de quelques éditeurs étaient régulièrement critiquées.



Syncfusion Essential Studio Enterprise

à partir de € 1,468



Interface utilisateur, rapports et composants BI dans un ensemble unique.

- Tout nouveau studio pour HTML 5, développement mobile multi-plateforme avec ASP.NET MVC
- Prise en charge des périphériques Windows Phone 7, Android, iPhone, iPad et BlackBerry
- Un seul code pour plusieurs plateformes mobiles
- Diagrammes, menus, boîtes de dialogue, barres de progression, boutons, listes et commandes d'évaluation, d'onglets et de curseurs, et bien plus encore



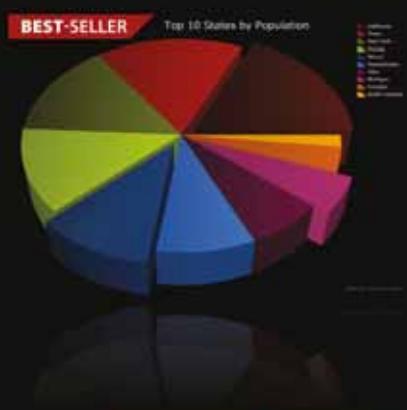
Spread.NET

à partir de € 882



Ajoutez des feuilles de calcul compatibles Excel aux WinForms et apps ASP.NET.

- 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
- Fonctionnalité de visualisation de données dont Sparklines et Camera Shapes
- Amélioration de jusqu'à 50 % des performances import/export d'Excel
- Créez des graphiques 2D et 3D complets dans vos feuilles de calcul



Codejock Xtreme Chart Pro ActiveX

à partir de € 169



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



DXperience Enterprise

à partir de € 957



Tous les outils DevExpress ASP.NET, WinForms, Silverlight, WPF et IDE Productivity en un.

- Abonnement de 12 mois pour tous les produits et mises à jour Developer Express et accès aux versions bêta en développement actif
- Composants et outils : grilles, entrée de données, outils d'écriture de code, analyse de données, graphiques, navigation/disposition, planification, solutions reporting, bibliothèques d'impression, outils de remaniement, bibliothèques ORM

\ le fait du mois

Java One 2011 : Java 8, pas avant 2013 3

\ actus

En bref 10

Hardware 12

\ données

SQL Server « Denali » : le projet Juneau 14

\ gaming**L'AR. Drone, un OVNI pour geek** (1re partie) 22**\ sécurité****Sécurité****Mais que fait le développeur ?!**

Comment devenir un développeur spécialiste en sécurité ? 27

La France dans la cyber-guerre 30

Du forensique à l'Epitech : l'autre face de la lutte contre le cyber-crime 32

Au cœur de l'Epitech Security Laboratory 33

Les botnets 35

Sécurité avancée avec PHP 38

\ dossier**Windows 8 : ce qui va changer pour le développeur**

Désactiver l'interface Metro et retrouver

le menu démarrer classique avec Windows 8 42

(Au cœur de) Windows 8 (1re partie) 43

(Au cœur de) WinRT (2e partie) 46

Ma première application Metro 49

Les nouveautés de Team Foundation Server 11 55

Les nouveautés de Visual Studio 2011 57

\ carrière

Travailler avec les administrations et collectivités 59

Ces entreprises qui recrutent 61

\ technique

Git : boostez votre gestion de configuration 62

\ code

Devenir un développeur iOS 65

XNA sur tous les fronts 69

LISP : hors du temps 74

Haskell : élégance, pureté, beauté 76

Caml : méconnu mais brillant 80

\ temps libre

Les livres du mois 82

Agenda 82



11



22



26



40



65



69

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.

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. Belkhayat-Fuchs, S. Saurel. Experts :

G. Renard, D. Caro, N. Freier, C. Villeneuve,

A. Verla, J. Antoine, E. Margraff, F. Santin,

A. Auroux, V. Fabing, S. Cordonnier, D. Catuhe,

A. Alnijres, Y. Grandmontagne.

Illustrations couverture : © istock/DutchIcon,

© istock/Mark Stay, Parrot

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

Ce numéro comporte un encart jeté Component Source sur 1 partie du tirage

Abonnement : Programmez, 17, Chemin des Boulanger, 78926 Yvelines Cedex 9 - 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 exclusive sur www.programmez.com

L'INFO PERMANENTE [WWW.PROGRAMMEZ.COM](http://www.programmez.com)



PROCHAIN NUMÉRO

N°147 décembre 2011

parution 30 novembre

✓ **Android 4.0**

Le premier système Google pour téléphone et tablette

✓ **Mobilité**

Développez des applications mobiles avec Adobe Flex

éditorial



©V.T.

A nos héros, l'informatique reconnaissante

Un des informaticiens les plus influents depuis l'invention des premiers processeurs est mort durant les premiers jours d'octobre. Non, il ne s'agit pas de Steve Jobs qui eut droit à un hommage planétaire. Il s'agit de l'inventeur du langage C et co-concepteur du système Unix, excusez du peu. Ces deux inventions remontent au début des années 1970. Une éternité diront certains. Et pourtant, ces deux éléments fondent notre informatique : Unix s'est imposé car fondateur de nombreux systèmes et le langage C demeure l'un des plus utilisés, avec C++ et Java. Le C++ étant un C amélioré.

Vous n'avez toujours pas trouvé ? Il s'agit de Denis Ritchie (souvent appelé dmr). Ce nom ne vous dit certainement pas grand chose. Que serait notre informatique sans son travail sur les langages et les systèmes ? Personne ne peut le dire. C++ et les langages objets n'existeraient pas ou pas sous leur forme actuelle, pareillement pour le système d'exploitation. Denis reçut plusieurs récompenses prestigieuses : Turing Award (avec Thompson, co-développeur d'Unix) pour Unix, la médaille Hamming de l'IEEE, la médaille américaine de la technologie (avec Thompson) et le prix du Japon pour l'information et la communication, là encore, pour Unix. Il disparaît comme il mena sa vie, discrètement. Il travaillait aux laboratoires Bell. Avec sa petite équipe, il créa dans les années 1990, les systèmes Plan 9 et Inferno.

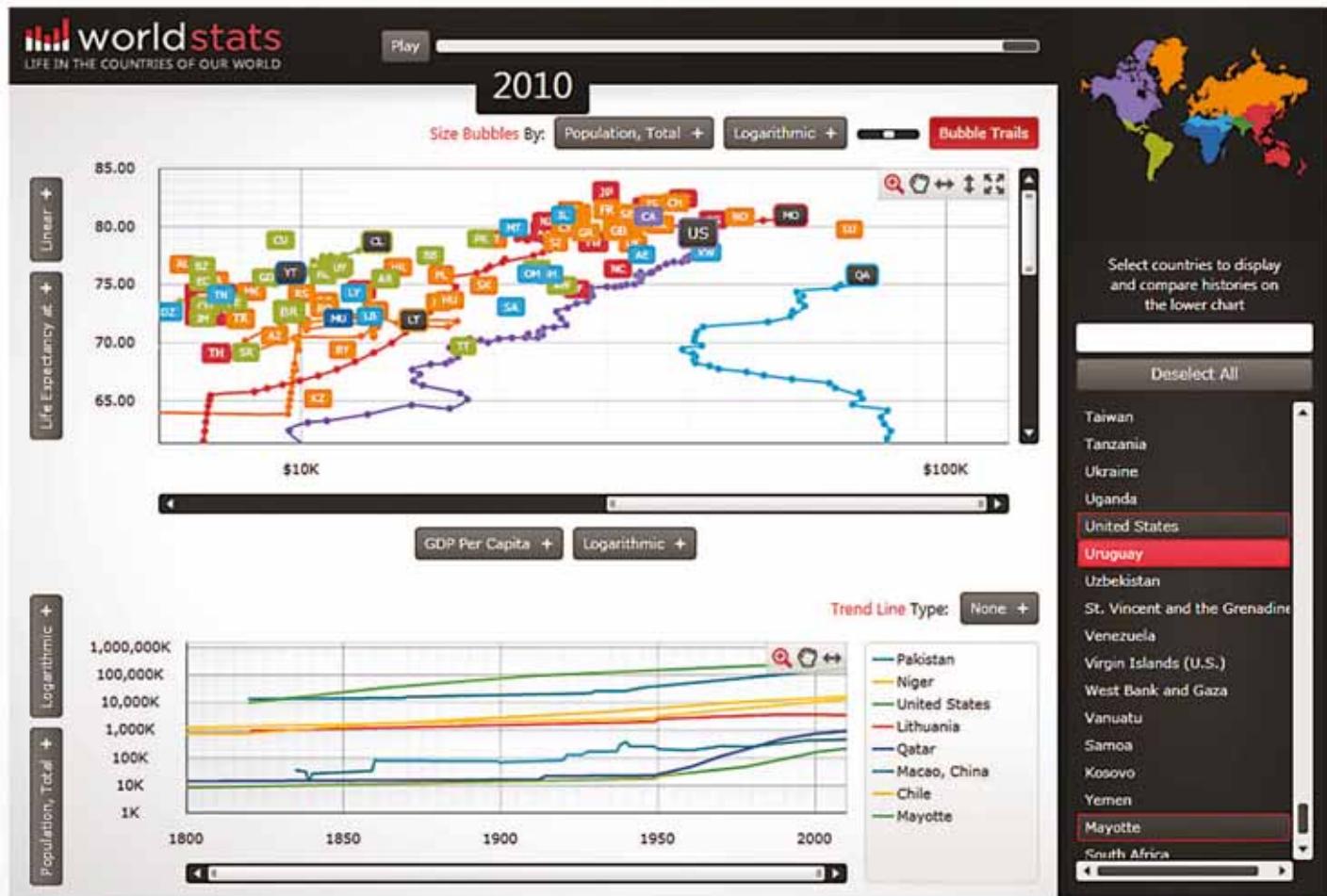
Il ne faut jamais oublier que derrière l'innovation informatique, il y a toujours des techniciens, développeurs, ingénieurs. Si aujourd'hui, nous utilisons Objective-C pour les terminaux iOS ou OS X, que nous utilisons Linux et OS X, que C++ existe, c'est parce qu'ils héritent de 30-40 ans d'informatique, mais les fondamentaux demeurent les mêmes. Et Denis Ritchie doit être vu comme un père fondateur de notre environnement informatique en 2011.

Impossible de ne pas finir cet éditorial 146 sans évoquer Steve Jobs. Tout a été dit ou presque. Son incroyable comeback en 1997 chez Apple pour amener la Pomme croquée là où l'on sait, est l'une des plus belles aventures industrielles. Même s'il ne fut pas lui-même un ingénieur, un développeur, il a su avoir la vision, avec les réussites et les échecs qui vont avec. NeXT fut un échec commercial mais une réussite technologique. Au départ Jobs était actif sur le projet Lisa mais mena le projet Macintosh, projet interne concurrent. Il fut convaincu de son approche après avoir vu les travaux du Xerox Parc.

Que l'on aime ou non Apple, force est de reconnaître que Jobs savait secouer les marchés, créer des marchés qui n'existaient pas. Peu de personnalités savent mener leurs troupes avec une telle énergie, vers un objectif commun. Sans doute l'esprit Apple. Et comme en clin d'œil, nous pourrions résumer ainsi notre fascination : 4S.

```
{  
printf ("rip\n");  
}
```

NetAdvantage® ULTIMATE



RAPPORTS, VISUALISATION DES DONNÉES ET
CONTRÔLES D'INTERFACE UTILISATEUR POUR ASP.NET,
WINDOWS FORMS, JQUERY/HTML5, WPF, SILVERLIGHT
ET WINDOWS PHONE 7



SCANNEZ ICI POUR
DECOUVRIR ULTIMATE!
www.infragistics.com/ultimate

FAITES PASSER VOS APPLICATIONS
AU NIVEAU SUPERIEUR
INFRAGISTICS.COM/ULTIMATE

INFRAGISTICS
DESIGN / DEVELOP / EXPERIENCE

Les Rapports DevExpress



The collage illustrates the following features:

- Left Screenshot:** A bar chart titled "Sales by Product" and a grid titled "Product Sales" showing data for various products like TV, Chair, Computer, etc.
- Middle Screenshot:** An "Employees List" report featuring a large photo of a woman, a pie chart, and a grid of employee profiles with details like Name, Position, Birth Date, etc.
- Right Screenshot:** The DevExpress Report Designer interface showing a report with a funnel chart, a report structure tree, and a preview of the report.



PRESENTATION CONTROLS | REPORTING CONTROLS
BUSINESS APP FRAMEWORKS | IDE PRODUCTIVITY TOOLS



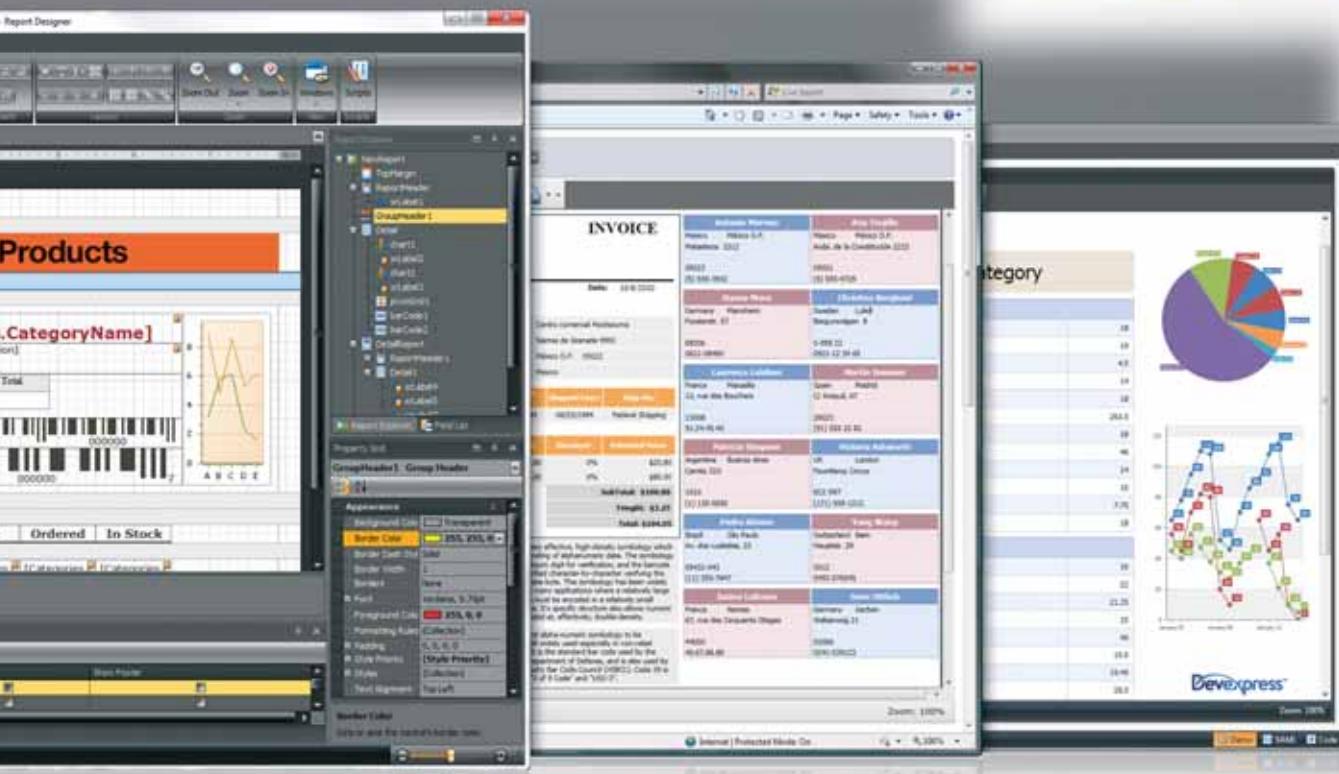
XtraReports Suite

Cross-Platform .NET Support

DevExpress crée des contrôles de présentation aux fonctionnalités complètes, des systèmes de rapports, des outils de productivité pour IDE et des frameworks d'applications Business pour Visual Studio .Net. Nos technologies vous aident à construire ce qu'il ya de meilleur, à avoir une vision plus claire des logiciels complexes, à améliorer votre productivité et à créer, dans le temps le plus court, des applications étonnantes pour Windows et pour le Web.

XtraReports Suite, de DevExpress, est une plateforme de Reporting de nouvelle génération pour Visual Studio. Elle vous permet de créer très rapidement des rapports professionnels, au pixel près, ciblant toutes les plateformes majeures Windows, incluant **WinForms**, **ASP.NET**, **Silverlight**, **WPF**. Au-delà de son ensemble de fonctionnalités de niveau professionnel, XtraReports Suite est livré avec un Designer de Rapports ergonomique pour l'utilisateur final, vous permettant de répondre avec la plus grande flexibilité aux exigences de Reporting.

Téléchargez aujourd'hui votre version gratuite d'évaluation et faites l'expérience de la Différence DevExpress.
www.DevExpress.com/Reporting



DevExpressTM

WWW.DEVEXPRESS.COM

■ **Google** annonce la disponibilité de MySQL en mode cloud pour compléter son offre PaaS avec App Engine. Ce dernier a du mal à décoller et l'éditeur tarde à compléter la plateforme. Aujourd'hui, l'arrivée de MySQL avec Google Cloud SQL, permet à App Engine de disposer d'un moteur relationnel comme on peut en trouver chez la concurrence. Ce service sera utilisable en Python et Java. Préversion limitée pour le moment.

■ Le **projet KDE** a dévoilé mi-octobre une nouvelle version de ownCloud, un service de stockage de données. Ce service est codé en PHP et doit respecter la vie privée, le contrôle des données par l'utilisateur. Il permet de stocker de la musique, images, contacts, calendrier, etc. ownCloud vous donne accès à vos données par un navigateur ou Web-Dav. Site : <http://owncloud.org/>

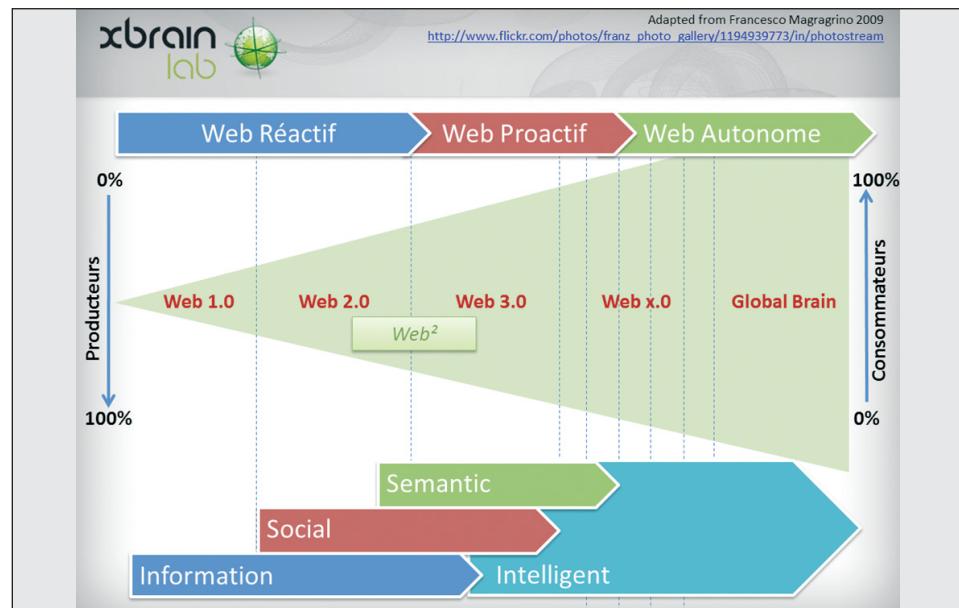


■ **Subversion 1.7** est sorti ! Cette version corrige et améliore de nombreuses fonctions : portefeuille unique .svn/, nouveau module d'administration de la commande svn, possibilité de changer le chemin dépôt... site : <http://subversion.apache.org/docs/release-notes/1.7.html>



■ **Intel** a sorti fin septembre dernier la version 1.5 de son SDK pour OpenCL. Ce kit doit permettre de développer, optimiser, tester du code pour l'architecture CPU – GPU, OpenCL. La v1.5 supporte Intel Advanced Vector Extensions, instructions 256-bit. Le compilateur introduit le « implicit CPU Vectorization Module ». Il s'agit d'une autovectorisation au niveau du compilateur. Site : <http://software.intel.com/en-us/articles/opencl-sdk/>

Prospective : l'avenir du web passera-t-il par le **proactif** ?



Le web a à peine plus de 22 ans. Le web 2.0 était très présent ces derniers temps, souvent avec des promesses qu'il ne pouvait pas réaliser. Une autre génération de web sera plus à même de concrétiser les attentes, les demandes des utilisateurs. Le web comme assistant personnel ? Est-ce réalisable ? Cette étape, baptisée par le plus grand nombre **Web Intelligent**, sera l'avènement d'un Web d'un nouveau genre : un "**Web Proactif**" ! L'ensemble des données, habitudes, achats, navigations, informations diverses (humeur, agenda, réseaux sociaux, etc.) vous concernant permettront à ce web d'anticiper vos attentes, d'anticiper vos besoins afin d'y apporter des réponses adaptées au bon moment, au bon endroit ! C'est cela le web proactif, une capacité à anticiper, à répondre à vos demandes. Imaginez quelques services bien concrets : vous quittez votre domicile, à Lille, le matin et le Web se charge de vous prévenir de prendre votre parapluie avec vous car il pleut à Paris, endroit où vous vous rendez ! ou vous êtes chez vous et le Web se charge de vous prévenir qu'une émission de télé va passer ce soir, susceptible de vous intéresser... par exemple votre artiste préféré dans une émission de variété. Nous pourrions résumer l'évolution du web ainsi :

- Le **Web 1.0** : Le Web de l'information dominé par Google

- Le **Web 2.0** : Le Web du Social et du Mobile dominé par Facebook et Apple/Google (Android).

- Le Web² [Square] : celui dans lequel nous évoluons chaque jour, le Web de la mise à disposition des données en temps réel avec un besoin crucial de structuration (cf. projet Linked Data de Tim Berners-Lee). Article fondateur du Web : <http://www.web2summit.com/web2009/public/schedule/detail/10194>

- Le **Web 3.0** (?) : qui reste pour le plus grand nombre encore lié à l'arrivée de la sémantique mais qui en définitive sera plus celui de l'émergence de l'intelligence du Web (d'une IA mélangée aux agrégats de données dans un contexte ubiquitaire et pervasif) ! Quels acteurs vont émerger sur ce « web » ? La technologie Siri d'Apple peut donner une petite idée de cette nouvelle interaction Homme - Machine où finalement, l'interface naturelle est la voix. Google avec son projet Guru ? Ou autre chose qui n'aurait pas été inventé ? Le « web 3.0 » pourrait être en réalité une agrégation de différentes technologies et approches : sémantique, microformat, recherche en langage naturel, gestion et extraction des données (avec l'explosion de la volumétrie des données c'est crucial), l'apprentissage de la machine, les agents de recommandation, la capacité aux logiciels, aux navigateurs, au Web globalement d'intégrer de l'intelligence artificielle.

Pour aller plus loin : <http://gregoryrenard.wordpress.com/2011/10/03/ere-du-web-proactif-web-intelligent-des-services-intelligents-sonnent-nos-portes/>

Et vous qu'en pensez-vous ?

Adobe sur un AIR de Flash

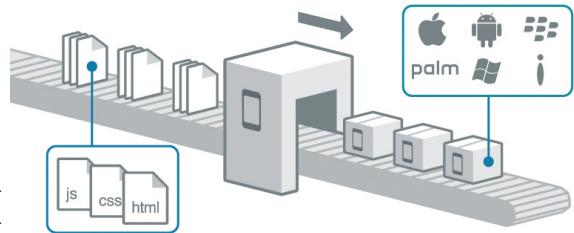
Leors de la grande conférence Adobe Max, Adobe a dévoilé plusieurs éléments clés de sa stratégie, tout particulièrement autour de Flash et de Air. Ainsi, Flash 11 arrive, accompagné de la v3 de AIR. L'éditeur insiste sur le caractère majeur de cette mouture, qui doit permettre aux éditeurs de diffuser des jeux 2D et 3D de qualité console. Les développeurs pourront tirer parti des nombreuses nouveautés et améliorations pour créer des jeux sur desktop, mobiles dont Android, iOS, grâce à AIR. Les extensions natives AIR sur iOS, les applications ont en outre accès à des fonctionnalités du terminal et des bibliothèques de code natif.

Retenons les nouveautés suivantes :

- Une accélération du rendu des graphismes 2D/3D : avec l'accélération matérielle, le rendu des images 2D et 3D est 1 000 fois plus rapide que dans Flash Player 10 et AIR 2. Les développeurs peuvent animer des millions d'objets à une

cadence de 60 images par seconde.

- Un moteur d'exécution captif : les développeurs peuvent automatiquement intégrer AIR 3 à leurs applications pour simplifier le processus d'installation sous Apple iOS mais aussi sous Android, Windows et Mac OS. Plus besoin donc pour les utilisateurs de télécharger et mettre AIR à niveau séparément sur l'une de ces plateformes, ou sur BlackBerry Tablet OS, qui intègre déjà le moteur d'exécution. Par ailleurs, avec l'option de runtime embarqué, les développeurs peuvent gérer les différentes versions de leur application, indépendamment des mises à niveau de AIR par Adobe.
- protection du contenu : le contenu vidéo peut désormais être protégé à l'aide d'Adobe Flash Access 3 sur les plateformes prises en charge
- Une diffusion multiplateforme de vidéos HD



- Une gestion des abonnements et des locations : grâce à la prise en charge d'Adobe Flash Access et Adobe Pass, les éditeurs de contenu peuvent tirer parti de formules d'abonnement et de location et proposer du contenu TVE (TV Everywhere) à plus de 80 % des abonnés aux chaînes payantes américaines.
- Une meilleure compatibilité : prise en charge du mode 64 bits sous Linux, Mac OS et Windows pour un résultat impeccable avec les tout derniers navigateurs 64 bits.

Adobe a aussi annoncé un renforcement de sa stratégie HTML 5 et multiplateforme mobile avec le projet Edge, l'outil HTML 5, et surtout le rachat de PhoneGap, une plateforme mobile open source.



Modeliosoft

Solutions d'entreprise basées sur l'atelier open source leader de modélisation
Modélisation et implémentation des Systèmes et Logiciels

Développeurs logiciels

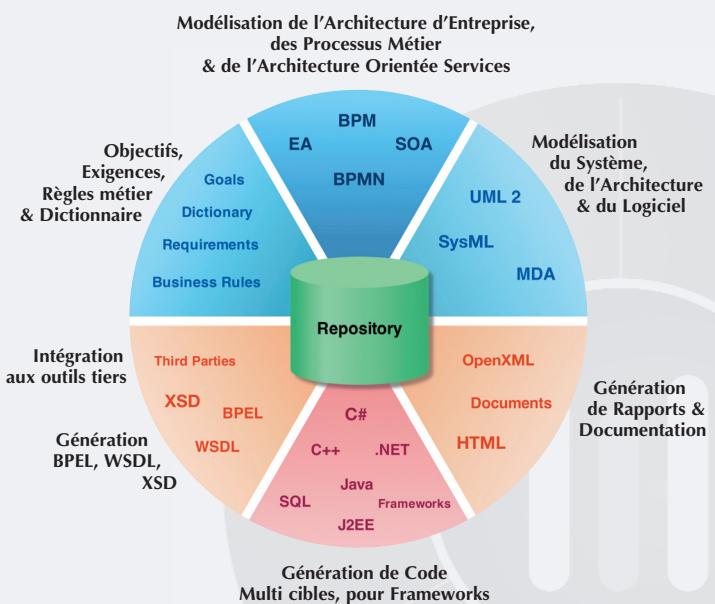
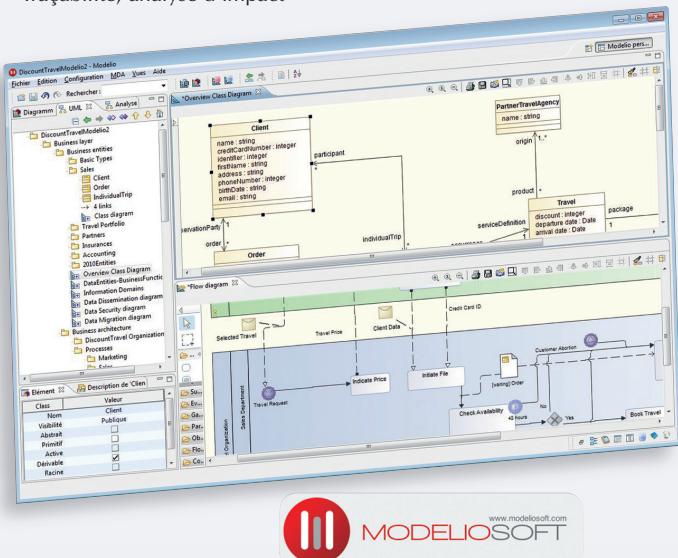
- UML pour Java, C++, C#, SQL, WSDL, XML
- Génération, roundtrip, reverse

Architectes métier

- Modélisation intégrée de UML, BPMN, Architecture d'Entreprise,
- Analyse intégrée des exigences, des objectifs, ...

Architectes système

- Modélisation Systèmes, Logiciels, Hardware, Exigences
- Traçabilité, analyse d'impact



Modelio.org

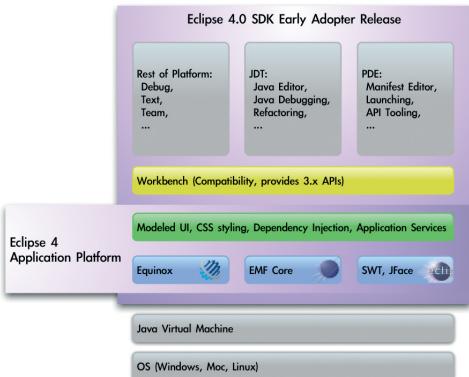
- L'atelier mature de modélisation UML/BPMN open source
- Le Store Modelio : modules et extensions pour modéliser et générer

Solutions d'entreprise Modeliosoft

- Solutions pour développeurs, architectes métier, architectes systèmes
- Génération documentaire Word, HTML : plans types dédiés
- Travail de groupe distribué, gestion de version et configuration

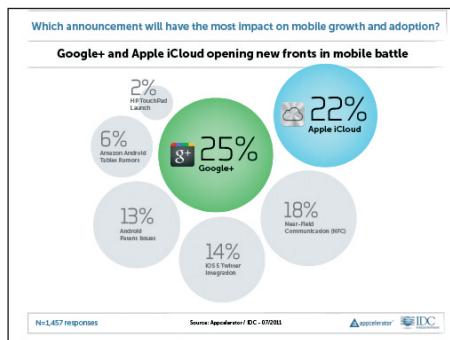
■ Le projet **LibreOffice** annonce qu'il sortira dans les prochains mois (dates non précisées) des versions mobiles et SaaS de la suite bureautique. LibreOffice a définitivement pris la place d'OpenOffice qui peine à rester à niveau.

■ La fondation Eclipse travaille toujours activement sur **Eclipse 4** et son incubateur e4. Cette version radicalement différente doit aider Eclipse à reposer sur de nouvelles fondations. L'environnement e4 s'appuie sur Eclipse 4.x SDK. Cette nouvelle génération de SDK deviendra le socle technique de



toute la plateforme Eclipse. Ce SDK devrait être intégré à la version finale à partir d'Eclipse Juno (prévue pour mai – juin 2012). Actuellement, nous sommes en version 3.7.1 (Eclipse Indigo).

■ La nouvelle norme **C++ 11** est désormais publiée. Cette évolution majeure du langage peut maintenant être implementée par les éditeurs, les développeurs. Microsoft a annoncé l'intégration de la norme dans les prochaines versions de Visual C++.

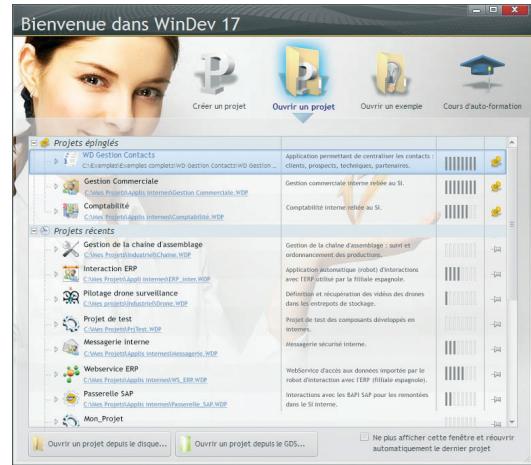


■ Le dernier rapport **Appcelerator** sur le développement mobile révèle que si iOS (91 %) demeure la premier système mobile pour développeur, Android est désormais très près (85%), même si la partie tablette demeure faible (à peine 70 % mais la moitié se dit réellement intéressée). Site : <http://www.appcelerator.com/company/survey-results/mobile-developer-report -april-2011/>

PC Soft dévoile WinDev 17

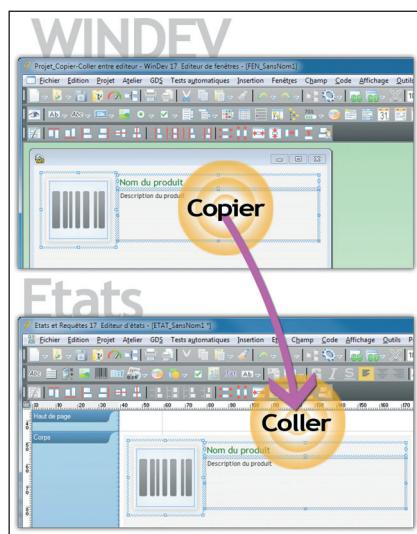
Déjà la 17e version de WinDev ou presque. La nouvelle version n'arrivera que dans plusieurs semaines. Une préversion est prévue courant décembre. Mais nous connaissons déjà les détails. La mobilité est l'un des mots-clés de la v17. Désormais, l'éditeur supporte nativement les terminaux iOS, une nouveauté très attendue, du fait du succès de l'iPhone et iPad en entreprise. Ce sont plus de 570 fonctions WLanguage qui sont supportées dans la génération iOS ! WinDev 17 supporte différents modes de distribution des applications : App Store, ad hoc ou InHouse. Aujourd'hui, WinDev supporte iOS, Android, Windows Phone 7.x, Windows Mobile. La partie Android gagnera en performances, par exemple, le tactile sur les images est maintenant automatiquement reconnu. Il supporte également le clic long. On retrouve aussi des fonctions pour l'appareil photo, l'accès à la synthèse vocale. Android 3 sera supporté. Parmi les grandes améliorations et nouveautés annoncées par l'éditeur, nous pouvons retenir :

- Nouveaux champs : champ organigramme, graphe entonnoir, notation, ruban... : ces nouveaux champs facilement personnalisables répondent à la demande en matière de création d'IHM moderne.
- Génération HTML 5 par WebDev.
- Possibilité de fusionner plusieurs documents PDF



- Accès 64-bit pour MySQL
- Support du protocole réseau IPv6
- Nouveaux ancrages de WEBDEV

La partie design interface s'enrichit régulièrement de nouveaux objets. Le ruban (à la mode MS Office) est maintenant personnalisable. En fonction du contexte de l'application, chaque volet peut être rendu visible, invisible ou grisé. Chaque champ est accessible en programmation (NomDuRuban.NomDuVolet.NomDuChamp), et toutes les propriétés sont accessibles. Parmi les améliorations d'interface, le champ planning évolue assez profondément : possibilité d'afficher le numéro de la semaine, option pour ne pas chevaucher plusieurs rendez-vous, modification des couleurs des plages horaires... Sur la partie graphe, on bénéficie d'un nouveau modèle : graphe entonnoir. On notera aussi une ergonomie affinée sur les différents éditeurs. Pour la partie internationalisation, Unicode est bien entendu présent pour les alphabets non latins. Cela devrait faciliter l'interfaçage avec d'autres logiciels et éviter des incompatibilités de caractères. Le cloud computing n'est pas oublié. Avec la v17, PC Soft propose un hébergement des applications et des données. Par exemple, depuis WebDev, vous pourrez déployer un site web sur le « cloud pc soft » (fonction disponible courant 2012 uniquement). Il faudra au préalable créer un compte et configurer celui-ci. Le coût dépendra des ressources utilisées. Côté langage WLanguage, nous pouvons noter une meilleure compilation dynamique, de nouvelles syntaxes dans les tableaux, la possibilité d'appeler des objets C++, d'utiliser Google Traduction. Outre les nombreuses améliorations, l'éditeur rappelle que WinDev 17 respectera les projets existants, une simple recompilation devrait suffire. La version standard sera vendue 1650 € (hors prix spécial jusqu'au 31/12/11). La mise à jour dépendra de votre version utilisée. ■



Android sur HP TouchPad

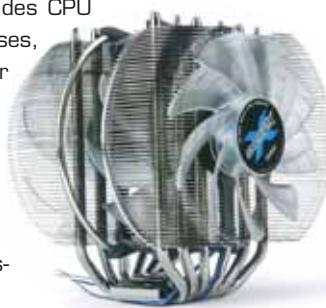
Pour les heureux propriétaires de la tablette TouchPad (brutalement arrêtée par HP durant l'été dernier), le projet CyanogenMod promet de fournir une version d'Android sur la tablette HP. Cependant, cette première version est à éviter selon les développeurs car trop immature, il s'agit d'une version alpha. Et beaucoup de travail reste à faire selon l'équipe : rajout de compatibilité envers les applications, pas de support des différentes fonctions matérielles, gestion de l'énergie peu efficace. Cependant, le projet supporte déjà les éléments de base : CPU, accélération matérielle, tactile. Mais avec l'avenir très incertain de webOS, la tablette HP n'a pas de réel avenir sauf à changer de système...



Pour en savoir plus : <http://rootzwiki.com/showthread.php?4011-Discussion-CyanogenMod-team-Touchpad-port>

Un problème de surchauffe ? Installer un ventirad

Le ventirad est un dissipateur de chaleur doté d'un ou plusieurs ventilateurs que l'on installe au-dessus d'un processeur (CPU, GPU, etc.). Ces ventirad sont parfois d'une taille impressionnante et dépassent le kilo ! Si vous utilisez des CPU overclockés ou des GPU monstrueuses, le ventirad sera indispensable pour refroidir l'ensemble, même si vous devez renforcer le contrôle de la chaleur interne au boîtier pour éviter une surface ailleurs... Les prix sont très variables dépassant 70 - 100 euros. Un des derniers nés est le CNPS12X de Zalman.



AMD développe Catalyst 11.10

AMD travaille activement sur une importante mise à jour de pilotes : AMD Catalyst 11.10, actuellement en préversion. Cette version inclura les pilotes pour les Radeon HD 6000 et 5000. Elle supporte Battlefiled 3, améliore les performances sur mono-GPU de type HD



4000, support de l'affichage Eyefinity 5x1. Cette version sera disponible pour Windows. La stabilité et les performances des pilotes sont des problèmes récurrents des GPU.

Les rendez-vous de l'open source

8
NOV

Réseaux Sociaux d'Entreprise : le meilleur de l'open source

Séminaire Smile
Paris / St Lazare - de 09:00 à 12:00

10
NOV

Drupagora

Événement européen sur Drupal
Paris - de 09:00 à 18:30

17
NOV

CMS open source : Le choc des titans

Séminaire Smile
Grenoble - de 09:00 à 12:00

22
NOV

GED open source

Séminaire Smile
Lille - de 09:00 à 12:00

24
NOV

Barcamp PrestaShop

Rencontre de la communauté PrestaShop
Paris / Espace Tapis Rouge - de 09:00 à 17:30

1er
DÉC

Jaspersoft, décisionnel : retours d'expérience et démonstrations

Séminaire Smile
Levallois-Perret - de 09:00 à 12:00



Pour vous inscrire
gratuitement,
www.smile.fr ou utilisez ce QR code

SQL Server « Denali » : le projet Juneau

Le but du projet Juneau est de fournir des outils intégrés à Visual Studio pour les développeurs d'applications ciblant SQL Server. Aussi connus sous le nom de SSDT – SQL Server Developer Tools, ces outils sont l'évolution des outils de gestion de base de données actuellement disponibles avec Visual Studio.

Le développement de ces nouveaux outils a plus particulièrement porté sur les points suivants :

- Apporter dans Visual Studio certaines capacités de **SQL Server Management Studio (SSMS)**.
- Améliorer la collaboration entre les développeurs d'applications, les développeurs de base de données et les administrateurs (DBA) grâce à une interaction poussée entre les projets de base de données et les EDM (Entity Data Model).
- Permettre de développer une application sur un modèle de données sans être connecté à une base (mode off-line).
- Permettre de débugger de bout en bout dans un seul outil les applications utilisant des bases de données.

Les outils de développement pour SQL Server (SSDT) se présentent comme un composant intégré à Visual Studio. En restant dans le même outil de travail, le développeur peut ainsi travailler sur le schéma de la base de données ou sur le développement d'applications. Ces outils se présentent aussi comme un package indépendant mais dans ce cas, les fonctionnalités spécifiques à Visual Studio (par exemple celles associées aux langages de programmation) ne seront pas disponibles. Juneau (ou SSDT) sera lancé avec **SQL Server - Denali** et sera aussi intégré à la prochaine version de Visual Studio. Afin de mieux comprendre les capacités de Juneau, nous allons créer une application de gestion de blog. Afin de se concentrer sur l'accès aux données et les fonctionnalités des outils, cette application sera en mode console. Nous allons procéder aux étapes suivantes pour créer notre application :

- Installation et découverte des outils SSDT.
- Création de l'application et test en mode déconnecté.
- Connexion à une base de données de test.
- Debugging.
- Publication dans SQL Azure.

Installation et découverte des outils

Pour commencer, nous allons installer les outils de développement pour SQL Server. Juneau est disponible en téléchargement libre à

cette adresse : <http://msdn.microsoft.com/en-us/data/hh297027> - [Fig.1].

L'ensemble des capacités que nous allons décrire dans le reste de cet article ne sont disponibles que si Visual Studio SP1 est déjà installé avant SSDT. La procédure utilise l'installateur **Web PI (Platform Installer)**, seuls les composants nécessaires à l'installation seront téléchargés. Pour Juneau, il est possible que l'installateur propose jusqu'à 20 composants additionnels.

Notes : Le setup de Juneau peut rendre *Crescent* instable. Si vous avez prévu de tester *Crescent* en version CTP3, je vous recommande d'installer Juneau sur une machine indépendante.

Une fois l'installation terminée, vous pouvez lancer Juneau depuis le menu démarrer – «**Microsoft SQL Server Developer Tools Codename Juneau**» ou lancer Visual Studio qui chargera automatiquement les composants SSDT [Fig.2].

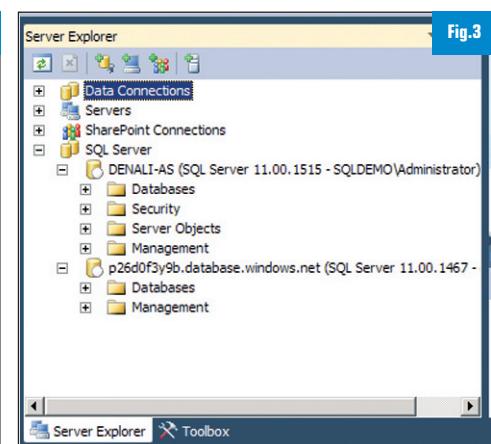
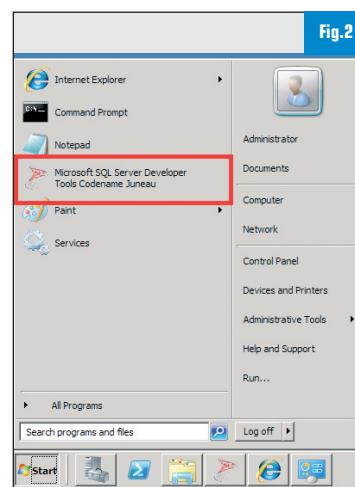
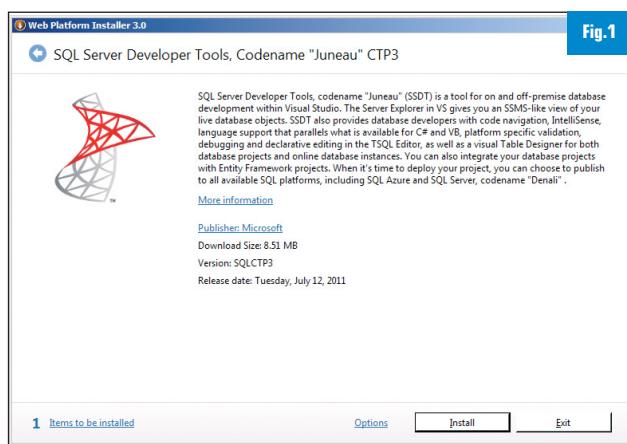
Le «**Server Explorer**» de Visual Studio contient maintenant un nœud «**SQL Server**». C'est sous ce nœud que l'on retrouve une hiérarchie et une organisation des fonctionnalités similaire à ce qui est disponible dans **SSMS (SQL Server Management Studio)**. Depuis cet emplacement, il est possible d'accéder et de piloter plusieurs instances SQL auxquelles le développeur a le droit de se connecter. Ces instances peuvent tourner sur SQL Server 2005, 2008, 2008 R2, Denali ou encore sur SQL Azure, comme dans l'exemple de la figure ci-dessous [Fig.3].

Nous allons maintenant créer notre application de gestion de blogs.

Création de l'application et test en mode déconnecté.

Nous allons commencer par créer une application en C# de type Console que nous appellerons «**BlogApp**». Dans l'application vide que Visual Studio instancie par défaut, nous allons ajouter un **Entity Data Model (EDM)**. SSDT apporte deux nouveaux types de contenu d'EDM : Database Model et Database Project:

- Le «**Database Model**» est un fichier de type .sqlx qui est un fichier contenant la représentation du modèle de base de données.



- Le «Database Project» crée un Entity Data Model dans l'application qui est synchronisé et maintenu à jour avec le projet de base de données.

Pour notre application, nous allons sélectionner la deuxième option «Database Project» et appeler notre projet de base de données «**BlogDbProject**». (Note: ces deux types de contenu sont disponibles dans une version ultérieure à la CTP3 des outils) [Fig.4].

Une référence vers le projet de base de données est ajoutée au niveau de l'application. Les changements effectués dans l'Entity Data Model seront synchronisés avec le projet de base de données et réciproquement. L'ajout de l'entité **Author** dans notre modèle génère automatiquement un script SQL représentant une table du même nom dans le projet de base de données. En ouvrant le fichier **Authors.sql** depuis le Solution Explorer on peut voir les propriétés de cette table plus en détail grâce au designer de table [Fig.5].

Depuis le designer de table, on remarque que toute modification effectuée dans l'une des vues de la table est automatiquement répliquée dans les autres. L'ajout de la colonne «**Name**» ayant des données de type «**nvarchar(50)**» dans notre entité est automatiquement répercuté dans le script T-SQL correspondant dans le projet de base de données. Ajoutons aussi à notre application l'entité **Post** avec les colonnes suivantes :

Name	Data type	Allow Nulls
Id	int	Unchecked
DateCreated	datetime	Checked
Title	nvarchar(50)	Checked
Tag	Nvarchar(50)	Checked

Maintenant, nous allons créer une association de type *many to many* entre l'entité des auteurs et celle des posts au niveau de l'Entity Data Model. Cela veut dire qu'un auteur peut avoir plusieurs posts et un post peut avoir plusieurs auteurs. SSDT va traduire et transposer cette contrainte de niveau conceptuel au niveau du schéma de la base de données. Dans le cas exposé, SSDT va automatiquement créer une **Joined table** dans le projet de base de données, les foreign keys et contraintes correspondant à cette association *many to many* [Fig.6].

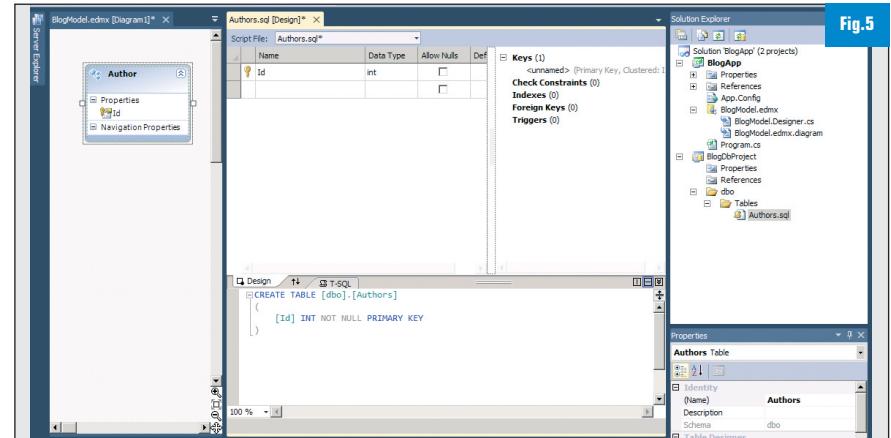
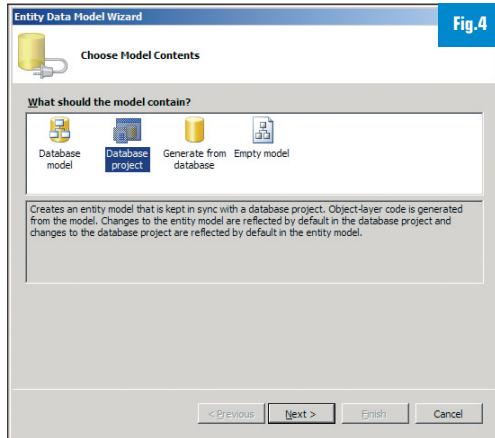
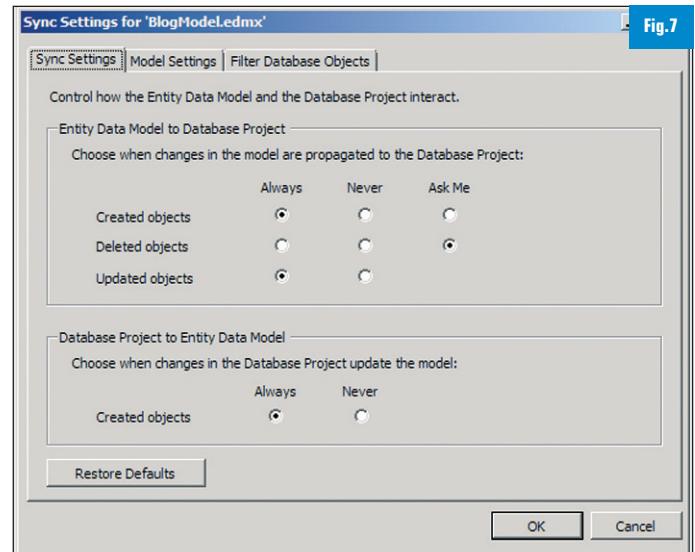
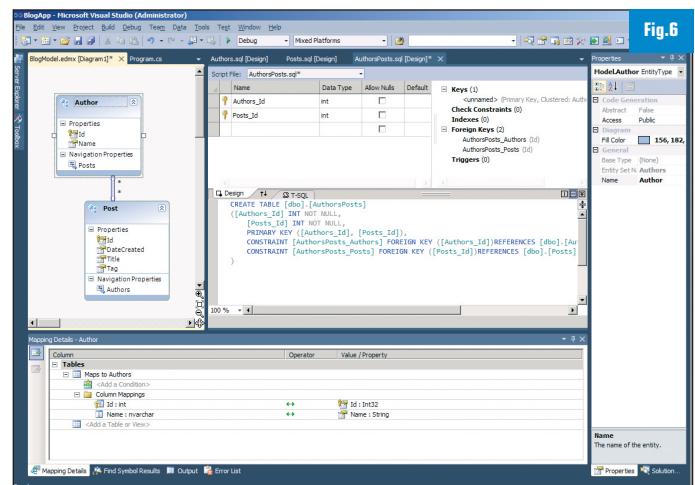
Nous allons maintenant ajouter une entité **GuestAuthor** qui hérite de l'entité **Author** créée précédemment. C'est après avoir créé l'entité qu'il est possible de la définir comme héritant d'une autre entité. SSDT traduit automatiquement cette opération en code au niveau du projet de base de données. Notez le mapping entre la colonne ID de la table GuestAuthor et la colonne ID de la table Author.

Il est possible de contrôler le mécanisme de synchronisation des

entités entre l'application et le schéma de base de données. Ce contrôle se fait au niveau de la création d'objets, de la suppression ou encore de la mise à jour quel que soit le sens dans lequel la mise à jour doit se propager [Fig.7].

La propagation des modifications effectuées sur une entité peut ne pas être souhaitée, comme par exemple dans le cas où le nom d'une propriété au niveau du modèle conceptuel doit être différent du nom de la colonne de la base de données.

Si, par exemple, on ajoute la propriété URL (nvarchar(50)) au niveau de la table **GuestAuthor**, cette propriété donnera lieu à la création



C'est l'histoire d'une entreprise qui voulait offrir des services virtualisés à ses propres clients. Mais après avoir mis en place sa nouvelle infrastructure informatique, elle s'est retrouvée dans une vraie galère : ses serveurs manquaient de puissance, et le réseau utilisé pour son stockage n'était pas aussi fiable qu'elle l'espérait. Alors qu'avec OVH...

- vSphere® as a service
- Des ressources livrées en 3 minutes
- Haute disponibilité
- Facturation horaire ou mensuelle
- Une infrastructure évolutive

100%
GARANTIE SLA*

Solutions Private Cloud

Créez votre datacentre
virtuel en un instant

Frais
d'installation
500€ HT
OFFERTS

Donnez vie à vos projets en ajoutant des machines virtuelles et des ressources. Puis adaptez cette infrastructure en temps réel.

- Une infrastructure évolutive selon vos besoins.
- Fonctionnalités VMware® incluses pour gérer vos ressources sans délai en toute sécurité.

À partir de

428,00 €
HT/mois

soit 511,89 € TTC/mois

Pour plus d'informations, visitez notre site

www.ovh.com/fr/pcc/ ou

contactez nos conseillers au **09 72 10 72 10**

(Coût d'un appel local)

*Garantie de disponibilité

N°1 de l'hébergement Internet en Europe

ADSL | SDSL | Domaines | Emails | Hébergement | Private Cloud | Serveurs dédiés | Cloud | Housing | Téléphonie | SMS & Fax


OVH.COM

d'une colonne dans la base de données. Il est possible de renommer cette propriété au niveau de l'EDM mais de ne pas propager cette modification vers la colonne de la base de données.

La construction de la base de données pour notre application est terminée. Ecrivons maintenant le code qui viendra consommer et afficher le contenu de la base. Pour cela, nous allons exécuter une requête dans la base de données et afficher tous les posts de la base, pour chaque post afficher les auteurs. Ci-dessous, l'exemple de code que nous allons utiliser pour afficher ce contenu.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace BlogApp
{
    class Program
    {
        static void Main(string[] args)
        {
            using (var oc = new BlogDbProjectEntities())
            {
                var posts = oc.Posts.Include("Authors").ToList();
                if (!posts.Any())
                {
                    Console.WriteLine("No posts added");
                }
                foreach (var p in posts)
                {
                    Console.WriteLine(Environment.NewLine + p.Title);
                    foreach (var a in p.Authors)
                    {
                        GuestAuthor ga = a as GuestAuthor;
                        if (ga == null)
                        {
                            Console.WriteLine(" " + a.UserName);
                        }
                        else
                        {
                            Console.WriteLine(" " + ga.UserName + " " + ga.BlogLink);
                        }
                    }
                }
            }
            Console.ReadLine();
        }
    }
}

```

En exécutant en mode debug l'application depuis Visual Studio, SSDT va démarrer localement une instance du moteur **LocalDB** et déployer le schéma de base défini dans le projet (BlogDbProject dans notre exemple). La chaîne de connexion est transmise à l'application grâce à la référence qui a été créée par SSDT entre l'application et la base de données (le fichier app.config est automatiquement renseigné).

Le runtime **LocalDB** peut être vu comme l'évolution de SQL Express, à la différence qu'il n'est pas nécessaire d'avoir installé d'instance sur son poste de travail et surtout qu'il ne nécessite pas d'avoir un service actif en tâche de fond. Un processus est instancié à la demande quand on appelle une instance de LocalDB et s'arrête automatiquement dès que l'instance n'est plus utilisée afin de libérer des ressources. Le processus tourne dans le contexte de sécurité de l'utilisateur l'ayant appelé.

Si vous surveillez les processus actifs de votre poste de travail après avoir démarré un nouveau projet de base de données, vous remarquerez un processus sqlservr.exe qui tourne dans le contexte de l'utilisateur qui a ouvert le projet. Les fichiers associés à cette base de données sont par défaut dans le répertoire C:\Users\<nom_utilisateur>\AppData\Local\Microsoft\Microsoft SQL Server Local DB\Instances\<nom_instance> [Fig.8].

L'instance par défaut est l'instance v11.0 mais dans notre cas, comme nous sommes dans le cadre d'un projet SQL, l'instance sera nommée avec le nom du projet. Cette instance peut être gérée par les outils traditionnels d'administration de SQL (en pensant à préfixer la connexion par [(localdb)]) ou encore à travers le nœud SQL Server apporté par SSDT dans le **Server Explorer** de Visual Studio. Ajoutons maintenant du contenu à notre base de données. En déployant l'arborescence sous SQL Server + [(localdb)]BlogApp jusqu'aux tables de la base. En allant sur **View Data**, Visual Studio affiche une data grid où il est possible d'insérer des lignes dans la base, par exemple avec un simple copier / coller d'un tableau créé dans Word ou Excel. Il est aussi possible d'utiliser un script en T-SQL pour effectuer les INSERT appropriés dans les tables Authors, Posts et AuthorsPosts (pour l'association entre les posts et les auteurs). Pour suivre le nombre total de posts que chaque auteur a publié il faut modifier le schéma de la base de données et ajouter une colonne dans la table *Author*, ajouter une procédure stockée pour incrémenter cette valeur à chaque fois qu'un post est ajouté et un trigger pour invoquer cette procédure. L'ajout de la colonne se fait de la même façon que ce que nous avons vu précédemment. L'intégration de SSDT avec Visual Studio permet de faciliter l'écriture de la procédure stockée grâce à IntelliSense. IntelliSense sait lire le projet de base de données et met à disposition du développeur, directement dans l'éditeur, le modèle de base de données utilisé [Fig.9].

Le trigger est créé au niveau de la joined database (AuthorsPosts.sql) qui a été générée automatiquement lors de la

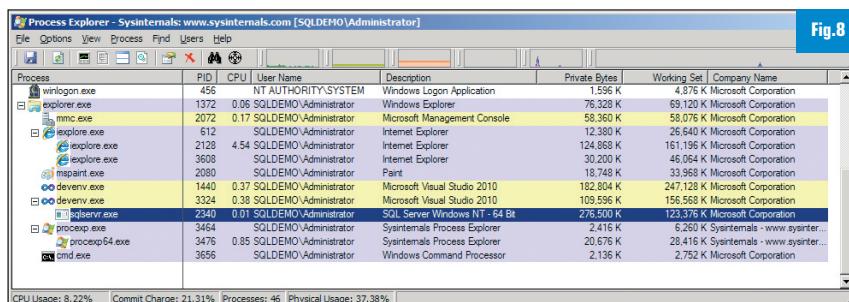


Fig.8

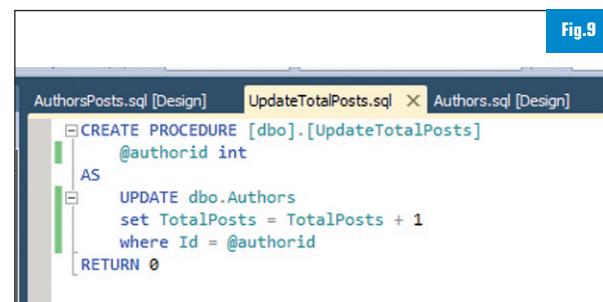


Fig.9

TOUT **SAVOIR** POUR TOUT **FAIRE**

Développez votre savoir-faire !

Programmez ! est le magazine de référence, depuis 1998, de tous les développeurs et chefs de projets logiciels.

Code, gestion de projets, développement web, mobile,

Programmez ! est à la fois :

- votre **outil pratique** : articles de **code**, par les meilleurs experts
- votre **veille technologique**



Et pour 10 euros de plus par an, offrez vous l'accès illimité à toutes les archives et numéros en format électronique.

Abonnez-vous à partir de 4€ seulement par mois

ABONNEZ-VOUS EN LIGNE

www.programmez.com

Toutes les offres sont en ligne



OUI, je m'abonne

Abonnement 1 an : 49€ 11 numéros par an au lieu de 65,45€, prix au numéro (*)

Abonnement 2 ans au magazine : 79€ (*)

Abonnement intégral : 1 an au magazine + archives : 59€ (*)

(*) Tarif France métropolitaine

à retourner, avec votre règlement à :
Groupe GLI, 17 chemin des Boulanger 78926 Yvelines cedex 9

M. Mme Mlle Société

Titre : Fonction : Adresse mail :

NOM Prénom

N° rue

Complément :

Code postal : Ville

Je joins mon règlement par chèque à l'ordre de PROGRAMMEZ Je souhaite régler à réception de facture (écrire en lettres capitales)

création de l'association au niveau de l'EDM [Fig.10].

On peut considérer que nous avons une première version de notre application. Regardons maintenant comment connecter cette application à un serveur de base de données et quels sont les apports de Juneau concernant le debugging de l'application.

Connexion à une base de données de test.

SSDT apporte la possibilité de créer un snapshot de la définition de la base de données. Ce snapshot est un fichier .SQLX qui est comme le schéma de la base par Juneau. Cette approche peut être utilisée pour le contrôle de version ou encore pour permettre aux développeurs d'application de travailler sur le code pendant que les DBA s'occupent de la base de données. Dans notre cas, nous allons créer un point de référence avant de connecter notre application sur un serveur de test.

Il est possible de publier la base de données sur le serveur de test depuis le projet de base de données par un clic droit – **Publish**. La publication va créer automatiquement la base de données, les tables, les procédures stockées et tous les éléments du schéma de la base sur le serveur de test. Comme l'opération peut prendre un peu de temps, il est possible de suivre dans la fenêtre **Data Tools Operations** la progression des différentes étapes de l'opération [Fig.11]. L'opération de publication ne déplace pas les données. Afin de vérifier le bon fonctionnement de l'application sur le serveur de test, nous devons injecter des données. Le plus simple est de reprendre le script en T-SQL que nous avons utilisé pour injecter les données dans la base localDB.

Nous allons modifier la propriété «**Storage Reference**» de l'Entity Data Model pour utiliser le fichier SQLX correspondant au snapshot créé précédemment. Le fichier SQLX est une représentation du modèle de la base que le développeur peut exploiter dans son application. Après avoir modifié l'application pour cibler le fichier SQLX, nous pouvons supprimer le projet de base de données de notre solution. Dans Solution Explorer, le fichier SQLX sera référencé sous le nœud «**Database Snapshots**» [Fig.12].

Le schéma de la base de données peut maintenant évoluer indépendamment du cycle de vie de l'application. Il est ensuite possible de prendre le nouveau fichier SQLX pour que l'application ait connaissance

sance des modifications qui auraient été apportées au niveau de la base sans pour autant détruire les personnalisations qui ont été réalisées par le développeur au niveau de l'Entity Data Model.

Il reste à modifier la chaîne de connexion dans le fichier app.config pour que notre application se connecte sur le serveur de test sur lequel nous avons publié le projet base de données.

Debugging

Explorons maintenant la fonctionnalité de Juneau permettant de faire du debug de façon transparente entre le code application et la base de données.

Commençons par ajouter le code suivant au début de notre application (entre le statement using et la déclaration de variable posts) pour déclencher une opération d'INSERT, de façon à vérifier si la procédure stockée fonctionne bien.

```
Post p1 = new Post
{
    Id = 5,
    Title = «A new Post»
};
p1.Authors.Add(oc.Authors.First());
oc.AddToPosts(p1);
oc.SaveChanges();
```

Comme c'est sur le `oc.SaveChanges()` que se fait l'opération d'INSERT, nous mettons un point d'arrêt sur cette ligne dans le code de l'application. De la même façon, nous allons poser un point d'arrêt dans la procédure stockée sur la base de test en utilisant le **Server Explorer** connecté à notre serveur. Les procédures stockées sont dans le nœud **Programmability** sous la base de données. Un clic droit sur **View Code** permet d'éditer le code de la procédure stockée et d'y poser un breakpoint [Fig.13].

Il faut aussi penser à autoriser le debugging d'applications au niveau du serveur avec un clic droit sur l'objet en question au niveau du Server Explorer ainsi que «**Enable SQL Server debugging**» dans les propriétés du projet C#. Il reste à lancer l'application en mode pas à pas. Le code s'arrête comme attendu sur le premier point d'arrêt dans le

Fig.10

Fig.11

Fig.12

code de l'application puis sur le second dans le code de la procédure stockée. Après avoir construit l'application en local puis testé sur un serveur de test toujours en local, il nous reste à mettre la base de données sur SQL Azure pour bénéficier des capacités du Cloud.

Publication de la base dans SQL Azure.

Lors de la création du projet de base de données, avec Juneau, il est possible de définir dans les propriétés du projet le serveur SQL et la version que l'on souhaite cibler [Fig.14].

En fonction de la version de SQL Server qui est définie comme cible, SSDT analyse en offline le schéma de la base et valide ainsi la viabilité du projet de base de données par rapport aux capacités de la plateforme cible. Par exemple, SQL Azure ne supporte pas les Filegroups alors que cette capacité est supportée avec SQL Denali. La publication du projet dans SQL Azure se fait exactement comme pour une publication vers SQL Server. Il faut penser à vérifier que le firewall a bien été configuré au niveau de SQL Azure pour que la publication se passe bien. Une fois que la publication s'est bien passée, une connexion à SQL Azure apparaît dans le Server Explorer à gauche. Il est possible de naviguer dans la hiérarchie et d'effectuer des requêtes de la même façon qu'on le ferait avec SQL Server (dans la limites des capacités de SQL Azure).

Il faut modifier la chaîne de connexion de l'application pour que celle-ci aille consommer les données depuis une base dans le Cloud.

Le projet **Juneau** ou encore **SSDT (SQL Server Developer Tools)** permet aux développeurs d'applications et DBA de mieux collaborer. Il est ainsi possible de créer une application consommant une base de données en mode déconnecté puis, à partir d'un point donné, de laisser l'application évoluer indépendamment de la base de données. Juneau permettra de resynchroniser les modifications effectuées. SSDT permet aussi d'apporter des fonctionnalités de SQL Server Management Studio dans Visual Studio et fluidifie au sein du même

environnement les méthodes de dépannage en permettant de passer en mode pas à pas du code de l'application au code des procédures stockées. Finalement, SSDT permet aussi de valider le schéma d'une base de données par rapport aux capacités de la cible avant de permettre une publication depuis Visual Studio.

Annexe – Script SQL

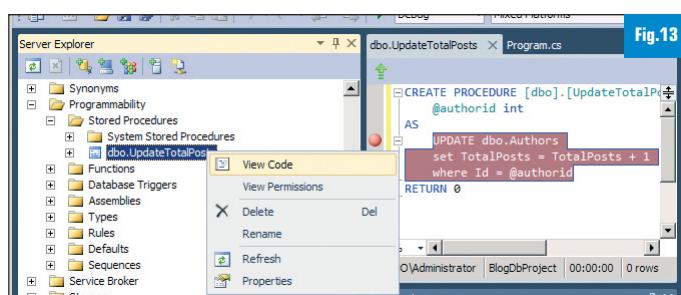
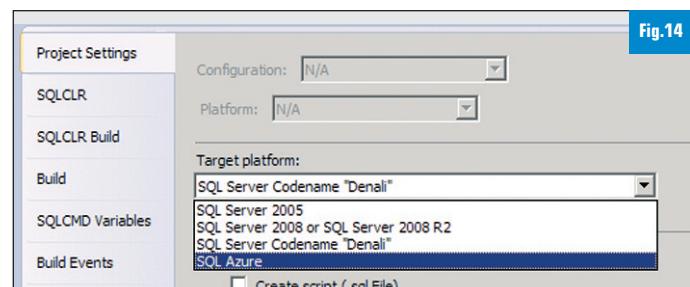
```
insert into dbo.Authors (Id, Name) values (1, N'Damien')
insert into dbo.Authors (Id, Name) values (2, N'Stanislas')
insert into dbo.Authors (Id, Name) values (3, N'Fabrice')
go

insert into dbo.Posts (Id, Title) values (1, N'Denali CTP 3 est au top !')
insert into dbo.Posts (Id, Title) values (2, N'Venez découvrir la prochaine version de Windows')
insert into dbo.Posts (Id, Title) values (3, N'Projet Juneau, la meilleure façon de développer')
insert into dbo.Posts (Id, Title) values (4, N'Denali: Always On')
go

insert into dbo.AuthorsPosts (Authors_Id, Posts_Id) values (1,1)
insert into dbo.AuthorsPosts (Authors_Id, Posts_Id) values (2,2)
insert into dbo.AuthorsPosts (Authors_Id, Posts_Id) values (3,3)
insert into dbo.AuthorsPosts (Authors_Id, Posts_Id) values (1,3)
insert into dbo.AuthorsPosts (Authors_Id, Posts_Id) values (2,3)
insert into dbo.AuthorsPosts (Authors_Id, Posts_Id) values (2,4)
go

insert into dbo.GuestAuthors (Id, URL) values (3, N'Microsoft')
go
```

■ Damien Caro



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

L'AR. Drone, un OVNI pour geek

1^{re} partie

Apparu pour la première fois lors du grand salon des nouvelles technologies, le CES à Las Vegas en 2010, l'AR. Drone est un petit hélicoptère télécommandé, plus précisément un quadricoptère, permettant de donner vie à nos rêves de pilote de haute voltige.

Nous allons nous intéresser à cet OVNI d'un nouveau genre.

Parrot, entreprise très connue pour ses accessoires Bluetooth nous livre cette fois un drôle de gadget qui va en ravir plus d'un parmi nous.

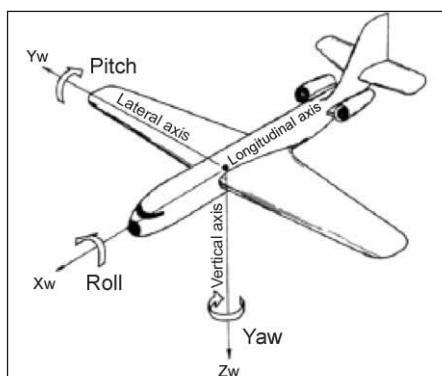
Pourquoi faire le choix d'un appareil de ce type-là ? J'imagine que nous ne le saurons jamais exactement, mais j'ose penser que les ingénieurs de l'entreprise ont su garder une âme d'enfant, qu'on peut retrouver dans tout geek et... chez beaucoup d'entre nous. Ce n'était pas forcément le marché de l'entreprise par le passé, oui, mais, le résultat est là pour prouver l'effet bénéfique pour l'entreprise en termes de résultat financier et surtout d'image de marque.

Ok, mais des appareils volants télécommandés, cela n'a rien de nouveau en soi ? Oui, mais contrôlables par WiFi à partir d'un simple Smartphone ? Cela prend tout de suite de l'intérêt pour nous, joyeux geek que nous sommes !

La nouveauté de l'AR. Drone est donc cette capacité d'offrir un engin volant pour un prix correct permettant à chacun de se prendre pour un grand pilote à partir des appareils que nous manipulons maintenant tous les jours. L'idée est vraiment de s'appuyer sur les technologies que l'on côtoie dans nos appareils électroniques, mais dans un



D.R.



contexte un peu plus fun. Il vous suffit donc d'un Smartphone, officiellement d'un iPhone / iPad ou d'un terminal Android à l'heure actuelle, pour prendre le contrôle par WiFi d'un AR. Drone. Une connexion est établie

entre les deux appareils et ensuite l'application sur l'appareil mobile vous permet de contrôler l'appareil volant. Il est donc temps de rappeler les bases de l'aviation et connaître le bon vocabulaire pour briller en société.

Petit cours d'aéronautique

Tout appareil volant, du moins sur la planète terre, repose sur les mêmes grands principes de physique, l'AR. Drone n'y déroge pas et même si nous ne sommes pas là pour piloter des avions de ligne, il est bon d'avoir les bases pour

être en mesure de faire un vol correct et surtout sans crash quelques secondes à peine après le décollage.

Outre le fait de savoir maîtriser sa vitesse et son altitude, chose que l'on peut encore accepter comme assez classique, il faut surtout noter l'existence des trois grands axes de contrôles d'un appareil volant. Ces axes sont essentiels pour bien comprendre et surtout maîtriser le vol d'un appareil tel que l'AR. Drone.

- Le pitch vous permet d'incliner d'avant en arrière votre appareil
- Le Roll vous permet de pivoter de gauche à droite votre appareil
- Le Yaw vous permet d'effectuer une rotation L'AR. Drone étant composé principalement

de quatre moteurs entraînant chacun une hélice, ces trois grands axes sont donc manipulables en effectuant de légères variations de puissance de tel ou tel moteur. On diminue la puissance des deux moteurs avant ? L'AR. Drone se penche en avant et par la force des choses... avance tout droit. Chaque action possible par le biais des différentes applications existantes pour contrôler l'AR. Drone est traduite en changement de puissance moteur permettant d'effectuer une rotation selon un de ces trois axes. Ces changements d'axes permettent à l'appareil de voler dans toutes les directions et de la manière la plus fluide possible, mais surtout, pour les bons pilotes, dans des espaces fort réduits permettant ainsi les vols en intérieur.

Il n'y a pas à en savoir beaucoup plus pour maîtriser l'AR. Drone, tout en n'oubliant pas qu'il ne pèse que 400 grammes, donc si vous le faites voler en extérieur, vos mouvements seront forcément plus difficiles à contrôler, pour ne pas dire impossibles. Le vent sera votre pire ennemi durant vos vols extérieurs et dès qu'il dépasse quelques km/h il est fortement déconseillé de voler, car c'est prendre le risque de retrouver votre AR. Drone dans un arbre ou pire ! Donc prenez votre mal en patience et attendez une période plus calme, voire envisagez un vol en intérieur si vous avez suffisamment de place.

Il faut noter que l'appareil possède également un accéléromètre et un gyroscope qui lui serviront pour se stabiliser du mieux possible quand vous ne lui donnez pas de commande à suivre, mais également pour tenter de rendre plus fluide l'ensemble de ses mouvements dans l'espace. Cela vous donnera un sacré coup de main pour garder un vol correct et sans dégât. Ces capteurs lui permettent d'effectuer de jolis vols et vu sa vitesse de pointe proche des 20km/h, on obtient cette vision de l'ovni filant devant les yeux qui ne laisse personne de marbre. À vous de faire le nécessaire pour ne pas finir ce superbe vol dans le mur du voisin, je ne vous cache pas que cela impressionne beaucoup moins !

Et sinon, il a quoi dans le ventre ?

L'AR. Drone est vraiment un petit bijou de technologie, nous avons déjà cité de nombreuses caractéristiques telles que ses quatre moteurs, l'accéléromètre et le gyroscope, mais n'oublions pas que tout cela est



contrôlé par un Linux... eh oui l'AR. Drone possède une version embarquée de Linux qui joue le rôle de cerveau permettant de contrôler et donner vie à l'appareil. On commence déjà à voir la différence avec un simple hélicoptère radiotélécommandé, car à l'inverse de ce dernier qui ne répond qu'à des ordres simples, mais surtout de manière directe sur ses moteurs, notre AR. Drone est capable de s'adapter et d'utiliser un panel large d'appareils pour améliorer ses vols. On trouve donc un CPU de type ARM qui tourne tout de même à 468MHz, ce qui est loin d'être négligeable pour un appareil de ce type, et nous verrons par la suite qu'il faut de la puissance pour effectuer toutes les actions qu'il doit supporter lors d'un vol.

Le système d'exploitation est donc là pour orchestrer l'ensemble des actions de l'appareil que ce soit de manière autonome, comme le mode urgence que nous détaillerons un peu plus tard ou bien pour effectuer les actions reçues par le biais de la

connexion WiFi, son moyen de communication avec l'extérieur, c'est-à-dire vous. Mais ce cœur possède également un rôle fondamental, celui d'utiliser les deux caméras qu'embarque l'appareil. La première est la caméra frontale, celle que l'on remarque le plus en regardant la forme de l'AR. Drone, elle permet de transmettre « ce que voit » l'appareil en avançant

dans l'espace avec un angle de vision de 90 degrés et avec une résolution de 640 sur 480 pixels. Mais une fonction forte intéressante également est sa capacité à reconnaître des Tags, vous savez, ces images faites de noir et blanc pouvant contenir une information. Ces Tags peuvent ensuite être utilisés dans des scénarios de jeux par exemple, voire de réalité augmentée, car l'image est traitée par le Linux embarqué permettant ainsi l'analyse du flux vidéo en direct. La seconde quant à elle est bien plus cachée, on peut la trouver sous l'appareil, elle peut également être récupérée pour observer sous l'appareil à partir de votre iPhone ou tout autre contrôleur, mais son rôle est bien plus important pour le cerveau qui tourne à l'intérieur de notre OVNI. Cette caméra permet à l'appareil de calculer sa vitesse et de l'aider dans ses phases de stabilisation. Il est également aidé dans ce but par l'altimètre à ultra-sons qu'il embarque également sous sa coque. C'est vrai qu'au premier abord on remarque surtout son

fuselage en fibre de carbone, mais au final ce n'est vraiment pas la partie la plus intéressante de ce jouet. Il recèle tellement d'appareils électroniques que cela en fait bien plus un drone intelligent qu'un simple « jouet ».

Peut-on lui trouver un point négatif ? Sans doute la durée de vie de sa batterie, car il faut bien alimenter tous ces appareils et forcément cela influe grandement sur la durée des vols. Avec une charge complète, comptez environ 2h de charge, il ne faut pas espérer voler plus de 15 minutes. Pour une utilisation classique de l'AR. Drone, on ne peut donc que conseiller l'achat au minimum d'une seconde batterie pour alterner les périodes de charge et d'utilisation entre les batteries. L'AR. Drone est vraiment un assemblage savant d'éléments utiles qui en font un appareil passionnant. N'oublions pas que tout cela ne pèse que 400 grammes et ce n'est pas la coque qui va ajouter beaucoup de points à l'ensemble, coque que vous pourrez retirer lors de vos vols en extérieur pour gagner encore quelques précieux grammes, mais si vous le faites, attention aux dégâts en cas de contact direct avec les hélices ! Notez tout de même la présence d'un système de sécurité qui coupera tous les moteurs au moindre contact, mais il n'en reste pas moins désagréable de se le prendre sur les doigts ! Le choix d'utiliser la coque est donc à faire avant chaque vol, dans le cas d'un vol en intérieur la question ne se pose même pas, équipez votre appareil de sa coque à moins de faire de la casse ou de se blesser. Dans le cas de vols extérieurs, vous serez les meilleurs juges pour savoir si elle à son utilité ou non. Prenez juste en compte que sans la coque votre vol sera sans doute un peu plus contrôlable, car il offrira moins de prise au vent, mais par contre les chutes pourront être plus brutales et pourront endommager votre AR. Drone.

Mise en route

Maintenant que vous avez reçu votre formation de base et que vous savez ce que contient la bête, il est temps de faire un premier vol. Pour cela, les différentes applications existantes fonctionnent toutes sur le même principe, une connexion en mode ad



hoc entre l'AR. Drone et votre Smartphone et un échange constant d'informations permettant de contrôler le vol, mais également de recevoir des informations de l'appareil comme ses flux vidéos, mais aussi son niveau de batterie par exemple. Une fois la liaison établie entre les deux appareils, les diodes se trouvant sur chacune des hélices vont prendre une couleur verte indiquant que le vol est prêt à débuter. Il existe deux façons de faire le pilotage, à vue ou avec l'aide des caméras embarquées. Chaque application offre les commandes permettant d'effectuer les actions vues durant notre cours d'aéronautique et bien sûr les actions telles que décoller, atterrir et une petite dernière que l'on nomme « emergency » ou « urgence » en français dans le texte qui, comme son nom l'indique, est à utiliser si on perd le contrôle de l'AR. Drone. Le logiciel embarqué s'occupera alors de prendre le contrôle et de se poser le plus proprement possible, entendez par-là que l'appareil va dans un premier temps se stabiliser puis s'approcher du sol jusqu'à se poser, puis couper ses moteurs. En général cela se passe tout en douceur et vous évite des atterrissages forcés générateurs de dégâts. L'ensemble des applications présentes sur le marché offre les mêmes fonctionnalités de

navigation et rend l'appareil contrôlable comme si nous étions dans un simple jeu vidéo. La gestion des moteurs ne doit pas nous préoccuper, on demande au drone d'avancer, il s'occupe lui-même de donner la bonne puissance aux moteurs avant ou arrière pour effectuer l'action demandée. L'ensemble des actions sont faites de manière propre et le cerveau de l'AR. Drone s'occupe de rendre cela le plus fluide possible malgré le fait que de temps en temps vous risquez d'être un peu trop stressé sur les commandes. Faites simplement confiance à vos qualités de pilote et profitez du vol qui a lieu devant vos yeux.

Mais au final, cela sert à quoi ?

Comme bien des activités qui nous passionnent, nous pourrions être tentés de répondre « pas à grand-chose », mais cela ne serait pas forcément vrai. Reprenons les bases si vous voulez bien :

- Un appareil volant
- Rapide et agile dans un environnement en 3 dimensions
- Deux caméras
- Contrôlable à distance
- Un noyau Linux
- Des capteurs en tous genres

Passé l'annonce de lancement, l'AR. Drone fut utilisé en premier dans des jeux intégrant de la réalité augmentée. Si vous faites voler plusieurs appareils de ce type, chacun ayant une caméra et une coque d'une couleur précise, comment ne pas imaginer ajouter une interface sur le flux vidéo et transformer un simple vol en reconstitution d'un champ de bataille du futur ? De nombreuses applications ont été publiées depuis sa sortie se basant sur ce principe de la réalité augmentée, le fait qu'il soit piloté par un Linux lui permet de lui-même de faire de la reconnaissance de Tag dans son flux vidéo. Rien que cela ouvre la porte à bien des expériences.

Avec toutes ces qualités, ne pourrions-nous pas lui trouver une utilité autre que celle de simple jouet ? Certains chercheurs ont récemment étendu un peu les possibilités de l'appareil pour lui permettre de cracker les réseaux WiFi à portée de détection. Oui, tout simplement une machine volante

capable de se promener autour de chez vous et de vous rapporter une collection de points d'accès avec leurs mots de passe pour y accéder. Un autre exemple serait bien sûr d'en faire un AR. Drone de surveillance, soyons honnête il est encore un peu bruyant pour espérer espionner votre voisine, mais je suis certain que vous pourrez trouver des cas d'utilisation où le bruit des moteurs ne sera pas si gênant que cela. Il faut vraiment voir l'AR. Drone comme un appareil à mi-chemin entre le jouet par son prix et le drone de surveillance pour le matériel qu'il contient. À voir par la suite si vous voulez continuer de jouer avec ou bien passer à la deuxième phase qui serait de l'utiliser pourquoi pas d'une manière un peu différente du simple vol fait pour impressionner les copains.

API ? SDK ?

Vous ne pensiez tout de même pas qu'un appareil de ce genre serait fourni sans un SDK ? Comment nous autres amateurs de nouvelles technologies et bidouilleurs dans l'âme ferions-nous pour nous occuper sans ces API ? Eh bien oui, il existe bien un SDK pour l'AR. Drone également. Il fonctionne sur une base de C, ce qui permet de l'utiliser au final sur à peu près toutes les plateformes et quel que soit le système d'exploitation de celle-ci. Je ne vous cache pas que l'API en elle-même est un peu brute, mais on sent par son biais la connaissance que possède Parrot des appareils liés au protocole de communication moderne tels que le WiFi ou Bluetooth. Le SDK est disponible sur le site de Parrot (<https://projects.ardrone.org>) où une simple inscription permet d'y avoir accès et de nombreuses documentations et exemples vous permettront de vous y plonger de manière concrète. L'API expose tout le nécessaire pour communiquer avec votre AR. Drone. Une fois qu'une connexion ad hoc aura été faite entre les deux appareils, avec un enchaînement précis de commandes vous pourrez donner l'instruction à votre AR. Drone de prendre son envol et ensuite de naviguer dans l'espace qui lui est offert, mais attention à la casse ! Il serait judicieux d'y aller par étape, en commençant par exemple par celle qui me semble primordiale... Permettre l'appel du mode emergency rapidement. N'oubliez jamais que votre AR. Drone est quelque chose de fragile, donc même si développer avec est un réel plaisir, il est bon de vous assurer qu'il n'ira pas s'abîmer

contre le premier mur rencontré lors de son décollage par le biais de votre implémentation d'un contrôleur.

Nous verrons dans un prochain article plus en détail comment tirer parti de l'API fournie par Parrot pour rendre votre AR. Drone pilotable à partir d'un PC, voire d'appareils moins classiques. Une fois que vous aurez mis les doigts dedans, vous ne pourrez plus vous empêcher de réfléchir à la nouvelle façon de contrôler ce drôle de drone. L'idée de Parrot, le constructeur de l'AR. Drone, est simple, les développeurs que nous sommes sont les personnes les plus aptes à ajouter de nouvelles fonctionnalités à l'appareil. Il faut être honnête, ce type de jouet cible un public assez précis, les amateurs de nouvelles technologies, donc, nous. À partir de là, dans l'entourage d'un geek il y a forcément un ou deux développeurs qui pourront donner vie à la nouvelle idée, donc le meilleur moyen reste de distribuer un SDK comme Parrot le fait, et de voir fleurir à travers internet les nouveaux projets reposant sur ce kit de développement.

Ou l'acheter ?

Lors de son lancement en France durant le mois d'août 2010, l'AR. Drone était vendu seulement dans les magasins de la marque Fnac, suite à un accord de distribution signé entre ces deux entreprises. Mais la situation a maintenant évolué et vous pourrez vous le procurer dans de très nombreuses enseignes que ce soit dans de petits magasins de modélisme ou de grandes chaînes de distribution possédant des magasins ou des sites de vente en ligne.

Malheureusement, son prix n'a que peu évolué. En un an, il est resté en effet peu ou prou le même. Vous pourrez donc trouver l'AR. Drone dans de nombreux types de magasins, mais excepté lors de potentielles offres promotionnelles, son prix restera toujours autour des 300 euros pour l'appareil en lui-même (à cela se rajoutent les nombreux accessoires. Ce prix peut sembler élevé pour un jouet, mais n'oubliez pas que ce n'est pas juste un jouet, mais un vrai appareil de modélisme et de surveillance, donc si on relativise un peu, son prix est tout à fait correct pour ce qu'il contient et offre en termes de matériels embarqués.

Nous l'avons aussi évoqué, l'AR. Drone est un appareil fragile, je pense avoir assez insisté là-dessus. Mais même si je suis persuadé que vous volerez avec toutes les pré-

cautions possibles, il faut garder en tête qu'une mauvaise chute peut arriver et que l'appareil risque de subir de petites casses. Parrot l'a bien compris et même si sa fragilité est due à son besoin de conserver un poids plume, on peut presque se poser des questions quand on voit la quantité de pièces détachées qu'il est possible d'ajouter de façon séparée de l'appareil. Comprenez par-là que vous avez tout le nécessaire pour changer l'ensemble des pièces de votre AR. Drone en rayon et cela pour des prix tout aussi élevés. Certes on peut admettre que remplacer la carte mère puisse coûter cher, mais la question reste en suspens pour certaines pièces plus classiques, mais si spécifiques que le choix n'existe pas quand il s'agit de donner un second souffle à votre appareil.

Quelques tarifs :

AR. Drone neuf : 299,90 €
Batterie : 29,99 €
Boîte à outil officiel Parrot : 13 €
Carte mère : 99,99 €
Engrenages / axes : 9,99 €
Hélices : 6,99 €
Moteur : - 40 €

Quelques adresses :

Site officiel : <http://ardrone.parrot.com/parrot-ar-drone/fr/>
Site d'actualité et boutique en ligne : <http://droneparrot.com/>

Conclusion

Nous avons survolé les possibilités et les éléments constituant cet OVNI sorti il y a maintenant déjà quelques mois et que l'on rencontre au détour des rassemblements d'amateurs de nouvelles technologies, et j'espère que vous en avez maintenant une vision plus claire. Je crois que l'essentiel à retenir de ce jouet pour adulte, car soyons honnête je doute que votre neveu de 5 ans en fasse bon usage, est qu'il faut plutôt le voir comme un croisement entre les plaisirs du modélisme et de la bidouille informatique. Parrot a su proposer un appareil permettant de remplir ces deux rôles pour nous autres passionnés. En espérant que cet article vous aura donné envie d'y goûter également et pourquoi pas, d'aller plus loin ensemble avec la découverte de l'API plus en détail une prochaine fois.

■ Neils Freier

Consultant .Net, expert geek - Wygwam



Sécurité

Mais que fait le développeur ?!

Ne jetons pas la pierre au développeur, car il n'est pas l'unique fautif de l'insécurité actuelle des ordinateurs, des données, des réseaux. Les facteurs sont multiples : mot de passe trop faible, serveurs et logiciels non patchés, réaction des éditeurs trop lente ou incomplète, bonnes pratiques de sécurité dans le code non appliquées ou trop partiellement, etc. Il faut mettre en exergue deux éléments : une programmation pas toujours respectueuses des règles de sécurité et de réduction des surfaces d'attaques, une veille technologique incomplète (quand elle existe).

La veille technologique sur la sécurité constitue un enjeu crucial pour le développeur, les responsables réseaux, etc. Il est important de se tenir au courant des évolutions, des dernières attaques, des



failles, des correctifs sorties, etc. Mais ce travail n'est pas fait, surtout en entreprise, car cela nécessite du temps et un suivi constant. Même la programmation selon une méthode de développement sécurisé demeure imparfaitement appliquée car le projet va durer plus longtemps et oblige à renforcer les cycles de tests, notamment avec l'usage intensif des outils d'analyse de code pour repérer les

vulnérabilités du code, le non-respect des normes de codage, etc. Et ce cycle est à renouveler à chaque nouvelle version.

Dans ce dossier, nous allons aborder plusieurs éléments de sécurité : botnet, le forensique, PHP ou encore, pénétrer au cœur du laboratoire de sécurité de l'Epitech.

Comment devenir un développeur spécialiste en sécurité ?

Directeur technique chez Prim'X technologies, éditeur français de logiciels de sécurité, Pierre-Jean Leca a toujours été fan d'informatique et de développement. Il nous raconte son parcours et la façon dont il recrute de nouveaux talents qui travailleront sur des logiciels dédiés à la sécurisation des Systèmes d'Information ...

Après avoir développé des jeux vidéo, Pierre-Jean Leca se lance dans des études d'ingénieur en informatique à l'INSA Rennes. A la fin de celles-ci, il entame sa carrière professionnelle dans une SSII puis intègre une start-up. Là, il collabore avec France Télécom sur les sujets Internet et Communication instantanée (prémisses de Messenger). En régie pendant trois ans chez différents clients, sa fonction est essentiellement celle d'un développeur, sans la composante sécurité à cette époque-là. Très vite, sur le terrain, il apprend des techniques et langages, des compétences que l'on acquiert difficilement durant un cursus scolaire du fait de la rapidité d'obsolescence des techniques en ces domaines. Mais l'école « apprend à apprendre » et finalement c'est ce qui compte sur le terrain : monter rapidement en compétence grâce à un apprentissage sur le tas, les environnements informatiques des clients étant tous différents la plupart du temps.

De la théorie à la réalité ...

Ainsi l'école sert plutôt à former au niveau des méthodes de développement : phase de spécification, phase d'architecture, le développement propre, test, validation ... Dans le cycle classique (cycle en V), l'apprentissage débute par la spécification et le cahier des charges, l'architecture qui permet de passer de l'aspect fonctionnel à organisationnel au niveau des différents composants du futur produit (logiciel client qui a des interactions avec des composantes serveur sur Internet ou sur tout autre réseau, par exemple), et enfin la manière dont chaque composant va être développé en module (se mettre d'accord entre les différentes interfaces entre les modules, séparer le travail entre différents développeurs). Cette dernière phase d'architecture correspond, en quelque sorte, au plan de ce qui va être développé. Puis suit la phase de développe-



ment propre dont le but est de coder les modules de façon unitaire (chaque programmeur devant assumer les tests entrée/sortie sur sa partie en charge). Là, la phase d'intégration débute (tests entre modules) qui s'achève sur celle de validation (vérification que le produit développé est bien en phase avec les spécifications et le cahier des charges). Dans la réalité, ce modèle scolaire se transforme en une approche beaucoup plus itérative. Impossible, en effet, de tout spécifier depuis le début de A à Z, car le besoin réel est difficile à cadrer dès le démarrage d'un projet, à établir d'un seul coup. Ce n'est qu'au fur et à mesure des développements que les commanditaires arriveront à mieux le définir... A chaque itération, le client réadapte les spécifications. Ce processus sur le terrain permet, au bout du compte, d'impliquer beaucoup plus le client pendant le développement et par conséquent, de prendre moins de risque au moment de la délivrance du projet quant à sa satisfaction. « *Ainsi l'école nous aura fait comprendre comment cela se passe sous le capot et ce, quel que soit le langage utilisé ou l'environnement ambiant, car au niveau matériel comme sur le plan de l'architecture technique (assembleur, mémoire ...), rien ne bouge. Fait qui permettra au développeur au bout du compte de mieux comprendre les couches supé-*

rieures et donc d'optimiser au mieux le code », commente Pierre-Jean Leca.

Une expérience terrain avant tout

Sur le terrain, l'apprentissage du langage (C, Java ...) n'est pas le plus important, seul compte véritablement l'algorithme. En d'autres termes, savoir résoudre, par exemple, l'indexation d'un document de façon rapide et donc traduire le problème en différentes étapes, elles-mêmes programmées par la suite dans un langage donné. La technique (autocommutateur téléphonique, fonctionnement d'une messagerie instantanée ...) est, elle, vue sur le tas. Pour Pierre-Jean Leca, la régie pendant 3 ans, ce fut l'assurance de connaître plusieurs types de terrains, d'emmagasiner de l'expérience, d'apprendre à comparer les méthodes et au bout du compte de développer un esprit beaucoup plus critique. Puis retour au bercail, avec le modèle « forfait » qui correspond au développement du projet client en interne dans la SSII. Là, en plus de l'expérience de développement, il prend la casquette « chef de projet » qui, non seulement, lui donne une vision globale du projet mais lui apprend également à diriger une équipe comme à communiquer avec la maîtrise d'ouvrage. « *Là, on apprend à laisser ses développeurs face à leurs diffi-*



cultés, à déléguer et à leur faire confiance. Cela peut être coûteux d'attendre trois jours que le problème se décale ... mais c'est le chemin de croix obligatoire pour un développeur pour apprendre la persévérance. Une persévérance qui lui permettra de contourner un mur, un problème ardu. Un apprentissage qui trouve toute son utilité lorsque le développeur prend de l'expérience : plus il monte en niveau plus il est difficile de l'épauler. Pour cela, il doit pouvoir compter sur lui-même et donc apprendre à progresser seul face à un problème donné. Devenir chef de projet, c'est aussi savoir distribuer la pression de manière fine aux personnes qui composent son équipe... tout en faisant tampon entre les impératifs extérieurs et les développeurs. »

Selon ce dernier, la difficulté est de sélectionner non pas la ligne dans le CV mais le potentiel ... Par exemple, une SSII recrute des « mots-clé » pour s'assurer de placer son personnel. Fait qui ne lui est pourtant pas arrivé lorsqu'il fut sélectionné par Serge Binet, DG et fondateur de Prim'X. « La sécurité est assez « sexy », ludique quand on vient de l'extérieur. C'est un domaine qui attire le développeur, notamment dans le domaine de la cryptographie qui éveille souvent la curiosité », se rappelle-t-il.

Acquérir la spécificité Sécurité ...

Pierre-Jean Leca relève le défi et décide de se lancer dans la carrière de développeur pour un acteur dans la sécurité. « Apprendre ne me fait pas peur et j'étais curieux. Je me suis dit que j'allais apprendre plein de choses sur un domaine qui m'intéresse, qui m'amuse ... Un produit cryptographique est composé à la base d'ingrédients dont, entre autres, des algorithmes de chiffrement (dont AES, chiffrement des données sur 256 bits en général). Prim'X ne fait pas dans l'algorithme fait maison mais reste dans le standard. De là découlent naturellement les notions de clés publiques et privées en cryptographie asymétrique, de certificat, de PKI (infrastructure de gestion de clés et fonctionnement). Une fois ces différents ingrédients passés en revue, vient l'apprentissage de la recette ou autrement dit du schéma de sécurité : comment faire en sorte que ces différents algorithmes mathématiques permettent de construire un schéma de sécurité (sachant que l'algorithme AES représente moins de 5% des sources du produit à développer) ? A savoir

comment simplement articuler les notions fonctionnelles pour arriver à sécuriser à l'aide de ces ingrédients ? Un schéma qui permettra de définir le format de la structure des fichiers du produit à développer (déchiffrage d'un fichier, comment retrouver la clé AES ...). Le schéma, c'est également ce qui sera soumis à l'approbation de l'ANSSI, organisme gouvernemental officiel qui donne des certifications aux solutions de sécurité du marché qui répondent à certains critères draconiens. Cette certification garantit, en d'autres termes, que sans le mot de passe ou une carte à puce, il est impossible de déchiffrer ou de lire les données de l'utilisateur. Le mot de passe ou la carte étant les sésames pour entériner les opérations mathématiques qui se succèdent permettant, au bout du compte, de chiffrer et déchiffrer les données. Le cœur du produit étant constitué des algorithmes et des schémas de sécurité. » L'écriture d'un produit de sécurité comprend également une part d'applicatif qui enrobe le cœur. L'idée est de ne pas perturber l'utilisateur final dans sa façon de travailler au quotidien. Ainsi l'idée est d'intercaler une couche

de chiffrement dans tout ce qu'il va manipuler : un document Word ? Le contenu est obligatoirement chiffré, surfer sur internet ou l'intranet d'entreprise ? Les données doivent être chiffrées ... Par conséquent, le travail à fournir ici est plutôt au niveau système (Windows, Linux ...), y insérer de façon transparente une couche de sécurité invisible pour l'utilisateur. Outre cette partie très technique, il y a également une partie applicative qui concerne, par exemple, l'administrateur qui doit gérer des accès partagés sécurisés et ce, de façon centrale. A partir de là, il faut déterminer les profils de développeurs qui pourraient convenir à chacune de ces parties ... Un débutant se formera d'abord sur le produit (cryptographie, schémas de sécurité ...) puis sur la partie qui lui sera dévolue (sa fonction). Au bout de 6 mois à un an, il commencera à avoir une connaissance du schéma de sécurité plus fine : il pourra commencer à corriger des anomalies dans le schéma. Et au bout de trois à quatre ans, il pourra proposer ses propres schémas de sécurité.

■ Solange Belkhayat-Fuchs

Les différentes étapes d'apprentissage

1 Apprendre les méthodes du monde de la cryptographie, notamment, en écoutant les fondateurs de l'entreprise qui ont une solide expérience en la matière. Pour les débutants il existe de la bonne documentation. Bruce Schneier, par exemple, propose une encyclopédie et également un autre livre avec une vision plus pragmatique, plus appliquée des différents algorithmes. Dans la réalité, il y a la gestion de changement de clés publiques, les logiques de péremption, de révocation ... Beaucoup de processus lourds qui génèrent du coup des failles dans le système mathématique initial, sans défaut.

2 La sécurité en soi : elle s'apprend sur le terrain avec des développeurs plus expérimentés. Dans la technique pure de développement, il faut savoir rester vigilant. Par exemple : libérer des mots de passe implique la libération de la mémoire allouée après avoir « vidé » le mot de passe lui-même. Là, il faut savoir qu'il y a des habitudes de programmation précises pour éviter d'insérer, à son insu, des failles dans le code quand on procède à ce genre de manipulation. Et ces techniques de programmation diffèrent si l'on travaille sur un produit dans un navigateur web ou encore sur un driver. Elles dépendent tout simplement de l'environnement en question et ne s'apprennent en aucun cas à l'école ... Généralement les développeurs seniors passent le relais. Sinon, en interne chez Prim'X, un « guide de style » indique comment structurer les sources. Un guide qui comprend également des parties dédiées aux techniques connues pour éviter des failles ou savoir mieux gérer les secrets. C'est-à-dire, par exemple, lors de la gestion des logs dans le produit développé, éviter d'y cataloguer des mots de passe ...

3 Comprendre les schémas de sécurité utilisés par l'application pour être capable, à terme, d'écrire son propre schéma de sécurité. Savoir imaginer de nouvelles architectures pour écrire de nouveaux produits : essentiellement être capable de s'intercaler de façon transparente au niveau système ; également savoir quelle architecture mettre en œuvre pour allier sécurité et déploiement facile pour l'entreprise. Un développeur expérimenté sera force de proposition sur le schéma de sécurité.

VOTRE CODE EST MULTI-PLATEFORMES
Windows, Linux, .Net, Java, PHP, Mac,
J2EE, XML, Internet, SaaS, Cloud,
Windows Phone, CE, Mobile, Android, iOS, ...

DÉVELOPPEZ
10 FOIS PLUS VITE

917
NOUVEAUTÉS

VERSION
EXPRESS
GRATUITE
Téléchargez-la !

Opérations en cours :
• 1 PC pour 1 Euro de plus
• Tour de France de la
version 17 (11 villes).
Tous les détails sur pcsoft.fr

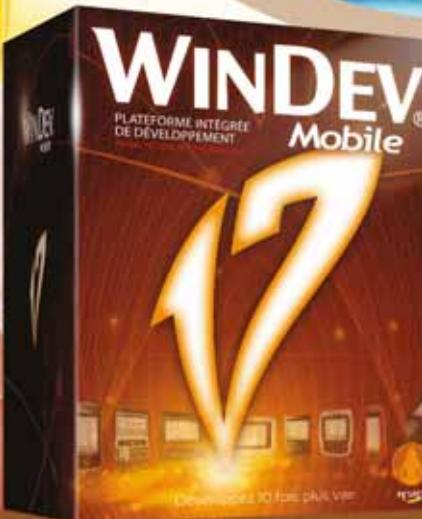
ATELIER DE GÉNIE LOGICIEL PROFESSIONNEL

WINDEV® MOBILE

NOUVELLE VERSION



ET MAINTENANT SUR
iPHONE & iPad



iOS (iPHONE, iPad)
ANDROID
WINDOWS PHONE
WINDOWS MOBILE



Et bien entendu votre code est compatible entre les matériels, mais également avec Windows, Linux, Mac, Internet : c'est ça la magie WINDEV !



► 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



La France dans la cyber guerre

Début 2011, un vaste attaque avait eu lieu contre Bercy, et plusieurs autres institutions (Elysée, quai d'Orsay). Plus de 150 postes ont été touchés. Une des portes d'accès, notamment à Bercy, fut le mail, via un « simple » message piégé. En 2009, c'était une attaque par DoS. L'éditeur McAfee en août 2011 parlait d'un cyber-espionnage mondial contre 70 organisations mondiales (<http://blogs.mcafee.com/mcafee-labs/revealed-operation-shady-rat>) et de nombreuses sociétés de défense, dont l'origine et la motivation restent à définir, même si la Chine fut citée comme possible instigatrice de cette vaste attaque.

Nous avions l'impression que durant des années, l'Etat français était laxiste, voire, en retard sur le sujet. Oui et non, serions-nous tentés de dire. Et les pouvoirs de l'ANSSI (agence nationale de la sécurité des systèmes d'information) se renforcent périodiquement. C'est le bras informatique de la Défense nationale qui s'occupe de tout ce qui est guerre électronique, attaques contre les intérêts nationaux, etc. Bref une véritable cyber-armée. Durant le printemps arabe, les pouvoirs bloquaient l'accès internet, sans être aussi extrême, l'ANSSI peut exiger des opérateurs un blocage par exemple des ordinateurs en cause dans une attaque. En cas d'urgence, la France dispose du plan Piranet. Ce plan remplit 4 objectifs de base : sécuriser les moyens de transmissions, sécuriser les systèmes d'information des administrations, mise en place de capacité opérationnelles pour répondre aux attaques, inscrire la politique de sécurité informatique et des systèmes d'information dans le cadre de la politique française de sécurité dans l'Europe.



Un plan d'action contre les attaques

Pour accomplir, cela, Piranet repose sur plusieurs mesures : formations, organisation, équipement, cadre légal, etc. Ainsi dans la définition de Piranet, il est clairement dit : « *Le manque de spécialistes compétents en sécurité des systèmes d'information au sein des administrations est devenu particulièrement alarmant. L'acquisition de compétences nouvelles et la mise à niveau permanente de celles qui sont disponibles sont donc une impérieuse nécessité pour l'ensemble des administra-*

tions. Des besoins considérables de formation et de sensibilisation en résultent. » (mesure F1). Mais il s'agit aussi pour l'ANSSI de sensibiliser l'ensemble des hauts responsables et les administrations des menaces. Une simulation de ce plan a été réalisée en 2010.

Pour compléter les dispositions et surtout la latitude d'accès de l'ANSSI, le décret du 11 février 2011 se révèle important. L'article a été modifié ainsi : L'article 3 du décret du 7 juillet 2009 susvisé est modifié comme suit : Après le deuxième alinéa est ajouté un autre alinéa ainsi rédigé : « *- elle assure la fonction d'autorité nationale de défense des systèmes d'information. En cette qualité et dans le cadre des orientations fixées par le Premier ministre, elle décide les mesures que l'Etat met en œuvre pour répondre aux crises affectant ou menaçant la sécurité des systèmes d'information des autorités publiques et des opérateurs d'importance vitale et elle coordonne l'action gouvernementale* ». cela signifie que l'ANSSI met en place et organise toutes les mesures nécessaires contre une attaque comme par exemple le blocage ou filtrage d'Internet.

Une cyber-armée bien réelle ?

D'autre part, en mai 2011, le Premier Ministre a annoncé la création d'un groupe d'intervention rapide, toujours dans cette logique de cyber-armée.

Sa mission primaire est :

- **rechercher et détecter** les compromissions ;
- **superviser** les opérations de traitement



d'incident ou de reconstruction des systèmes ;

- **porter** assistance à nos alliés en cas de crise informatique.

Ce groupe sera doté d'une logistique pour se projeter à l'étranger, notamment pour aider nos alliés dans une crise informatique majeure.

Parmi les mesures annoncées, on apprend que la carte à puce sera généralisée dans l'administration (projet piloté par l'ANSSI). Un intranet étatique sera créé. Cette cybersensibilisation passe aussi par la formation, l'enseignement. Ainsi, il est dit : « *Une fondation sera créée, sous la forme d'un partenariat public-privé, pour soutenir les capacités de recherche nationales et développer un centre spécifiquement dédié à la recherche en sécurité des systèmes d'information, dans les domaines techniques et non techniques (politique, géostratégique, économique, etc.)...*

Trop d'ingénieurs ou d'universitaires, y compris dans les filières techniques, arrivent aujourd'hui sur le marché du travail sans avoir jamais été formés à "l'hygiène numérique" ». Enfin, un système d'alerte doit pouvoir alerter et sensibiliser les administrations, et le public.



Plus largement sur la guerre électronique, nous vous conseillons deux ressources :

<http://www.cheat.defense.gouv.fr/fr/pdef/histoire/guerrelecry.pdf>

<http://www.scribd.com/doc/32580009/PIA-03-363-Schema-directeur-de-la-guerre-electronique-France-2009>

et aussi :

<http://www.securite-informatique.gouv.fr/>



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.*



Actus / **Événements** / **Newsletter** / **Vidéos**



www.solutions-logiciels.com

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

Envoyer par la poste à : Solutions Logiciels, service Diffusion, GLIE - 17 chemin des Boulanger 78926 Yvelines cedex 9 - 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



Du forensique à l'Epitech : l'autre face de la lutte contre le cyber-crime

Marwan Burelle, enseignant et directeur du laboratoire « système et sécurité » à l'Epita revient sur l'étude et l'application du forensique à l'Epitech et notamment dans le cadre de l'accord avec l'OCLCTIC (office central de lutte contre la criminalité liée aux technologies de l'information et de la communication).



Programmez ! : l'approche forensique en informatique est peu connue en France. pouvez-vous expliquer en quoi consiste cette démarche ? Et son importance.

Marwan Burelle : Le terme initial désigne l'analyse scientifique des preuves sur une scène de crime. Appliquée au monde numérique, cette démarche s'intéresse à l'étude des traces ou des preuves de nature informatique aussi bien dans le cadre de la cyber-criminalité (piratages, arnaques informatiques ...) que dans un cadre criminel plus classique. L'analyse forensique s'étend aussi bien de l'étude des fichiers présents sur un ordinateur, jusqu'à l'analyse de la mémoire d'un téléphone portable pour retrouver des messages texto. L'extension de la criminalité dans la sphère numérique rend la mise en place de ce type d'approches incontournable aussi bien pour les forces de police que pour des services privés. Il ne faut pas non plus négliger l'importance croissante des nouvelles technologies dans la vie quotidienne, même lorsque la criminalité reste dans un cadre «classique», les objets numériques de tous les jours (téléphones portables, «smartphones», ordinateurs personnels ...) peuvent être une source importante d'information pour les enquêteurs.

P ! : Quelles sont principales techniques et les outils que l'on peut utiliser en forensique ? Quels seraient les scenarii types d'utilisation du forensique ?

MB : Il est difficile de faire une réponse exhaustive sur les techniques ou les outils car ceux-ci sont très nombreux. Schématiquement, il existe deux formes d'analyse en forensique: l'analyse dite « post-mortem » s'intéresse à l'extraction d'informations de périphériques (principalement des mémoires de masse comme les disques durs ou les mémoires « NAND » de téléphones.) Cette

approche correspond tout à fait au travail de forensique pour la criminalité classique (comme les autopsies.) et l'analyse « à chaud » quant à elle va s'intéresser à la recherche d'informations et de preuves directement sur place. Notamment, on essaie d'extraire la mémoire d'un ordinateur pour récupérer les informations « volatiles ». L'analyse post-mortem est un domaine bien établi, pour lequel il existe de nombreux outils, on peut citer par exemple: EnCase ou X-Way (outils commerciaux.). En marge des outils logiciels, on est parfois amené à utiliser des outils « plus matériel » comme ceux nécessaires à l'extraction des mémoires de téléphones ou d'équipement embarqué, ou les « bloqueurs » qui permettent de copier un disque dur en garantissant que celui-ci ne pourra pas être modifié pendant l'opération.

Pour l'analyse à chaud, la situation est beaucoup plus complexe. Il est possible d'extraire des informations d'un ordinateur en cours d'exécution à l'aide d'outil d'administration classique (par exemple la suite COFEE de Microsoft qui propose de récupérer à l'aide d'une clef USB des résultats d'analyses faits avec les outils systèmes standard de Windows).

Malheureusement, l'analyse à chaud est principalement nécessaire dans des situations de compromissions (intrusion ou malware) où les éléments recherchés sont probablement capables de se cacher des outils classiques. Dans ce genre de situation, on est souvent amené à utiliser des techniques similaires à celles des pirates pour extraire une image complète de la mémoire en vue d'une analyse ultérieure. Il existe un outil, Volatility, qui permet l'analyse d'une image mémoire (récupération de la liste des processus, de leurs pages associées ...). De manière générale, les outils de forensique sont souvent des « boîtes à outils » regroupant tout un ensemble de fonctionnalités, principalement des outils de recherche et de récupération de fichiers.

P ! : Début 2011, l'Epita a signé un partenariat avec l'OCLCTIC. En quoi consiste cet accord et quels sont vos projets ?

MB : L'OCLCTIC a plusieurs missions qui vont de la formation à l'investigation de terrain, en passant par le support pour les équipes de terrain. Les objectifs du partenariat entre EPITA et l'OCLCTIC sont doubles: un échange de compétence dans le domaine de la formation en rapport avec la sécurité informatique et une activité d'étude et de développement d'outils. Dans le cadre du LSE (Laboratoire Système et Sécurité de l'EPITA) nous travaillons, en particulier, à l'élaboration d'outils destinés à l'extraction d'information « à chaud » (extraction puis analyse de mémoire). Certains élèves de l'école (dans la majeure SRS, Système Réseau et Sécurité) maintiennent une activité de veille technologique et de collecte d'informations.

P ! : A l'Epitech, il y a le projet Digital Forensics Framework, quel est l'objectif de ce travail et comment s'inscrit-il dans la sensibilité des étudiants au forensique en général ?

MB : DFF est un ancien projet de fin d'études Epitech devenu un vrai outil OpenSource. Les anciens étudiants qui maintiennent ce projet ont même monté une structure autour de celui-ci, la société ArxSys. DFF est typiquement une « boîte à outils », il regroupe de nombreux éléments (montage de disque ou d'image disque, recherche de fichiers par format ou contenu ...) ainsi que la possibilité d'ajouter des plugins afin d'intégrer d'autres outils OpenSource (comme Volatility). L'un des points fort de DFF est son côté « extensible » : l'ensemble du projet a été pensé pour offrir un cadre simple au développeur qui souhaite intégrer un nouveau plugin afin d'exploiter l'outil dans son ensemble mais aussi les autres plugins disponibles. Par exemple, dans le cadre du LSE, nous utilisons DFF comme interface à Volatility pour regrouper et piloter des analyses d'image de mémoire.



■ F.T.



Au cœur de l'Epitech Security Laboratory



L'ESL, pour Epitech Security Laboratory, est le laboratoire de sécurité de l'école. Il réunit près d'une vingtaine de membres qui s'intéressent à la sécurité informatique au sens large. Son but est d'aborder avec les étudiants les problèmes liés à la sécurité. Le cursus se déroule sur 4 ans. Julien Castets de l'ESL a répondu à nos questions.

Programmez ! : L'insécurité informatique est un sujet d'actualité, comment voyez-vous son évolution ? Est-ce pire, mieux ou pareil qu'il y a 2 ou 3 ans ?

Julien Castets : Avec l'avènement du cloud computing, les données sont de plus en plus externalisées. Pourquoi s'occuper soi-même de son serveur, des sauvegardes, de sa haute disponibilité alors qu'Amazon — par exemple — nous propose de gérer tout cela à moindre coût ? Tout va bien... jusqu'au jour où il y a un problème. Donner ses données à une entreprise, c'est comme donner ses clés de voiture à un homme devant un restaurant : tant qu'il s'agit d'un voiturier, tout se passe bien. Jusqu'au jour où...

P ! : Absence de patch, pas de mise à jour des outils et serveurs, mot de passe trop simple, imprudence et inconscience des utilisateurs et des développeurs / administrateurs, aujourd'hui, quelles sont les causes des failles, trous et autres passoires informatique ?

JC : Il faut distinguer l'attaque globale de l'attaque ciblée. Dans le premier cas, si l'attaquant n'a pas de cible particulière, l'absence de mises à jour s'avérera fatale : pourquoi perdre du temps à attaquer un serveur à jour alors que les dix suivants ne le seront pas ? Pour l'attaque ciblée s'ajoute le problème de l'utilisateur. J'ai fait une expérience il y a quelques mois. J'ai laissé le message suivant sur facebook : "quand on écrit notre mot de passe sur facebook, il est automatiquement remplacé par des étoiles ! ***** !" - un ami dans le coup répond : "ça marche ? *****" - un autre, aussi dans le coup : "oh ça marche !! *****" D'après-vous, que répondent les suivants ? Vous seriez étonnés du nombre de mots de passe que vous pouvez récupérer. Ce n'est pas de la sécurité à proprement parler, mais c'est très révé-

lateur du comportement des utilisateurs : pour avoir un mot de passe, le plus simple est souvent... de le demander. Du côté du développement des logiciels, on assiste à de véritables échecs. C'est le cas du PSN, le Playstation Sony Network sur lequel les joueurs de la célèbre console doivent se connecter pour pouvoir jouer en réseau. Il y a de cela quelques mois, Sony s'est fait voler plusieurs millions de comptes, incluant informations personnelles et numéros de cartes bleues. Pour cela, les attaquants ont utilisé une simple injection SQL, exploitable par n'importe qui. Aujourd'hui donc, les failles qui font le plus de dégâts me semblent être les failles les plus simples à exploiter.

P ! : Il est souvent dit que l'utilisateur est le premier en cause dans les problèmes de sécurité, mais le développeur, et les éditeurs, ont un rôle à jouer. Il existe des méthodes strictes pour mieux coder ou tout du moins appliquer une sécurité dès le codage des sites, des applications. Comment voyez-vous l'attitude du développeur envers la sécurité ?

JC : On ne peut pas garantir la sécurité d'une application : celui qui vous affirme sans aucun doute que votre application n'a aucune faille, est soit un dieu, soit un escroc. On peut cependant essayer de l'améliorer et pour cela je pense qu'il n'y a qu'un mot : formation. Si un développeur web ne sait pas ce qu'est une injection SQL et ne sait pas comment l'exploiter, je vois difficilement comment il pourrait s'en protéger. Il ne faut pour autant pas oublier que peu importe les qualités du développeur, il n'est jamais à l'abri d'une erreur. C'est pour cela qu'il est primordial de faire tester et revoir le code par des personnes externes au projet. Plus nombreuses sont les personnes qui se penchent sur la sécurité d'un produit et plus il sera, en toute logique,

sécurisé. Pour finir, il existe bien évidemment des moyens d'éviter les failles les plus courantes. Je pense ici à l'utilisation de frameworks opensource. Pourquoi utiliser un framework rend-t-il souvent le code plus sûr ? Pour les deux raisons évoquées ci-dessus : parce qu'ils sont créés par des développeurs qui ont conscience des problèmes liés à la sécurité ; et que leur code est régulièrement revu par des personnes externes.

P ! : Quels sont pour vous et le ESL, les réflexes à avoir quand on gère un serveur, un site web ?

JC : Si on ne peut s'assurer de la fiabilité du code d'un site web, on peut au moins essayer de limiter les dégâts en cas d'exploitation. Prenons par exemple codepad.org, site permettant de compiler et d'exécuter du code sur un serveur distant. On est donc plus ou moins dans le cas d'un attaquant qui arriverait à injecter du code depuis notre site web. Comment le serveur est-il sécurisé ? Voilà un résumé de ce qu'on nous apprend sur <http://codepad.org/about> : les droits sont limités à leur strict minimum : si l'utilisateur n'a pas besoin d'un service, il ne faut pas hésiter à lui interdire. Ainsi, le jour où l'utilisateur malicieux prendra le contrôle de votre application, il ne pourra pas non plus s'en servir ; les exécutables sont isolés dans des jails. Si vous devez héberger plusieurs sites web sur le même serveur, il ne faut pas pouvoir remonter de l'un en passant par l'autre : la sécurité dépend toujours du maillon le plus faible ; le site web est développé en utilisant des frameworks opensource dont la sécurité a été testée à maintes reprises par un grand nombre de personnes (ici, Pylons et SQLAlchemy). Ainsi, on profite de l'expérience des développeurs de ces frameworks ; et enfin des superviseurs ont été mis en place afin de détecter de possibles attaques. codepad



est, je pense, un très bon exemple des règles à appliquer quand on veut administrer un site web puisqu'on y trouve plusieurs couches de sécurité : si une couche est défaillante, celles se situant au-dessus devraient limiter la progression de l'attaquant.

P ! : Comment au ESL sensibilisez-vous vos étudiants ? Et plus généralement à l'Epitech ?

JC : Comme expliqué dans la description du laboratoire, nous organisons de nombreux projets tout le long du cursus. Notre but est de former les étudiants aux problèmes de sécurité souvent rencontrés autant d'un point de vue protection (patcher le code qui est faillible) qu'exploitation (réussir à exploiter la faille). L'informatique est un domaine si vaste que tous les étudiants ne s'intéressent bien évidemment pas à la sécurité. Notre but est de donner des bases à tous afin qu'ils comprennent les enjeux qui sont liés à la mauvaise conception d'un produit. Quant aux passionnés, nous leur proposons de nous rejoindre et ainsi d'accéder aux ressources du laboratoire !

P ! : En France, nous avons l'impression que la sécurité informatique, les grandes

attaques, les failles sont quasi des gros mots. Forensique, connaît pas, keylogger, idem (et pourtant autorisé par l'Etat), hyperkit, etc. est-ce la réalité ou la déformation des médias ?

JC : Il est très difficile de parler de la sécurité informatique car lorsqu'on veut vulgariser des problèmes qui sont parfois très techniques, on tombe rapidement sur des reportages dans lesquels le public est véritablement pris pour un âne. En montrant un Anonymous qui craindrait pour sa vie s'il retirait son masque ou encore en faisant un reportage sur trois adolescents qui essaient sans succès de pirater le réseau wifi d'un hôpital, les média tentent de faire du sensationnel. Après de tels reportages, les professionnels de la sécurité rient ou pleurent, selon les personnes. Et le grand public n'en sait pas plus : on veut lui montrer un feu d'artifice, pas comment les choses fonctionnent vraiment.

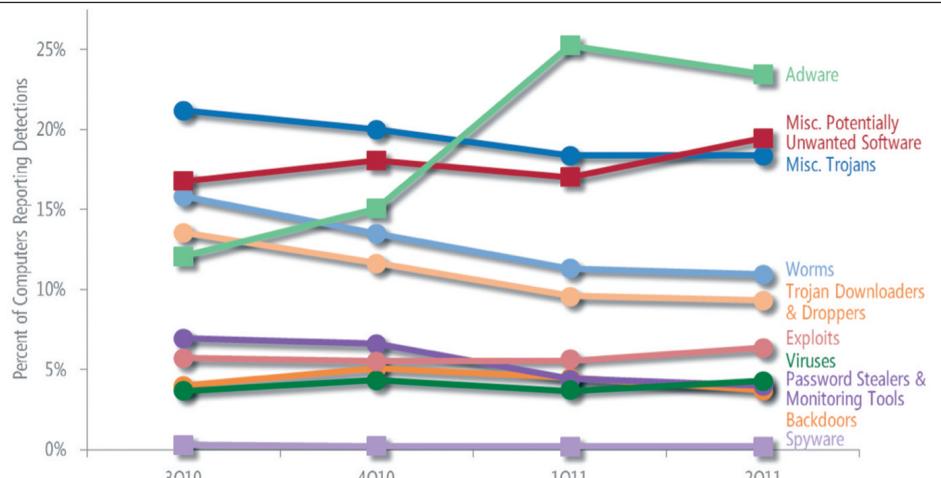
Donc oui, il y a une véritable désinformation des média. Je pense que pour parler de sécurité, il faut pouvoir être didactique, vulgariser, mais certainement pas prendre le public pour un idiot. Le public peut comprendre ce qu'est un keylogger. Ce n'est pas quelque chose de mystique ou d'insurmontable : il faut juste lui expliquer.

Le cursus de l'ESL

- En 1re année, le **Pepito**. Nous fournissons aux étudiants un serveur TCP écrit en C qui intègre des failles plus ou moins dissimulées. Leur but est de trouver et corriger ces failles
- En 2e année, le **ThotProject** se déroule sur deux semaines. Nous mettons à disposition des étudiants une cinquantaine de sites web écrits en PHP. L'objectif est de réussir à obtenir des accès administrateur, injecter du code ou encore récupérer des informations dans la base de données. C'est donc une véritable introduction au pentesting : si les premiers niveaux sont relativement simples (avec des failles facilement exploitable), les derniers sont d'un tout autre genre
- En 4e année c'est le projet **Jamby** qui réunit les étudiants. Le but est de créer un programme permettant d'infecter un exécutable sur Windows afin qu'il se répande. Ne serait-ce pas la définition du virus ?
- En 4e année encore se déroule l'un des projets phares de l'école : le **Security-Quest**. Il s'agit là de récupérer des certificats sur des ordinateurs situés dans l'école mais aussi dans les entreprises partenaires qui souhaitent tester la fiabilité de leur architecture informatique.

Détections par catégorie de menaces, entre le troisième trimestre de 2010 et le deuxième de 2011, par pourcentage de l'ensemble des ordinateurs signalant des détections

© Microsoft, rapport de sécurité, 11e édition



ABONNEMENT PDF 30 € par an
soit 2,73 € le numéro www.programmez.com

Les botnets

Le terme *botnet* est construit d'après la contraction des deux mots «roBOT» et «NETwork», ce qui signifie littéralement réseau formé par des robots. Le robot étant ici un logiciel localisé sur un ordinateur.



Les premiers robots logiciels ont vu le jour dans l'univers des gros systèmes : ils étaient le plus souvent utilisés pour réaliser des opérations de maintenance (sauvegarde). Par extension, ces robots ont eu à gérer à distance des tâches de plus en plus complexes, telles que le contrôle centralisé de capteurs, ou d'autres ordinateurs (pour détecter des unités centrales en panne dans un centre informatique, par exemple).

Les botnets malicieux ont d'abord été inspirés par ceux qu'utilisaient les opérateurs IRC. Ils servaient à assurer la gestion d'un canal de dialogue (signalisation, bienvenue aux arrivants) pour décharger l'administrateur de ses tâches répétitives. La première application malveillante fut d'utiliser ces outils d'administration afin de prendre le contrôle d'un canal IRC. C'est ainsi qu'Eggdrop, ancêtre des bots malveillants, est apparu en 1993.

Lorsqu'un ordinateur personnel est affecté par un cheval de Troie tel que ceux que nous venons de décrire, il devient une machine zombie. C'est-à-dire une machine contrôlée à l'insu de son utilisateur par un tiers. Généralement, ce dernier l'utilise alors le plus souvent à des fins malveillantes, par exemple pour voler les données de votre disque dur et usurper votre identité. Le PC infecté par un troyen n'est pas un botnet, tout au plus possède-t-il les fonctionnalités d'un bot unique. Imaginons maintenant que ce tiers malveillant ne détienne plus seulement l'accès à un troyen sur une machine mais à des milliers, identiques, sur des milliers de machines ! Que ces multiples troyens soient par ailleurs reliés entre eux et puissent recevoir des ordres pour mener des actions coordonnées. Ce ne sont plus alors des troyens, mais de véritables centres de calculs informatiques, distribués (donc difficiles à atteindre), granulaires, donc peu vulnérables. Un cauchemar !

En tout cas, voilà ce qu'est un botnet : un bot associé à un troyen, et actif sur un réseau de machines.

En tant qu'utilisateur individuel, vous êtes peu concerné par les nuisances du botnet lorsqu'il est utilisé en tant qu'ensemble dont la finalité est la perturbation ou l'extorsion de fonds. Car c'est l'une des activités de ces réseaux fantômes, qui sont utilisés principalement par les organisations qui les opèrent à des fins d'extorsion de fonds (après avoir bloqué ou réduit le trafic de vos serveurs ou de votre site informatique).

Zdnet (<http://www.zdnet.com/blog/security/study-finds-the-average-price-for-renting-a-botnet/6528>) a publié un exemple de message envoyé par des attaquants munis de botnet à leurs cibles et dont voici ci-dessous la transcription en français : « Bonjour, si vous souhaitez que votre site demeure opérationnel, vous devez nous payer 10 000 roubles mensuellement. Attention, après la date du (DATE) votre site sera victime d'attaques DDoS. Votre site deviendra inutilisable tant que vous ne nous paierez pas. La première attaque impliquera

2000 bots. Si vous contactez une entreprise impliquée dans la protection contre les attaques DDoS nous augmenterons le nombre de bots de 50 000 et la protection contre 50 000 bots est très onéreuse ».

Suit une offre... de fourniture de bot pour attaquer les sites des concurrents ! Cet usage du botnet nous montre que les botnets posent deux problèmes : le premier est qu'ils sont utilisés par des organisations criminelles à des fins de perturbation et de racket, le second est que votre ordinateur, qui héberge peut-être à votre insu un élément du botnet, peut également vous rendre coupable de ces activités criminelles. Mais cette application, spectaculaire, est extrêmement marginale. Il existe bien d'autres usages possibles d'un botnet :

- Envoyer des spams. Depuis que les logiciels les plus sophistiqués permettent d'assurer une gestion relativement efficace de la problématique du publipostage sauvage, il devient difficile d'envoyer des masses d'e-mails à des adresses. Le botnet permet d'utiliser tous les ordinateurs épars pour envoyer des pourriels difficiles à détecter car émis anonymement depuis des milliers d'ordinateurs d'apparence fiable.
- Escroquer des services commerciaux du Web en simulant des centaines d'utilisateurs cliquant sur des liens publicitaires, ou achetant des biens fictifs en utilisant des listes de numéros de cartes de crédit (eux-mêmes volés).
- Utiliser la puissance de calcul du botnet pour casser des mots de passe par force brute. Les méthodes de cryptage en vigueur rendent en effet de plus en plus difficile le décodage de mots de passe sur une machine isolée, mais parfois possible sur une grappe d'ordinateurs puissants.
- Héberger des contenus interdits sur un disque virtuel (organisé exactement comme un «système cloud»).

Fonctionnement du botnet

Un botnet vit plusieurs phases. La phase d'infection est la première. Elle consiste à élaborer une stratégie pour planter le bot sur un maximum de machines, puis à activer chacun des bots, c'est-à-dire à rendre le programme fonctionnel. Dans un second temps vient la phase de construction du réseau à proprement parler (le net du botnet). Le bot du PC infecté se déclare actif auprès d'une autorité centrale (un serveur) qui prend son contrôle. Finalement, la troisième et dernière étape va consister à activer les groupes de bot pour mener une opération (de spam, de cliquage publicitaire, de déni de service, par exemple).

On observera que la première phase est identique à celle d'installation d'un malware ou d'un spyware. On notera aussi que le bot peut intégrer de très nombreuses fonctions complémentaires pour se protéger, se cacher (furtivité du logiciel au sein du système hôte) ou même évoluer, par exemple *via* des mises à jour.



L'aspect le plus délicat du botnet est généralement son mode de centralisation : Microsoft, qui mène une bataille féroce contre les botnets – qui dévalorisent l'image de ses produits – a mené contre eux plusieurs opérations qui consistaient principalement à couper les voies de communication entre les bots et leurs serveurs. Lors de la destruction de Valedac (<http://www.01net.com/editorial/530303/microsoft-decapite-l-un-des-plus-gros-botnets-au-monde/>), Bernard Ourghanian, directeur technique et sécurité de Microsoft France, indiquait «[avoir] utilisé une conjonction de mesures techniques et légales afin de rompre la connexion entre la structure de commande et de contrôle du botnet et l'ensemble des machines infectées par le malware, avec pour objectif de stopper la totalité de ce botnet». La même manœuvre fut employée pour Rustock. 01Net indique qu'il «a fallu agir de façon coordonnée, afin de couper physiquement et simultanément des serveurs situés dans plusieurs États américains et pour certains à l'étranger. L'opération a notamment consisté, avec l'appui des forces fédérales américaines, à saisir des dizaines de disques durs chez cinq hébergeurs situés à Kansas City, à Scranton, à Denver, à Dallas, à Seattle, à Chicago et à Columbus». «Nous avons fait le nécessaire pour que l'on bloque l'enregistrement de domaines en Chine que le code du malware prévoyait d'utiliser dans le futur pour des opérations de contrôle», indiquait M. Ourghanian. Et, contrairement à Waledac, dont le code pointait vers des noms de domaine bloqués avec le concours de l'Icann (régulateur des noms de domaines et propriétaires des Root DNS), Rustock avait en réserve 106 adresses IP à contacter en cas de besoin, par nature plus complexes à neutraliser. On voit bien ici le degré de sophistication et les moyens mis en oeuvre par les opérateurs de botnet.

La coupure des liens est désormais contournée par les concepteurs de bots en utilisant de véritables Internet privés portés par des réseaux pair à pair de hackers, ce qui rend leur blocage en amont à nouveau très complexe (comme dans le cas de TDL4, présenté plus loin). C'est donc sur les utilisateurs que repose encore très souvent la responsabilité d'affaiblir individuellement les botnets en les éradiquant de leurs PC.

La commercialisation du botnet

Une fois installé et opérationnel, le botnet est généralement loué. Une étude menée en 2010 par Verisign Idefense (<http://labs.idefense.com/>) a permis de dresser un profil commercial du botnet. Car le botnet est avant tout un produit commercial de nature criminelle. Il est une sorte de triade virtuelle qui pourrait se louer au plus offrant, comme on se loue dans les milieux underground du hacking des identités ou des numéros de cartes de crédit valides.

Selon l'étude de Verisign, un botnet se louerait environ 67 dollars par jour et environ 9 dollars de l'heure. Le degré de criminalité des opérations menées avec un botnet peut varier du clic automatisé sur des bannières publicitaires à l'utilisation des ordinateurs hôtes pour surfer anonymement. On imagine quelles discussions peuvent être véhiculées par ce type de réseau ultrasécurisé et anonyme... sauf pour les malheureux internautes hébergeurs de bots qui offrent leurs appareils gratuitement et souvent sans le savoir.

Implantation du bot et construction du botnet

Le botnet se construit très progressivement, au gré des infections de machines. La priorité pour l'opérateur du botnet est de trouver

une méthode d'infection de PC performante, c'est-à-dire susceptible d'implanter rapidement sur des machines hétérogènes rapides le bot. En la matière, la meilleure solution est de trouver un vecteur d'implantation logiciel qui se charge de «porter» et d'installer le bot. L'une des voies les plus efficaces est de cibler les utilisateurs de jeux, réputés pour leur propension à chercher sans cesse une expérience ludique nouvelle, tout en étant équipés de machines modernes reliées par Internet chez de grands opérateurs. Ce sont des conditions idéales pour donner un haut niveau de performance au futur botnet : réseau moins protégé et maintenu que celui d'une grande entreprise (un opérateur Internet ne peut que difficilement fermer les ports de la connexion Internet de ses clients, par exemple), machine peu sécurisée. Une grande partie des jeux pirates et diffusés sur les réseaux de téléchargement pair à pair est ainsi utilisée comme vecteur d'infection. Idem pour les logiciels commerciaux déprotégés.

Dans un passé proche, la plupart des failles de sécurité multimédias et des documents du Web ont été exploitées pour véhiculer des bots. Ils utilisaient les vecteurs classiques employés par les virus et les malwares pour se propager : dépassement de piles pour charger un code malicieux, possibilité d'exécution des fichiers évolués, démarrages automatiques détournés (tel l'Autorun de Windows). Récemment, ce sont les réseaux pair à pair eux-mêmes qui ont été mis à contribution: on a ainsi parlé de TDL-4, découvert par Kaspersky et qui est un logiciel malveillant de type «bootkit» (qui comme les «rootkits» est un logiciel d'infection livré clé en main) capable d'infecter la «zone d'amorce» du disque dur (celle qui est activée automatiquement au démarrage de l'ordinateur). Lors du démarrage de l'ordinateur, le programme malveillant se lance avant le système d'exploitation, ce qui lui permet de rester invisible aux yeux de nombreuses applications de sécurité.

L'une des particularités de TDL-4 est qu'il est capable de fonctionner sur les systèmes d'exploitation récents (Vista en particulier), jusqu'à présent considérés comme sécurisés contre ce genre de virus.

Le TDL4 utilise un protocole chiffré pour communiquer avec les serveurs des personnes qui le contrôlent, et surtout le réseau Peer 2 Peer KAD, exploité par la plupart des logiciels de téléchargement illégal pour infecter d'autres ordinateurs distants.

Comment gérer une infection par bot

Votre PC est-il membre d'un botnet ? Plusieurs signaux vous indiqueront que votre PC est infecté par un bot. Trafic non justifié sur votre réseau, activité de votre machine alors qu'elle n'est pas en fonction sont quelques-uns des symptômes.

Reste maintenant à résoudre la question de l'éradication. Les premiers botnets étaient facilement localisés et neutralisés lorsqu'ils demandaient une connexion permanente vers leur serveur référent. C'est en particulier le cas des bots dialoguant via protocole IRC (*Internet Relay Chat*, utilisé dans les forums de discussion). La localisation, l'éradication et la neutralisation sont devenues plus difficiles depuis que les botnets utilisent le protocole P2P (Peer to Peer) : on se retrouve aujourd'hui souvent confronté à des bots complexes, furtifs, autoadaptatifs, très délicats à supprimer.

La première manière de se protéger des botnets consiste à prévenir plutôt que guérir. Toutes les actions préventives indiquées dans



le chapitre sur les virus seront donc adaptées et préconisées. Voilà qui va faire de la peine à certains lecteurs : ceci inclut malheureusement de se débarrasser de certains logiciels de téléchargement en pair à pair.

Ensuite, après avoir évité que votre ou vos machines ne fassent partie d'un réseau de bots, il peut être utile de mettre en oeuvre des moyens de prévention permettant sinon d'éliminer complètement le risque (la sécurité à 100 % n'existant pas), au moins de le limiter.

Quelques mesures de protection logicielles simples peuvent être mises en place :

- mettre son système à jour régulièrement, comme toujours ;
- installer un logiciel antivirus et s'assurer de sa mise à jour régulière ;
- installer un logiciel pare-feu personnel ;
- installer un outil chargé de la détection des logiciels espions (Spybot, par exemple) ;
- mettre à jour les applications logicielles (et en particulier les logiciels de bureautique, navigateurs web, clients de messagerie, clients de messagerie instantanée, lecteurs multimédias, etc.).

Dans la chaîne de l'organisation de la sécurité, en particulier en matière de bots, c'est incontestablement l'humain qui est le maillon faible. Agir sur les comportements peut considérablement limiter les risques d'infection.

Il est ainsi impératif de se montrer méfiant quant au contenu des e-mails reçus et en particulier :

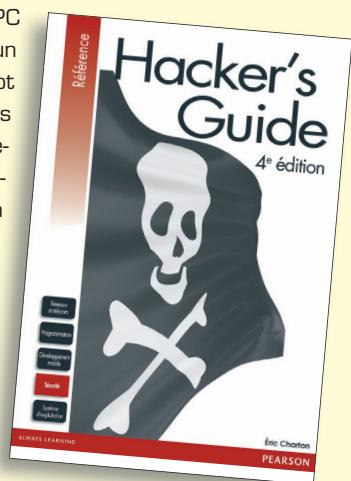
- De lire ses messages au format texte plutôt que HTML dans la mesure du possible.
- De ne jamais répondre aux messages de type spam ni suivre les liens qu'ils contiennent.
- De ne jamais exécuter de pièce jointe reçue par e-mail. Il est possible d'ouvrir les fichiers image ou document, qui sont relativement bien protégés, mais évitez à tout prix les exécutables.

Désactivez aussi les fonctions d'autoplay (lancement automatique d'applications de Windows), et évitez d'insérer dans votre machine des clés USB inconnues (l'une des méthodes d'amorçage de bot la plus performante consiste à laisser traîner des clés USB dans des conférences ou des salons).

Pour la navigation sur Internet :

- Priviliez la navigation sur des sites de confiance, des sites institutionnels. Les sites de hacking, diffusant des logiciels dont la protection est supprimée ou des torrents, sont malheureusement souvent la source des codes malveillants.
- Configurez votre navigateur de manière à limiter aux sites web autorisés l'exécution de code dynamique par votre navigateur (Javascripts, contrôles ActiveX, etc.).
- De manière générale, quel que soit le canal utilisé (IRC, Web, messagerie électronique, P2P), il faut absolument éviter de télécharger les programmes trouvés sur Internet.

Si, malgré ces précautions, votre PC est infecté, il vous faudra appliquer un ou deux antispywares tels que Spybot ou BitDefender. Sachant que, dans certains cas, des bots particulièrement bien conçus exigeront le téléchargement d'un outil de désinfection spécialisé, téléchargé sur le site du fournisseur de votre système ou par un éditeur de logiciels antivirus.



■ Extrait de **Hacker's guide**
4^e édition, de Eric Charton, Pearson, 2011

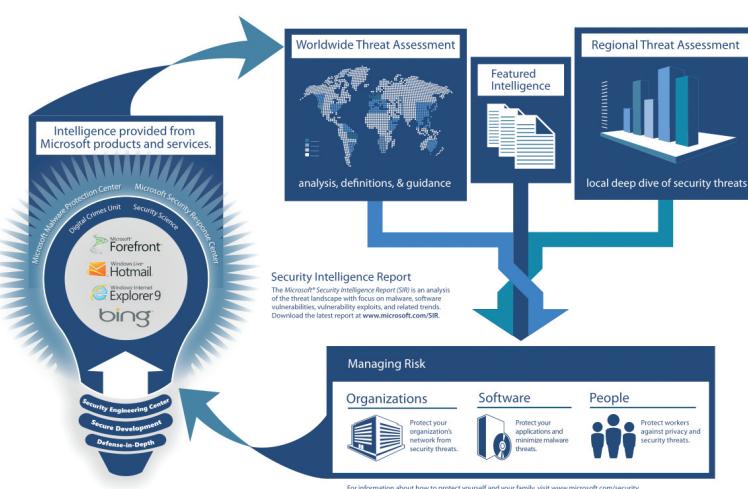
Le 11e rapport sur la sécurité de Microsoft

Les Adwares continuent de proliférer en France et infectent 72,4% des ordinateurs attaqués au deuxième trimestre 2011 contre 66,1% trois mois plus tôt. La deuxième catégorie d'attaques en France est constituée des logiciels indésirables qui touchent 27,7% des machines infectées.

Un léger recul comparé aux 31,3% du premier trimestre 2011. Enfin, les chevaux de Troie se placent en troisième position avec 12,1% des machines infectées, touchées par ce type d'attaques (15,4% au premier trimestre 2011). La vulnérabilité ne représente que 1 % des attaques.

Les attaques d'ingénierie sociale

Ces attaques représentent près de la moitié (45 %) de toutes les infections recensées dans le monde et utilisent généralement des techniques d'hameçonnage (phishing) qui exploitent des mes-



sages contrefaits afin de tromper les utilisateurs. Les attaques par phishing via les réseaux sociaux comptaient ainsi pour 83,3 % de toutes les tentatives de phishing au cours du premier semestre 2011.

Site de référence : <http://www.microsoft.com/security/sir/default.aspx>



Sécurité avancée avec PHP

La sécurité est un des thèmes récurrents de la programmation. Chaque technique ou protocole utilisé dans le développement contient un risque, comme l'utilisation du HTTPS ou la transmission de données, c'est pourquoi il est important de prendre conscience de la sécurité et des techniques liées.

La transmission de données signifie l'envoi d'informations d'un endroit vers un autre, c'est-à-dire lorsque vous validez un formulaire de saisie, ces données sont chiffrées avant d'être transmises puis envoyées à la base de données qui les récupère et les traite. La sécurisation de ces données peut s'appliquer pour l'ensemble des sites internet, quel que soit le langage, en utilisant le protocole HTTPS. Ce protocole va crypter la communication (on parle souvent de session) entre le client et le serveur, permettant ainsi de protéger la session ouverte et les informations qui y transitent. Pour mettre en place ce protocole, il est nécessaire d'utiliser TLS (Transport Layer Security) appelé avant SSL (Secure Sockets Layer), qui propose un certificat de sécurité.

L'utilisation de ces protocoles est multiple mais on peut citer les cas suivants :

- Une connexion sécurisée
- La transmission d'un numéro de carte bancaire
- Accès privé non partagée
- Ftp sécurisé

Enfin, lorsque vous souhaitez utiliser ce protocole, le but est d'avoir une adresse IP différente dédiée à la transmission.

Https et TLS/SSL ensemble

Un des premiers points importants à considérer lorsque vous souhaitez sécuriser la transmission de données, est d'associer ces 2 protocoles car il ne suffira pas juste de mettre en place dans la barre de navigation 'https://', il faut lui associer impérativement un certificat (TLS/SSL).

Au niveau de la sécurité TLS/SSL, elle est considérée comme une mesure efficace pour sécuriser les informations sensibles (attention : aucune sécurité n'est jamais infaillible ou parfaite). L'utilisation de ce protocole entre le client et le serveur se traduit par un cryptage de toutes les données y transitant. Cependant, la mise en place s'effectue avec certaines connaissances côté serveur. Par ailleurs, elle ne peut être employée partout car vous risquez de perdre au niveau de la performance de votre site si celui-ci génère beaucoup de trafic. Les protocoles de sécurité peuvent en effet peser sur les performances globales.

Mise en place

La mise en place de ce certificat de sécurité n'est pas la préoccupation première des développeurs car quel que soit votre site, statique (HTML) ou dynamique (PHP, ASP...), l'installation du certificat s'effectue d'abord du côté du serveur : configuration de Apache, IIS, Nginx, etc. Ces serveurs sont capables de supporter le protocole TLS/SSL sécurisé et de chiffrer avec des clés de 128 ou 256 bits. Pour installer ce certificat, vous pouvez l'acheter auprès de sociétés spécialisées ou sinon utiliser celui proposé par les différentes distributions Linux. Ensuite, lorsque le serveur est configuré, un fichier texte est généré, qu'il faut positionner sur le serveur.

Utilisation

L'utilisation d'un certificat TLS/SSL avec le langage PHP peut rapidement devenir inefficace s'il y a un moyen de le contourner, c'est pourquoi, vous devez vous assurer que vos utilisateurs se servent bien de l'adresse sécurisée, c'est-à-dire HTTPS et non HTTP. Sans cela, votre procédure de sécurité ne servira à rien ! Un favicon sera représenté par un petit verrou montrant que l'accès est sécurisé. Pour utiliser correctement ce protocole assurez-vous que toutes les pages HTTPS sont authentifiées et utilisées. C'est-à-dire les fichiers CSS, les images et les scripts. Si vous ne réalisez pas l'ensemble de cette opération, vous risquez un trou de sécurité, c'est pourquoi il est préférable de mettre les chemins en dur (même si cela oblige à une gestion du site plus « lourde »).

Un autre point qu'il ne faut pas oublier : la publication du lien du site web. Il est impératif de fournir une URL non sécurisée, un simple `http://`. Ne fournissez jamais une URL de type `https://`.

htaccess

Le fichier htaccess peut aussi jouer son rôle et forcer certaines fonctionnalités provenant de la barre de navigation et par conséquent forcer le mode TLS/SSL.

Listing : exemple .htaccess

```
RewriteEngine on
# HTTPS pour URL sensible
RewriteCond %{HTTPS} !=on
RewriteRule ^prive.* https://urlDuSite.com/prive$1 [R=301,L]
```

L'exemple montre que le mode HTTPS est activé et que si dans l'URL, le serveur trouve le mode 'prive' il faut utiliser un accès sécurisé. Cependant, si vous utilisez plusieurs formes d'URL, vous pouvez le générer directement par un script en PHP comme ceci :

```
<?php
if($_SERVER['SERVER_PORT'] != 443)
{
header('Location: https://urlDuSite.com/prive.php');
exit;
}
?>
```

Utilisation avec PHP

Pour utiliser PHP avec les certificats TLS/SSL, vous devez au préalable configurer votre serveur pour le rendre compatible avec ce protocole, en activant l'extension `mod_ssl`.

Lorsque celle-ci a été activée, vous devez prévoir quelques configurations au niveau du `virtualHost`, comme ceci :

```
<VirtualHost *:443>
```



```
SSLEngine On
SSLCertificateFile /etc/ssl/private/urlDuSite.pem
SSLCertificateKeyFile /etc/ssl/private/urlDuSite.key

SSLCACertificateFile /etc/ssl/private/certificat.crt
</VirtualHost>
```

Les quelques lignes ci-dessus montrent un extrait de la configuration du serveur.

Ensuite, lorsque vous utiliserez ce protocole dans votre développement, si vous n'appliquez pas complètement les règles, vous risquez de provoquer une absence de relations. Si aucune relation n'existe, un message signalant le changement de protocole sera affiché. Pour éviter cette gêne, les utilisateurs pourront obtenir automatiquement un certificat lorsqu'ils accéderont directement à la page demandée, grâce à PHP.

Ainsi, l'utilisateur pourra demander et redemander à plusieurs reprises un nouveau certificat ou encore le perdre et par conséquent en obtenir un autre facilement sans besoin de reconfirmer.

Lorsque vous utilisez le langage PHP, vous attribuez un certificat automatiquement à une nouvelle personne en utilisant certaines fonctionnalités. Les fonctionnalités pour accéder aux données sont :

- **SSL_CLIENT_VERIFY** : vérifier le certificat
- **SSL_CLIENT_M_SERIAL** : identification unique
- **SSL_CLIENT_L_*** : variables émises par le client
- **SSL_CLIENT_S_*** : variables émises par le serveur
- **SSL_CLIENT_S_DN_CN** : nom de l'utilisateur
- **SSL_CLIENT_S_DN_Email** : email de l'utilisateur

L'ensemble des fonctionnalités présentées ci-dessus permet de valider la présence d'un certificat, de l'attribuer à un utilisateur unique, avec la possibilité de contrôler les valeurs venant du client et/ou de contrôler les valeurs venant du serveur.

Cela se traduit de la façon suivante :

```
<?php
function hasValidCert()
{
    if (!isset($_SERVER['SSL_CLIENT_M_SERIAL'])
        || !isset($_SERVER['SSL_CLIENT_V_END'])
        || !isset($_SERVER['SSL_CLIENT_VERIFY'])
        || $_SERVER['SSL_CLIENT_VERIFY'] !== 'SUCCESS'
        || !isset($_SERVER['SSL_CLIENT_I_DN']))
    ) {
        return FALSE;
    }

    if ($_SERVER['SSL_CLIENT_V_REMAIN'] <= 0) {
        return FALSE;
    }

    return TRUE;
}
?>
```

Bien sûr, l'ensemble de ces informations peuvent être stockées dans une base de données pour éviter de recréer un certificat à chaque fois que la personne se reconnecte sur la page.

Au niveau des tables de la base de données, celle-ci peut être une table unique avec une jointure vers une autre table 'utilisateur' et par conséquent, la structure de la table peut se traduire de la façon suivante :

```
CREATE TABLE `utilisateur_ssl` (
  `id` INT(11) NOT NULL AUTO_INCREMENT ,
  `utilisateur_id` INT NOT NULL ,
  `ssl_serial` VARCHAR( 32 ) NOT NULL ,
  `ssl_client_issuer` VARCHAR( 1024 ) NOT NULL ,
  `ssl_name` VARCHAR( 64 ) NOT NULL ,
  `ssl_email` VARCHAR( 64 ) NOT NULL ,
  PRIMARY KEY ( `id` ) ,
  UNIQUE ( `id` )
) CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Au niveau de la requête SQL, l'opération se traduit comme ceci :

```
<?php
$sql = «
INSERT INTO `utilisateur_ssl` (`utilisateur_id`, `ssl_serial`, `ssl_client_issuer`, `ssl_name`, `ssl_email`) VALUES ($user,
'$_SERVER['SSL_CLIENT_M_SERIAL'],
$_SERVER['SSL_CLIENT_I_DN'],
$_SERVER['SSL_CLIENT_S_DN_CN'],
$_SERVER['SSL_CLIENT_S_DN_Email']
);
» ;
?>
```

L'exemple qui a été publié, est un extrait pour vous montrer le principe de fonctionnement

Les outils

Lorsque le développement a été fait, il est utile de vérifier si tout a été correctement réalisé par l'intermédiaire d'outils. Il en existe de nombreux pour vérifier si votre site web respecte bien les normes et les protocoles de sécurité. Le plus connu est 'VEGA', disponible à l'URL suivant : <http://www.subgraph.com/>

Même si la version actuelle est en bêta, ce projet possède de nombreuses fonctionnalités pour vérifier si l'ensemble des protections importantes ont bien été prises en compte.

Ces protections concernent les points principaux :

- Les formulaires
- Les données entrantes
- La validation des champs

Conclusion

L'utilisation du protocole HTTPS est très importante pour crypter la transmission d'informations sur le réseau. L'article a montré un aspect de l'utilisation des certificats, car ils peuvent aussi être mis en place pour le FTP et l'envoi d'emails.

Pour compléter, vous pouvez lire le chiffrement SSL/TLS sur http://httpd.apache.org/docs/2.3/fr/ssl/ssl_intro.html

■ 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 DrupalFR, AFUP et membre LeMug.fr, PHP-TV, PHPteam...

Windows 8

Ce qui va changer p



Le prochain système Microsoft sortira d'ici 10 – 12 mois. Cela fait quatre semaines que l'éditeur a dévoilé les premiers éléments de Windows 8, ne manquant pas de faire s'interroger les développeurs sur plusieurs points : quelle interface sera adoptée ? Quels impacts de WinRT sur le code actuel ? Faut-il migrer sur WinRT ? Peut-on concevoir un code unique pour toutes les plateformes ?

Bref, beaucoup de questions et encore peu de réponses claires. Nous ne sommes qu'au début du long processus menant à Windows 8. Microsoft devra préciser de nombreux éléments et fournir de nouvelles prévisions du système et des outils de développement.

Dans ce premier dossier, nous avons exploré pour vous les entrailles de Win8, de WinRT et des nouvelles interfaces. Et tout ne sera pas simple !

■ François Tonic

The image shows a Windows 8 Start screen. The tiles include:

- Mail**: A green tile showing an email from mattberg@live.com with the subject "RE: Wedding Announcement!" and a snippet of the message.
- Internet Explorer**: A blue tile with the IE logo.
- Store**: An orange tile with a white shopping bag icon.
- Messaging**: A green tile with a white speech bubble containing a smiley face.
- Weather**: A blue tile showing the current temperature (78°) and forecasts for Tomorrow, Thursday, and Friday.

Avertissement !

Windows 8 est en Developer Preview, une version préliminaire du système. Les descriptions et modèles de développement décrits dans les articles peuvent changer ou être supprimés dans les futures versions.

pour le développeur

Ben
Martens



Surfboard kickflip
in Santa Cruz

Now Playing
a
er Creek Club
ers

Anaheim, CA

88° / 65°

87° / 65°

/ 65°

Make awesome apps
BUILD conference
9:00 AM - 11:00 AM



13
TUESDAY

Photos

6

Investments ▲ 2.3%

- ▲ 3.6% Roth IRA (TD Ameritrade)
- ▲ 2.7% 401K Fund (Fidelity)
- ▼ 1.4% Investment Portfolio



Video



People



Control Panel



Wind
Explor

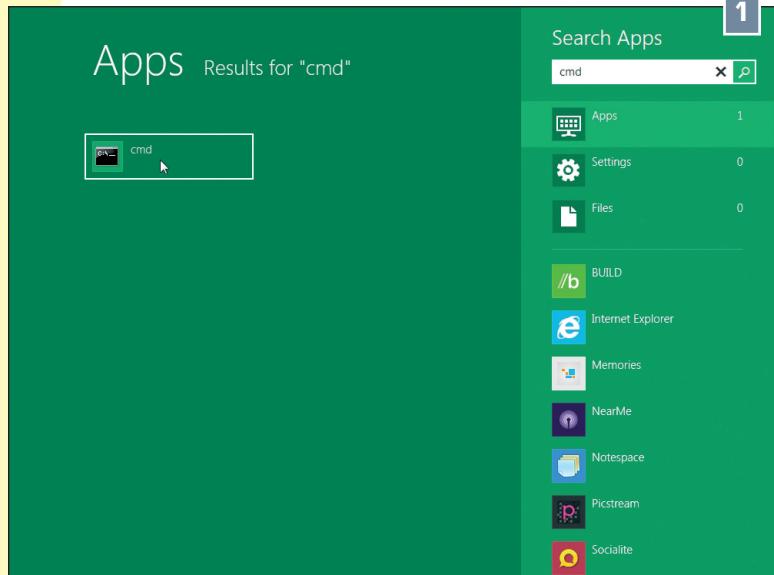


Desktop

Désactiver l'interface Metro et retrouver le menu démarrer classique avec Windows 8

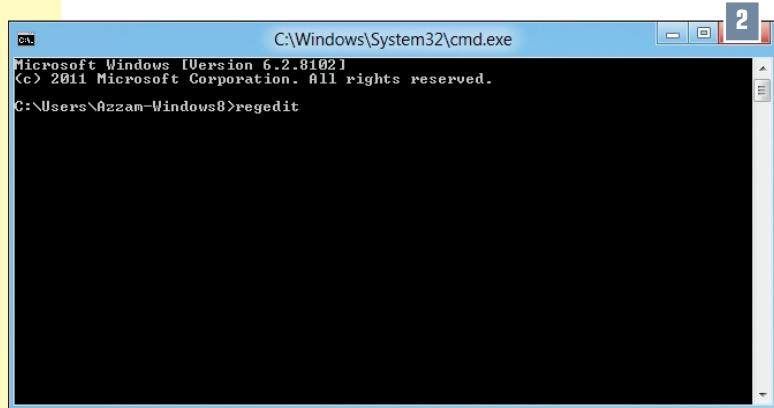
Vous avez installé Windows 8 Developer Preview l'interface Metro, c'est sympa mais vous trouvez que ce n'est pas pratique quand vous l'utilisez avec la souris, en même temps vous êtes habitué au menu démarrer. Il y a un moyen pour désactiver l'interface Metro et retrouver le menu démarrer classique.

ATTENTION : Cette opération n'est pas approuvée par Microsoft.

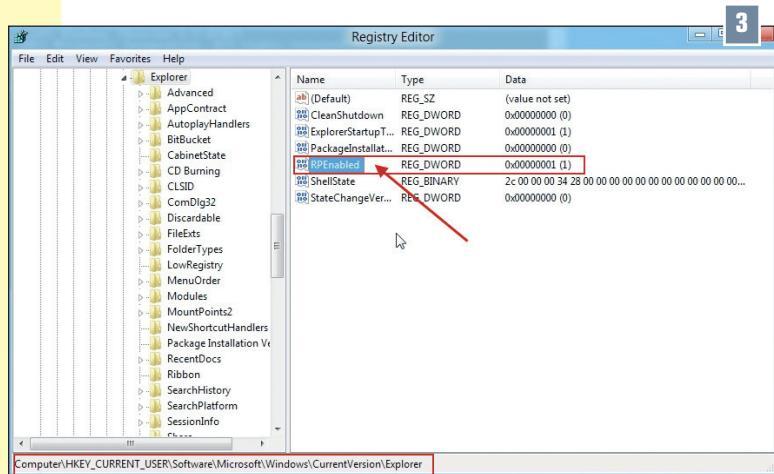


- 1 A partir de l'interface Metro, tapez cmd, vous allez voir Windows commence la recherche pour cmd
- 2 Cliquez sur cmd, l'invite de commandes est ouvert, tapez regedit pour lancer l'éditeur du Registre
- 3 Allez à HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer Double-cliquez sur RPEnable
- 4 Changez la valeur de 1 à 2
- 5 Fermez l'éditeur du Registre, Vous allez voir le menu démarrer classique, et au démarrage du Windows 8, l'interface Metro est désactivée.

Pour revenir à l'interface Metro et le menu démarrer du Windows 8, changez la valeur de RPEnable de 2 à 1



- Azzam ALNIJRES - Club Windows 8
www.club-windows8.com



(Au cœur de) Windows 8

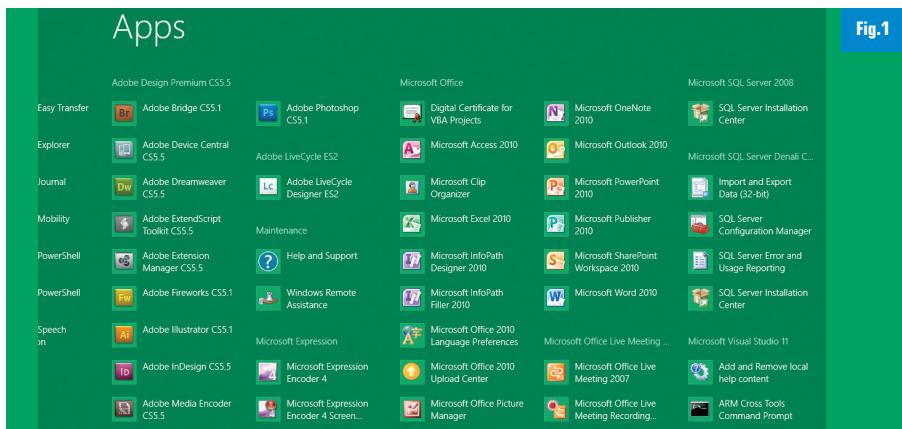
1^{re} partie

Voilà maintenant plus d'un mois que l'évènement majeur des développeurs Microsoft, la Build, a eu lieu et les réactions vont bon train sur la toile. Le plus intéressant reste toutefois la provenance de ces réactions. Effectivement, afin d'intéresser la communauté à la mise en place de Windows 8, les équipes de Steven Sinofsky ont créé un blog dédié à la nouvelle plateforme où chacun est libre de venir donner son avis quant aux décisions prises à Redmond (<http://blogs.msdn.com/b/b8/>).

D'aucuns pourraient se dire qu'il s'agit d'une initiative vouée à l'échec et que l'éditeur n'écouterait pas les retours des internautes. Les évènements récents tendent à démontrer le contraire. Les réactions fusent, notamment autour de la nouvelle interface « Metro ». Beaucoup d'utilisateurs se sont plaints de ne plus s'y retrouver dans cette approche immersive et commençaient à regretter leur menu démarrer. Qu'à cela ne tienne, la réponse de Steven Sinofsky ne s'est pas faite attendre en publant un billet dans lequel il répond ouvertement aux utilisateurs en citant leurs remarques et les orientations prises par la firme afin de faire comprendre les choix effectués. Je ne peux que vous conseillez de le consulter : <http://blogs.msdn.com/b/b8/archive/2011/10/11/reflecting-on-your-comments-on-the-start-screen.aspx>. On y apprend notamment que les remarques des utilisateurs ont été prises en compte et que les équipes travaillent actuellement à une révision de l'écran « Apps » afin de faire un « mix » entre le mode métro et la hiérarchisation des menus type Windows 7 : [Fig.1]. On remarque bien la nouvelle présence de groupes permettant de classer les applicatifs, cela ressemble déjà un peu plus à notre façon de rechercher et de trouver une application via le menu démarrer actuel. C'est au travers des différents billets de ce blog, des documentations fournies par Microsoft et d'un mois de travail acharné sur l'environnement que nous allons essayer de vous présenter le cœur de Windows 8 en quelques points fondamentaux.

Gestion de la mémoire

La volonté de Microsoft de se positionner comme acteur majeur dans le monde des tablettes tactiles avec sa mouture Windows 8 force à la réflexion sur le cœur même du système. Effectivement, même si les tablettes se font de plus en plus puis-



santes, l'un des enjeux réside dans la durée maximale de vie de la batterie. Aussi, il est important de prendre en compte que la consommation mémoire a un impact direct sur cette dernière. A cet effet, Microsoft a repensé en grande partie la gestion de la mémoire pour Windows 8. L'une des annonces faites lors du keynote de la build porte sur les prérequis système qui seront les mêmes que ceux annoncés pour Windows 7. Il faudra donc au minimum 1Go de RAM afin de faire fonctionner Windows 8 [Fig.2 et 3]. Cette amélioration repose sur des modifications au cœur de Windows 8, voici une liste non-exhaustive de certains concepts qui ont amélioré grandement la consommation mémoire.

La combinaison de zone mémoire

Tout d'abord, le « Memory Combining ». Une analyse poussée de la RAM montre que certaines allocations faites par le système d'exploitation, les services et les programmes sont les mêmes. Cela se produit notamment lors des allocations anticipées qui ne seront pas forcément utilisées si elles ont été faites pour une fonctionnalité dont l'utilisateur ne se servira pas. A cet effet, Windows 8 détectera ce genre d'allocations sur l'ensemble du système et se chargera de libérer les redondances tout en gardant une seule et unique copie. Si

toutefois l'applicatif qui avait pré-alloué l'une des redondances tente d'écrire dans cette zone, le système se chargera de lui fournir une copie privée. Tout ceci étant géré nativement par le système, il n'y aura aucun impact dans le code applicatif.

Modifications apportées aux services Windows

Le second travail effectué par les équipes porte sur la gestion des services Windows. Tout d'abord, 13 d'entre eux ont été tout simplement supprimés. Ensuite, un ensemble de services est passé en mode manuel en place du mode automatique de démarrage. Mais il semble que ces opérations n'étaient pas suffisantes et un nouveau mode de démarrage de services a donc fait son apparition, le mode « start on demand ». Ce nouveau mode se base sur du déclenchement à partir d'évènements système, par exemple l'apparition d'un nouveau périphérique. A cette notification, le service démarre, fait son travail, réside un laps de temps en mémoire afin d'être sûr qu'il n'y a pas d'autres tâches à effectuer, puis s'éteint purement et simplement.

Chargement différé du bureau

Un autre travail effectué porte sur le chargement différé du mode bureau tel que nous le connaissons sous Windows 7.

Fig.2

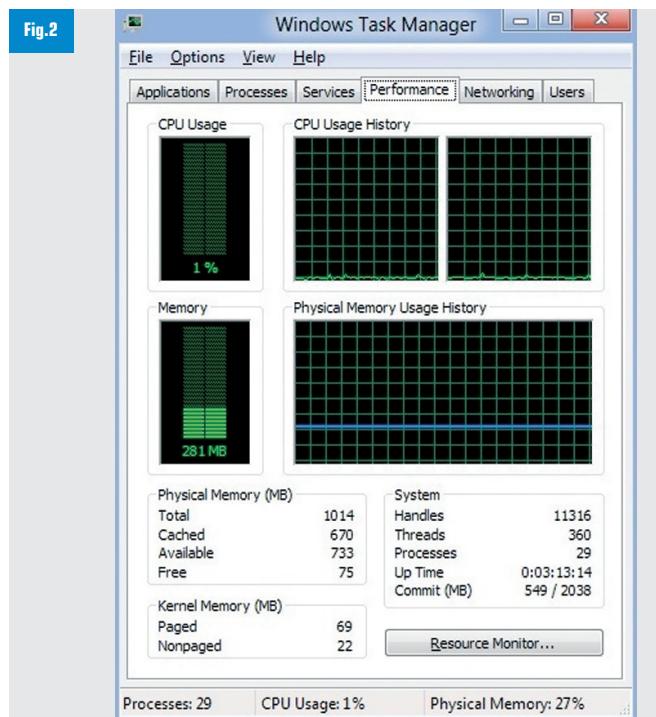
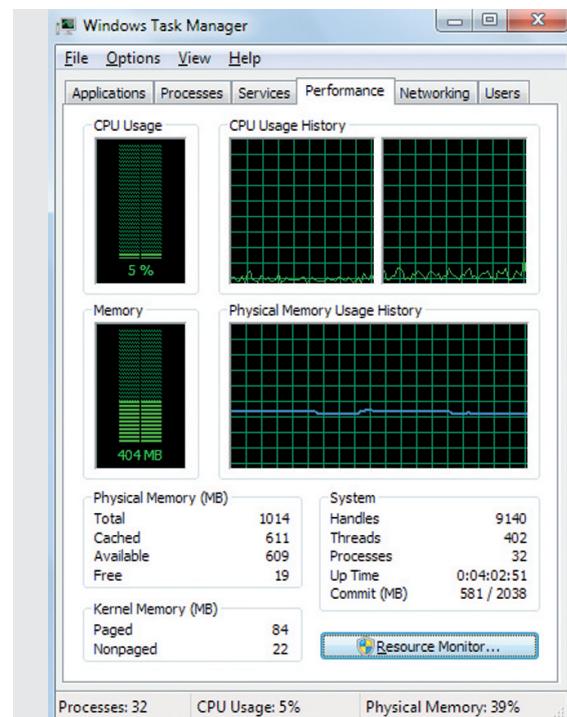


Fig.3



Cette initiative très discutable si nous nous plaçons d'un point de vue « entreprise » où l'interface Metro doit encore faire ses preuves. Effectivement, cette réflexion part du postulat que Windows 8 sera majoritairement consommé en mode immersif au travers de son nouvel écran de démarrage Metro. Avec ce point de vue, il est acceptable de charger le bureau en arrière-plan de façon différée afin de limiter encore plus la consommation mémoire. D'un point de vue « entreprise » où la majorité des opérations se feront via le bureau traditionnel, cette décision semble somme toute discutable si toutefois l'attente de chargement différé se fait trop longue.

Granularité et priorité mémoire

Pour finir, traitons la notion de granularité. A ce jour, le système Windows 7 ne fait pas de différence de priorité sur la mémoire, que ce soit au niveau du système ou bien des programmes. Il se peut donc que le système décide, lors d'une saturation de la mémoire, de récupérer des allocations appartenant à un autre processus pour exécuter les tâches nécessaires. Par exemple, un anti-virus résidant en mode protection temps réel pourrait voler des allocations à Excel pour faire ses traitements. Avec Windows 8, une notion de priorité apparaît et permettra de définir une granularité « faible » sur certaines allocations. Dans le cas d'une saturation de la mémoire, ce sont ces zones qui seront

récupérées en priorité. Si cela est bien géré par les différents éditeurs, nous pouvons envisager une meilleure expérience utilisateur, le cas échéant où tout le monde se définirait en priorité dite « haute », cela ne changerait absolument rien. Une initiative intéressante si toutefois le marché implémente cette fonctionnalité correctement.

La sécurité

Afin d'améliorer la qualité de son nouveau système d'exploitation, la sécurité est encore et toujours au cœur du débat. Pour cela, Microsoft propose plusieurs nouveautés dont le « secure boot » basé sur la norme UEFI. « La norme Unified Extensible Firmware Interface (UEFI, « interface micrologicielle extensible unifiée») définit un logiciel intermédiaire entre le micrologiciel (firmware) et le système d'exploitation (OS) d'un ordinateur. Cette interface succède sur certaines cartes-mères au BIOS. (Source : Wikipedia) ». Comprenez donc par-là que le démarrage sécurisé est bien un protocole UEFI et non une fonctionnalité de Windows 8. L'OS se contente de faire appel au démarrage sécurisé afin de garantir la sécurité de l'environnement.

Au-delà de l'implémentation de la norme UEFI, le travail de Microsoft s'est aussi concentré sur la notion d'attaques malveillantes. Dans la continuité du travail fourni depuis Windows Vista, les équipes se forcent à utiliser le processus « Security Development Lifecycle » (<http://www.microsoft.com/security/sdl/default.aspx>) afin de garantir une protection maximale du système d'exploitation. Pour cela, de nombreuses améliorations ont été effectuées.

Tout d'abord, une modification de l'algorithme aléatoire de l'ASLR (Address Space Layout Randomization) apparue avec Windows Vista, dont le rôle consiste à modifier les emplacements mémoire du code et des données. De cette façon, il est difficile pour un « malware » de connaître à l'avance les emplacements mémoire qu'il peut attaquer. De plus, certaines protections qui étaient auparavant uniquement utilisées pour les programmes sont maintenant partie intégrante du noyau. Notamment l'interdiction de l'allocation des premiers 64Ko de la mémoire des processus en mode utilisateur permettant les attaques de type « déréférencement de NULL ». Toutes ces améliorations restent cependant techniques mais il est aussi à remarquer que « Windows Defender » sera maintenant installé par défaut si toutefois le système ne propose pas d'alternative d'anti-virus. Cette modification grandement orientée grand public viendra renforcer la protection déjà proposée par l'UAC (User Access Control) pour les utilisateurs non avertis.

Le développement

Passons maintenant à la partie développement. L'arrivée de Windows 8 apporte un grand nombre de nouveautés, sans même parler de l'arrivée du Framework .NET 4.5.

Nous allons ici nous attarder sur les améliorations proposées aux développeurs d'applications dites « desktop » tant du point de vue « Système » que « Technologies ».

Développement « Système »

Entrons dans le vif du sujet avec la nouvelle version du « Windows Driver Kit » qui fournit les versions 1.11 de « Kernel-Mode Driver Framework (KMDF) » et « User-Mode

Driver Framework (UMDF) ». Ceci étant très technique, nous ne pouvons que vous conseiller de vous référer aux spécifications (<http://msdn.microsoft.com/en-us/library/windows/hardware/hh451227%28v=VS.85%29.aspx>) afin d'entrer dans les détails des nouvelles fonctionnalités. Nous noterons tout de même le support du « system-mode DMA » sur les SoC (System on a Chip) pour les drivers en mode « Kernel » ainsi

que la capacité de création de mini-drivers HID (Human Interface Device) pour ceux en mode « User », sans oublier l'ajout du support de la spécification USB 3.0 en mode « Kernel ». A cela vient s'ajouter la capacité, pour les fournisseurs de périphériques, de fournir une application « Metro » s'installant automatiquement en mode silencieux afin de fournir une expérience immersive à l'utilisateur.

La liste des modifications dites de développement « Système » est longue !

Il est difficile d'en faire une liste exhaustive.

Nous ne pouvons cependant pas passer à côté de ce que nous pensons être un revirement de situation avec l'ajout du support d'OpenXPS en place de MSXPS. Pour rappel, OpenXPS est une spécification ECMA (<http://www.ecma-international.org/publications/standards/Ecma-388.htm>) concernant « OpenXML Paper Specification ». Les premières implémentations de Microsoft au sein de son système d'exploitation reposaient sur MSXPS, une version proposée par l'éditeur. Cependant, à la suite de différentes remarques des constructeurs et développeurs voulant reposer sur une norme moins assujettie aux modifications intempestives de l'équipe Microsoft, ces derniers ont donc répondu par la mise en place de la norme ECMA au sein de Windows 8 en proposant la mise à disposition de « guidelines » sur la mise en place d'applicatifs et sur les drivers d'impression pour l'utilisation de ce format.

Développement « Technologies »

Passons maintenant à la partie « Technologies » ou API mise à disposition au sein de Windows 8. Cette partie fera abstraction de « Windows Runtime » et des applicatifs dits « Metro » afin de se concentrer sur les nouveautés et améliorations des API du système. Nous tenterons ici d'en faire une liste exhaustive telle qu'annoncée par Microsoft dans ses documentations.

App Packaging : Cette API permet aux développeurs de créer ou d'interagir avec des « app packages » basés sur le format OPC (ECMA-376). Effectivement, les applicatifs « Metro » étant distribués au format OPC via des fichiers APPX, il était impensable que le système ne propose pas une API permettant d'interagir avec ce dernier.

Assessment Execution Engine : De son petit nom AXE, ce système met à disposition un ensemble d'outils permettant la mise en

place de systèmes d'évaluations tant sur le plan fonctionnel que sur les performances. Ces outils permettront de mettre en place les tests d'évaluations, de capturer les mesures et d'analyser les résultats produits lors de ces derniers.

Compression : Cette API expose les algorithmes de compression MSZIP, XPRESS, XPRESS_HUFF et LZMS utilisés par le système. Nous y retrouverons notamment la capacité de compresser ou de décompresser des données avec les algorithmes natifs de Windows 8.

Data Deduplication : A destination des développeurs C/C++, cette API permettra aux applicatifs gérant des systèmes de backup de profiter du système optimisé de déduplication.

DirectComposition : Ce composant Windows permet la composition de « bitmap » avec transformations, effets et animations à un très haut niveau de performance. Opérant indépendamment du thread de l'interface, il permet de mettre en place des transitions riches et fluides en utilisant le GPU. Acceptant tous types de « bitmap » produit par GDI, DirectX ou HWND, son rôle consistera en l'orchestration des animations et transitions à haute performance.

DirectXMath : Cette bibliothèque implémente une interface optimale et portable pour les opérations arithmétiques et d'algèbre linéaire sur les vecteurs « floating-point » (2D, 3D et 4D) ou les matrices (3x3 et 4x4). Fournissant de nouvelles fonctionnalités, elle a pour vocation le remplacement, par exemple, de D3DX 10 Math ou encore de Xbox Math.

Help API : Cette API permet au développeur de créer un système d'aide qui reposera sur le nouveau « Windows Help Runtime ». Elle permettra notamment de chercher, naviguer au sein d'un catalogue d'aide.

Operation Recorder : Utilisable en C/C++, cette API a pour but de mettre à disposition le système de « prefetching » en dehors du

contexte de démarrage d'une application. Il s'avérera pertinent sur des applicatifs accédant de manière répétée et intensive aux mêmes données.

Web Socket Protocol Component API : Le but de cette API est de proposer le support d'une communication bidirectionnelle asynchrone sur le protocole http entre deux serveurs intermédiaires et que ces derniers soient capables de revenir sur le protocole sous-jacent (TCP / SSL) pour la communication interne.

XAudio2 : Remplaçant attendu de ses prédecesseurs DirectSound et XAudio, cette API bas niveau de gestion audio a surtout pour vocation le développement dans le monde du « gaming ».

XInput : Encore dédiée au monde du « gaming », cette API facilite l'interaction avec le joystick de la Xbox 360 quand ce dernier est connecté à un système Windows.

Conclusion

En sus de sa stratégie Windows 8, Microsoft pense aussi à ses développeurs. Cette nouvelle version ne s'accompagnera pas seulement d'une nouvelle interface dédiée aux tablettes tactiles, elle sera aussi enrichie d'un nouveau jeu d'API permettant encore et toujours plus de contrôle sur le système en lui-même. L'annonce d'un timing « Quality Driven » par Steven Sinofsky quant à la sortie de Windows 8 laisse rêveur et dubitatif à la fois. Effectivement, l'éditeur prendra-t-il le risque de retarder la sortie de son nouveau système d'exploitation tant que la qualité n'est pas au rendez-vous, tout en laissant l'opportunité à la concurrence de s'inspirer des idées proposées lors de la build ? La réponse semble être dans la question et il y a fort à parier que Microsoft mettra les moyens pour atteindre ce but dans les plus brefs délais.

■ Aurélien Verla - CTO Wygware

(Au cœur de) WinRT

2^e partie

Windows 8 arrive avec un nouveau modèle de développement entièrement dédié à la mise en place d'applicatifs « Metro ». Ouvrant ses possibilités en termes de langage, la plateforme Windows 8 tente par là-même d'intéresser une population de développeurs n'ayant pas forcément d'affinités avec le monde Microsoft. La preuve flagrante, qui nous l'espérons sera une meilleure tentative que la barre de Gadget apparue avec Windows Vista, est l'apparition du triplet HTML5 - CSS3 - JavaScript. La présence du couple C/C++ - XAML semble aussi étonnante que pertinente pour l'intégration d'applicatifs « Metro » avec DirectX et HLSL (High Level Shader Language). [Fig.1]

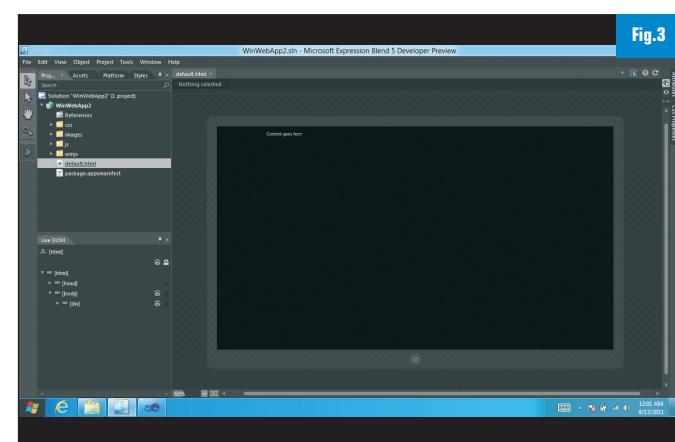
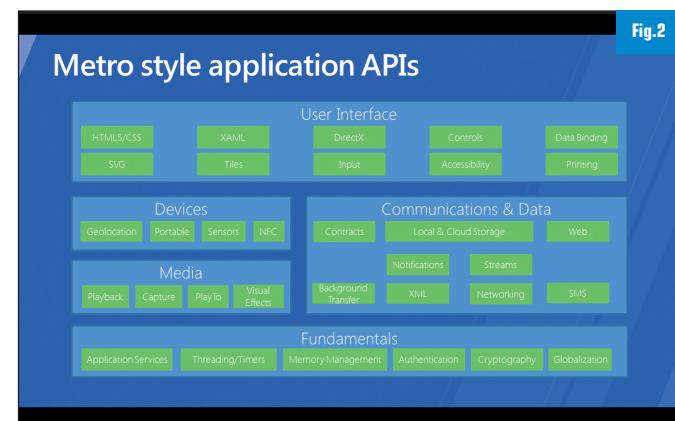
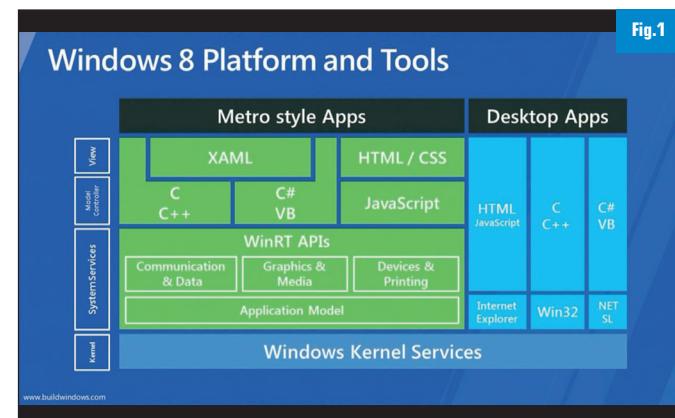
Nous allons traiter ici du « Windows Runtime » ou autrement nommé WinRT, nouvelle brique imposée pour les applicatifs « Metro ». Le but de cette API est de fournir au développeur l'ensemble des fonctionnalités Win32 nécessaires à la mise en place d'applicatifs « Metro » tout en s'assurant de la maîtrise de la performance et de la sécurité au sens système du terme. Disponible au sein de tous les langages par un système de projection, WinRT n'est qu'un sous-ensemble des fonctionnalités du Kernel. Cependant, WinRT sera amené à évoluer avec le temps si toutefois la mise en place de certains scénarios applicatifs n'avait pas été pensée à ce jour.

STRUCTURE DE WINRT

WinRT est exposé par métadonnées au travers de fichiers « *.winmd ». Sa forme et sa structure peuvent être manipulées tant par des langages statiques comme C# que des langages dynamiques comme JavaScript. Ceci marque une grande différence avec les anciennes versions dans lesquelles l'accès à l'API native n'était possible que par C / C++ ou au travers de « wrappers ». Dorénavant, les nouvelles API sont publiées au travers du système de métadonnées et donc l'ensemble des classes, interfaces, propriétés et événements sont directement projetés vers les langages supportés. Le point intéressant de cette projection est qu'elle s'adapte au langage ciblé. Par exemple, la création d'une instance via WinRT qui utilise la fonction COM CoCreateInstance sera complètement masquée par l'utilisation du mot clé « new » en C#. Pour en revenir aux métadonnées, chaque fichier « *.winmd » expose un ou plusieurs espaces de noms groupés par fonctionnalités [Fig.2]. Voici un extrait des espaces de noms disponibles à ce jour au sein de WinRT.

Windows.ApplicationModel : Gère le démarrage, l'activation, la mise en veille et le redémarrage des applicatifs. Il met aussi à disposition les classes nécessaires à la gestion des Contacts ou bien encore le système de communication entre deux applicatifs.

L'assimilation de cette partie de WinRT est primordiale pour la bonne gestion de votre cycle applicatif. Toute application, une fois démarlée, reste à disposition de l'utilisateur, qu'il la consulte ou non. La gestion d'un état « suspendu » permet à Windows 8 de libérer le CPU et d'assurer que l'application n'en utilisera pas tant qu'elle ne sera pas activée mais elle reste tout de même en mémoire. Si toutefois la mémoire venait à manquer, les applications dites « suspendues » seront fermées en priorité. Comprenez



donc par là qu'il est important de gérer la sauvegarde de ses données dès que l'applicatif passe en mode « suspendu » car rien ne laisse présager qu'il repassera au statut activé. A cet effet, le moteur laisse un laps de temps au moment du changement d'état afin d'effectuer ces opérations de sauvegarde.

Windows.Data : Gère les données au format **JSON** et **XML**.

Windows.Devices : Permet de gérer les périphériques tels que les capteurs ou bien la caméra.

Windows.Foundation : Met à disposition les fonctionnalités fondamentales de WinRT comme la lecture / écriture en mode asynchrone, la gestion de collections et bien d'autres.

Windows.Globalization : Comme son nom l'indique, cet espace de nom facilitera la mise en place d'applicatifs internationalisés.

Windows.Graphics : Support pour les éléments graphiques.

Windows.Management : Sert à la gestion des packages *.appx

Windows.Media : Gestion audio et vidéo.

Windows.Storage : On dénotera notamment dans cet espace de nom la présence du FileOpenPicker.

Windows.UI : Fournit un Framework de composant et une API pour définir les interfaces utilisateur.

Windows.UI.Xaml : Fournit un Framework de composant et une API pour utiliser XAML afin de définir les interfaces utilisateur.

Windows.Web : Permet de gérer les flux de syndication et d'accéder à des ressources en utilisant le protocole AtomPub.

Les autres espaces de noms parlant d'eux-mêmes, il n'est pas nécessaire de fournir une description pour comprendre leur fonctionnement. Voici donc leur liste : Windows.Networking, Windows.Security, Windows.System.

Cependant, WinRT ne s'arrête pas uniquement à cette API système. Prenons par exemple le gap existant entre XAML et HTML/CSS en termes de création d'interfaces. Qu'en est-il des animations, des contrôles « riches » de type ListView, du Binding et consorts ? La réponse à cette problématique se nomme « Windows Library for JavaScript » autrement nommé WinJS, dont vous trouvez la référence complète à l'adresse suivante : <http://msdn.microsoft.com/en-us/library/windows/apps/br211669%28v=VS.8.5%29.aspx>.

Ce Framework contenant un Framework CSS, entendez par là un « guideline » de classes CSS à utiliser pour respecter le style « Metro », un Framework JavaScript permettant de combler les différences entre XAML et HTML afin de produire le même type de rendu applicatif tout en tendant vers le même niveau de productivité grâce à son intégration complète au sein de « Microsoft Expression Blend 5 » [Fig.3].

On remarque bien la volonté de Microsoft, qui au travers de cette initiative, tente de convaincre tout développeur qui aurait touché de près ou de loin au monde du « web » de venir produire des applications Windows 8. Cependant, il faut rester vigilant. Etant moi-même un « adorateur » de JavaScript, j'en connais surtout les méfaits. La puissance de calcul est bien loin de ce que peut offrir du C# / VB ou du C++ et malgré l'utilisation d'Internet Explorer 10 et de ses capacités d'accélérations matérielles, il est évident que certaines applications basées sur ce modèle « traîneront la patte » vis-à-vis des autres possibilités de la plateforme.

Attention, n'entendez pas par là qu'il ne sera pas possible de faire des applicatifs performants et réactifs au travers du triplet HTML5 / CSS3 / JavaScript, le doute émis porte surtout sur le fait qu'il est, à mon humble avis, beaucoup plus facile de faire une

« mauvaise application » avec ce dernier qu'avec les autres. Il semble d'ailleurs que Microsoft soit conscient de cette problématique à la vue de cette publication entièrement consacrée à la « bonne » utilisation de JavaScript au sein des applicatifs « Metro » : <http://msdn.microsoft.com/en-us/library/windows/apps/hh465172%28v=VS.85%29.aspx>.

<http://msdn.microsoft.com/en-us/library/windows/apps/hh465172%28v=VS.85%29.aspx>.

LES CONCEPTS DE PROGRAMMATION

Le principe d'activation

Créer une nouvelle instance d'un objet WinRT est nommé le processus d'activation. Allant bien au-delà d'un simple « new », ce dernier a à charge de vérifier si vous pouvez accéder aux données utilisateurs ou bien à la ressource demandée. Si toutefois l'accès demandé nécessite des droits supplémentaires, ces derniers pourront être définis au travers du manifeste de l'applicatif. C'est la garantie pour l'utilisateur final qu'une application fait réellement ce qu'elle annonce. Autre point important, il n'est pas nécessaire de gérer le cycle de vie de l'objet sous-jacent, Windows se chargera de libérer l'objet ainsi que l'instance l'ayant activé.

Le fonctionnement asynchrone

Un applicatif « Metro » doit répondre au toucher ou à toute autre manipulation de façon instantanée. A cet effet, le thread de l'interface doit toujours être réactif afin de garantir la meilleure expérience utilisateur. Pour cela, WinRT utilise le fonctionnement asynchrone. Toute opération qui dépasse les 50 millisecondes d'exécution a donc été implémentée de façon asynchrone au sein de WinRT. Le schéma général supporté par WinRT est de récupérer le résultat de l'appel asynchrone en utilisant les méthodes « Completed » et « GetResults ». Il faut aussi noter qu'il sera possible d'utiliser la syntaxe « `async / await` » proposée par C# 5. Côté JavaScript, l'implémentation de la méthode « `then` » permettra de faciliter la mise en place du code asynchrone de manière élégante se basant sur la proposition de standard Common JS Promise/A (<http://wiki.commonjs.org/wiki/Promises/A>).

Voici, par exemple, comment créer un fichier de façon asynchrone en C# :

```
using System.Diagnostics;
using Windows.Storage;
using Windows.Networking.BackgroundTransfer;
async Task CreateTempFileAsync()
{
    // Récupération du répertoire temporaire
    IStorageFolder tempFolder = Windows.Storage.Application
Data.Current.TemporaryFolder;

    // Création du fichier en mode asynchrone.
    var tempFile = await tempFolder.CreateFileAsync("programmez.txt");

    Debug.WriteLine("Fichier créé : " + tempFile.Path);
}
```

Et son pendant en JavaScript utilisant la syntaxe `then()` :

```
function createTempFileAsync()
{
    // Récupération du répertoire temporaire
    var tempFolder = Windows.Storage.ApplicationData.current.
temporaryFolder;
```

```
// Création du fichier en mode asynchrone.  
// Puis affichage sur le retour asynchrone.  
tempFolder.createFileAsync("programmez.txt").then(function  
(tempFile) {  
    console.log("Fichier créé : " + tempFile.path);  
});  
}
```

Les capacités

Point important du respect de la vie privée et bien connu par les développeurs Android ou iPhone, la notion de capacité fait aussi son apparition au sein des applicatifs « Metro ». Par défaut, un applicatif « Metro » ne peut pas accéder à des données de l'utilisateur ou du système sans le déclarer au sein de son manifeste. La seule garantie assurée est celle d'accéder au CPU pour l'exécution de son code ainsi qu'aux périphériques d'interaction avec l'utilisateur tel que le « touch » ou le clavier.

Pour toute autre opération nécessitant un accès au système de fichier, au réseau ou toute autre fonctionnalité protégée, vous devrez le déclarer. La majorité des validations de l'utilisateur concernant les capacités de l'application se feront lors du cycle d'installation, mais il se peut que certaines demandes se fassent lors de l'exécution de l'applicatif.

Par exemple, un accès programmatique aux librairies de l'utilisateur est protégé.

Comprenez par librairies la notion de documents, musiques, images et vidéos. Autrement, vous devrez vous contenter du « File Picker » disponible en tant que contrôle WinRT. Il en va de même pour l'accès au microphone ou à la webcam.

A cette liste, vous devez ajouter les disques externes, la géolocalisation, la notion de proximité, les systèmes de messagerie instantanée type MSN, les SMS, les informations relatives à la connexion sur un domaine ou encore la notion d'Internet public, travail ou maison qui pourrait être intéressante pour les systèmes « peer-to-peer » par exemple.

Les contrats

Les contrats permettent aux applicatifs « Metro » de partager de l'information avec d'autres applicatifs ou avec le système en lui-même. A ce jour, 5 contrats sont disponibles.

Le contrat « Search » : Permet à votre applicatif de fournir des résultats directement intégrés à la fonctionnalité de recherche de Windows 8.

Le contrat « Share » : Permet de partager du contenu produit par votre application avec d'autres applications. A ce jour, le contenu peut être une photo, un lien ou du texte produit localement ou en provenance d'un réseau social.

Le contrat « PlayTo » : Permet la lecture de vidéo, audio ou image directement sur un périphérique connecté à Windows 8 par un simple bouton.

Le contrat « Settings » : Permet de gérer les préférences utilisateur de votre applicatif de façon intégrée au modèle de paramétrage de Windows 8.

Le contrat « App to App Picking » : Permet à votre application d'accéder à de l'information qui appartiendrait à une autre application. Il est tout aussi facile pour une application de déclarer et d'utiliser ces contrats

Les composants

Il est possible de créer des bibliothèques réutilisables avec les langages C#, VB et C++. Là où cela devient intéressant, c'est que ces bibliothèques peuvent être utilisées avec n'importe quel langage supporté par la plateforme « Metro ». Il est donc possible d'utiliser une bibliothèque C# directement au travers de JavaScript comme si vous consommiez un autre objet WinRT. L'ensemble des composants est donc packagé avec l'applicatif. Il y a cependant certaines règles à respecter :

- Votre projet C# / VB / C++ doit avoir comme type de sortie « WinMD File ».
- Les membres, paramètres et valeurs de retour de vos types et méthodes publiques doivent obligatoirement être un type WinRT.
- Une classe ne peut pas être « Generic ».
- Une classe ne peut pas implémenter une interface qui ne serait pas WinRT.
- Une classe ne peut pas utiliser un constructeur « protected ».
- Une surcharge du type « Object » autre que sur la méthode « ToString() » est interdite.
- Tous les types publics doivent être dans un espace de nom qui correspond au nom de votre assembly et qui ne peut pas commencer par « Windows ».

Une dernière règle s'applique uniquement lorsque le projet utilisant votre composant est en JavaScript :

- Votre classe doit être obligatoirement « sealed » (ou « NotInheritable » en VB).

Ce qui est intéressant, c'est la facilité d'utilisation une fois le composant projeté. Voici un petit exemple illustrant ce propos.

Partie composant C# :

```
using System;  
  
namespace Programmez.Utils  
{  
    public sealed class StringHelper  
    {  
        public static string Reverse(string s)  
        {  
            return new String(Array.Reverse(s.ToCharArray()));  
        }  
    }  
}
```

Partie JavaScript « Metro » :

```
alert(Programmez.Utils.StringHelper.reverse("Programmez"));
```

CONCLUSION

Ce petit tour du « Windows Runtime » nous montre bien la volonté d'ouverture et d'accessibilité de la part de Microsoft pour son prochain opus. La mise à disposition de pas moins de 199 exemples d'utilisation des différentes fonctionnalités liées aux applicatifs « Metro » renforce cette volonté de prise en main rapide de la plateforme par la communauté de développeurs. Voir <http://code.msdn.microsoft.com/Windows-Developer-Preview-6b53adbb>.

■ Aurélien Verla - CTO Wygwam

Ma première application Metro

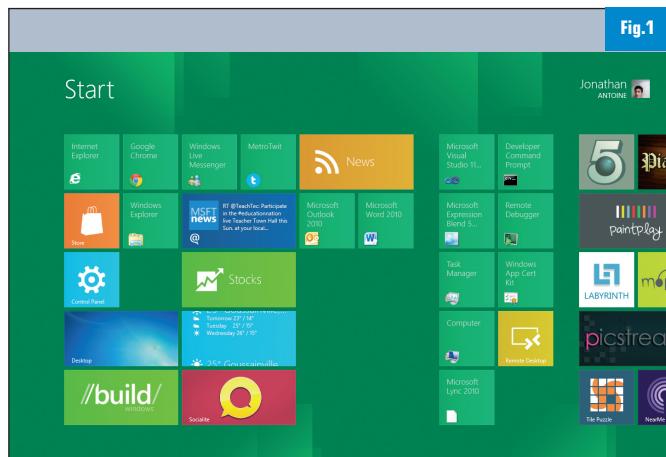
Du 13 au 16 Septembre dernier se tenait à Anaheim, aux États-Unis, la conférence BUILD organisée par Microsoft. Ce fut l'occasion pour l'éditeur de dévoiler une première version de son prochain système d'exploitation, Windows 8. Celui-ci se démarque notamment de ses prédecesseurs par l'apparition d'un nouveau type d'applications dénommées « Metro ». L'objectif de cet article est de présenter cette catégorie d'application ainsi que ses spécificités à travers la création d'une première application.

Metro désigne surtout et avant tout un nouveau type d'interface graphique. Celle-ci est déjà utilisée dans certains produits Microsoft tels que Zune ou, plus récemment encore, Windows Phone. Ce style graphique/ergonomique repose sur la mise en avant du contenu, en particulier textual, dans les interfaces utilisateur plutôt que d'utiliser des représentations graphiques. Ceci sonne par exemple le glas des barres de menus et du système de fenêtres. Toutes les applications s'exécutent en plein écran et ne peuvent pas être redimensionnées, elles ne possèdent pas non plus le bouton standard permettant de les fermer en haut à droite.

Une autre fonctionnalité récurrente des applications Metro est l'utilisation des Live Tile, sortes de vignettes colorées représentant une action possible pour l'utilisateur, enrichies d'informations fournies en temps réel par l'application elle-même. La Live Tile de l'application météo que vous pouvez voir dans la capture d'écran du menu démarrer de Windows 8 affiche par exemple les prévisions pour les trois prochains jours. Voici par exemple le menu démarrer introduit dans Windows 8 : [Fig.1].

Outre les avantages en termes de fluidité et de concentration sur l'information principale, la cible visée par ce type d'application est avant tout les périphériques tactiles. En effet, ce type d'interface facilite une utilisation dite « naturelle » de son ordinateur de par la taille des composants facilement utilisable au doigt. Cette sensation d'évidence à l'utilisation est renforcée à l'aide d'animations qui permettent de rendre l'interface « vivante ».

Les applications classiques (WPF, Winforms, Direct X) ne sont néanmoins pas absentes de Windows 8, Metro apporte une nouvelle façon de mettre en forme leurs interfaces graphiques.



Les applications Metro doivent respecter des exigences très strictes en termes d'ergonomie et de design. De ce fait, elles répondent mieux à certains besoins :

- Les applications sociales : Twitter, Facebook, Google+, etc.
- Les applications media : bibliothèque photos, vidéos, etc.
- Les applications ludiques : jeux simples (cartes, etc.) mais aussi immersifs (3D, etc.)
- Les applications de présentation de données (bourse, météo, etc.)

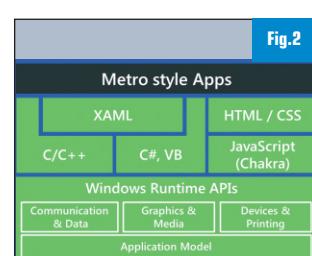
Ce sont donc les applications de consommation de données qui sont principalement visées par la plateforme Metro, mais rien n'empêche bien sûr de créer des applications d'un autre type si cela s'y prête.

L'outillage

Un des points forts de Microsoft sur cette plateforme tient au fait que les outils utilisés pour développer une application Metro sont les mêmes que ceux utilisés depuis plusieurs années par les développeurs. Ils sont en effet constitués du couple Visual Studio pour les développeurs ainsi qu'Expression Blend pour les designers (et les développeurs les plus courageux !). Attention cependant, seuls les outils mis à disposition en préversion lors de la BUILD sont utilisables pour créer des applications Metro : Visual Studio 11 et Expression Blend 5

Metro repose sur le langage XAML pour son interface graphique et peut être complété par du code C ou VB. Un développeur WPF ou Silverlight n'a donc aucun mal à développer une application Metro car le code est très semblable à ce qu'il connaît.

La grande surprise vient cependant du fait que Microsoft ne s'est pas arrêté là. Il est maintenant possible de créer une application Metro en utilisant HTML5 côté interface et Javascript pour le code applicatif. L'outil Expression Blend a d'ailleurs été adapté pour pouvoir créer une application Metro en HTML 5 et l'éditer comme nous pouvions en avoir l'habitude avec une interface écrite en XAML. Il est aussi possible d'utiliser le langage C++ avec une interface graphique décrite en XAML [Fig.2].



Fonctionnement

Depuis l'introduction de Windows NT en 1993, la communication d'une application avec le noyau du système d'exploitation est réalisée à l'aide d'une API nommée Win32. Celle-ci exposée en C, vient

d'arriver à son terme et sa remplaçante, en C++ est nommée « Windows Runtime » plus communément appelée WinRT. Cette librairie est pensée pour être facile d'utilisation, rapide et focalisée dans sa conception sur les langages objets.

Toutes les applications Metro utilisent cette API et bénéficient ainsi de ces améliorations. Alors qu'auparavant les appels aux méthodes de Win32 se faisaient à l'aide d'une syntaxe difficile à utiliser, tous les appels à WinRT se font de manière transparente depuis une application Metro. Cela est possible grâce à un mécanisme appelé « Projection » et à l'aide des « Metadatas Windows ». Ces dernières permettent de pouvoir utiliser l'IntelliSense de Visual Studio sur des objets WinRT. Les applications écrites en JavaScript utilisent quant à elles un fichier WinJS permettant d'accéder aux API WinRT.

La majorité des classes et des méthodes du framework .NET et Win32 se retrouvent dans WinRT mais pas toutes. De plus, afin de privilégier la performance des applications, toutes les API pouvant prendre plus de 50ms ont été rendues asynchrones.

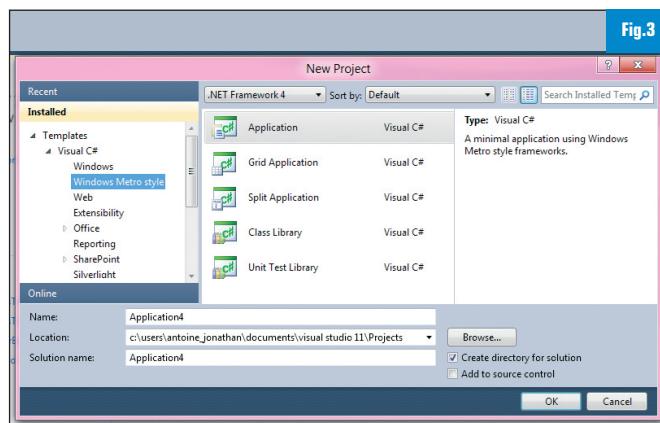
Une autre grosse différence : les applications fonctionnent de manière cloisonnée. Comme les applications mobiles, elles n'ont pas accès au système de fichiers complets et sont soumises à certaines restrictions que nous ne détaillerons pas dans cet article.

Le développement

A présent, nous allons créer notre première application Metro. Celle-ci va être un lecteur de flux RSS, réalisé en C#/XAML. Les différents billets de blogs seront réunis par catégorie et il sera possible de lire les articles directement depuis l'application. De plus, il sera possible de partager les éléments visionnés aux autres applications. Nous rentrerons plus en détail sur cette partie par la suite.

Créer un premier projet Metro avec Visual Studio 2011

La première étape consiste à créer une nouvelle solution avec Visual Studio 11. Sélectionnez le modèle de projet « Visual C# > Windows Metro Style ». Plusieurs choix s'offrent alors à vous : une application « grille » permet de créer une application Metro affichant des données à trois niveaux de façon tabulaire, une application « split » affiche les données sur deux niveaux. Dans notre cas nous allons créer une application sans but prédéfini. Sélectionnez alors « Application », donnez le nom « MaPremiereApplicationMetro » à votre projet et cliquez sur « Ok » [Fig.3]. Votre première application Metro est maintenant créée ! Elle est pour le



moment constituée de trois fichiers principaux :

- App.xaml : Ce fichier représente votre application, vous pouvez rajouter des gestionnaires d'événements correspondant à votre application (lancement, fermeture, etc.)
- MainPage.xaml : Ce fichier correspond à la page principale de votre application. Comme indiqué précédemment, il n'y a pas de fenêtres dans une application Metro, mais des pages qui sont le « contenu » de l'unique fenêtre.
- Package.appxmanifest : ce fichier correspond aux informations de déploiement de votre application. Il sera utilisé par le Windows Store et le système pour savoir ce dont a besoin votre application et comment l'installer sur votre système.

Définir les capacités de l'application

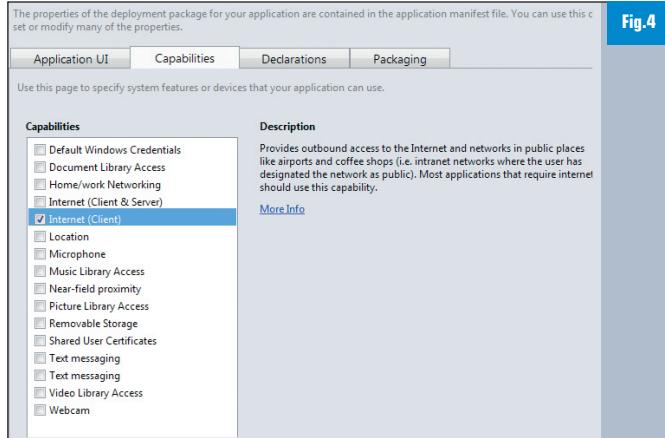
Les applications Metro doivent définir, dans le fichier nommé « *Package.appxmanifest* », quelles sont les différentes API « sensibles » (plus communément appelées « **Capabilities** ») qu'elles vont utiliser. Ce fonctionnement est en fait très semblable à celui de Windows Phone [Fig.4].

L'application d'exemple va nécessiter un accès à internet. Afin de déclarer cette capacité, double-cliquez sur le fichier *Package.appxmanifest*. L'éditeur graphique s'ouvre alors et il faut vous rendre sur l'onglet « Capabilities ». Cochez la case « Internet (Client) » et sauvegardez ce fichier.

Intégration à Windows 8

Un autre onglet de l'éditeur graphique du fichier *Package.appxmanifest* est nommé « *Declarations* ». Il permet en effet de déclarer comment l'application va s'intégrer à Windows 8. Cela est possible grâce à des contrats fonctionnels que l'application va implémenter. Cela peut être considéré comme une interface au niveau applicatif. Ces contrats peuvent, par exemple, indiquer que l'application est associée à une extension de fichier, qu'elle peut être appellée lors de la découverte d'un nouveau périphérique disque, qu'elle peut être exécutée en tâche de fond ou encore qu'elle peut partager des informations.

Aussi, une application peut déclarer être une source de partage d'informations en indiquant qu'elle implémente le Contract « *Share Target* ». C'est justement le cas de notre application : cliquez sur l'onglet « *Declarations* », sélectionnez « *Share Target* » dans la liste déroulante et cliquez sur « *Add* ». Une fois ceci effectué, vous pouvez spécifier les formats de données pouvant être partagés (ceux-ci peuvent être trouvés dans la classe *StandardDataFor-*



mats) dans la section « Data formats ». Notre application va se contenter de partager des URL et il faut donc rajouter le format de données « Uri » [Fig.5].

Définition des objets métiers

Il est maintenant temps de définir nos objets métiers utilisés par l'application. Pas de surprise cette fois-ci, ce sont des objets C# traditionnels. Afin de pouvoir les utiliser comme source des Bindings dans l'interface graphique, nous allons les faire implémenter l'interface `INotifyPropertyChanged`. Attention cependant de bien utiliser celle définie dans le namespace 'Windows.UI.Xaml.Data' et non pas 'System.ComponentModel'. Un site est constitué de plusieurs catégories contenant elles-mêmes plusieurs billets de blogs [Fig.6].

Utilisation de l'asynchronisme

Vous pouvez remarquer dans le diagramme de classe que la classe Site porte une méthode appelée « `ObtenirLesBilletsAsync` ». Cette méthode se charge d'obtenir les billets de blogs d'un site et de créer les catégories correspondantes. L'obtention des éléments d'un flux RSS est très simple avec WinRT (comme cela l'était avec .NET) car la classe `SyndicationClient` se charge d'effectuer tout le travail. Voici le code correspondant à cette méthode :

```
public async Task ObtenirLesBilletsAsync()
{
    Categories.Clear();
    var client = new SyndicationClient() { BypassCacheOnRetrieve
    = true };

    try
    {
        SyndicationFeed feed = await client.RetrieveFeedAsync(RssURI);

        foreach (SyndicationItem item in feed.Items)
        {
            BilletDeBlog billet = new BilletDeBlog();
            string categoryName = RemplirLesInformationsDuBillet
            (item, billet);

            //Créé la catégorie si nécessaire.
            AjouterLeBilletALaCategorie(billet, categoryName);
        }
    }
}
```

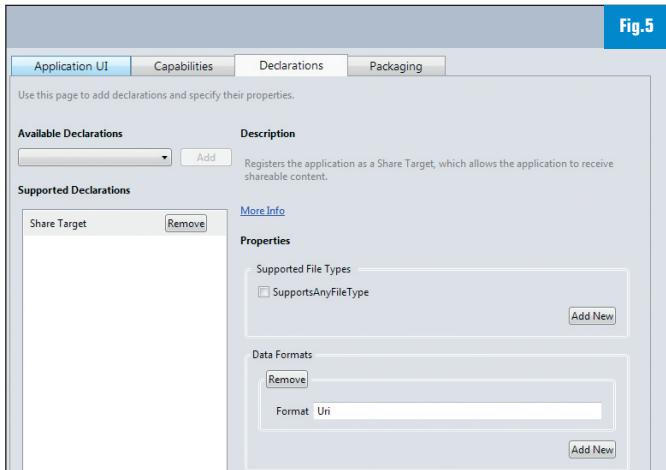


Fig.5

```

    }

    }

    catch (Exception ex)
    {
        Error = ex.Message;
    }
}
```

Ce qu'on remarque dans cet extrait de code, c'est l'utilisation du mot clef `await` combiné au type de retour `Task`. Comme indiqué précédemment, WinRT priviliege l'accès aux méthodes asynchrones et ceci est donc le cas pour la méthode `RetrieveFeedAsync` de la classe `SyndicationClient` car la récupération des éléments d'un flux RSS est une opération potentiellement longue. Le reste du code est classique et consiste à renseigner les propriétés de la classe `BilletDeBlog` avec les informations présentes sur les objets de types `SyndicationItem`.

Cette méthode est présente sur la classe Site. Pour des raisons de simplicité, nous ne présentons pas une architecture optimale mais seulement les différentes techniques à utiliser. Aussi, nous allons mettre en place un singleton représentant le site visé par l'application au sein de la classe App correspondant à celle-ci. Il sera utilisé dans toutes les vues comme source de données et nous allons donc l'initialiser au démarrage de l'application. Au lancement, les applications Metro affichent un SplashScreen jusqu'à ce que la fenêtre courante soit activée. Nous allons donc profiter de cela pour effectuer le chargement des billets et n'afficher la fenêtre principale qu'une fois ce dernier terminé. Voici le code correspondant :

```
public static Site Site { get; private set; }

public App()
{
    InitializeComponent();
    Site = new Site("http://feeds.feedburner.com/JonathanANTOINE");
}

protected override void OnLaunched(LaunchActivatedEventArgs args)
{
    Task task = Site.ObtenirLesBilletsAsync();
```

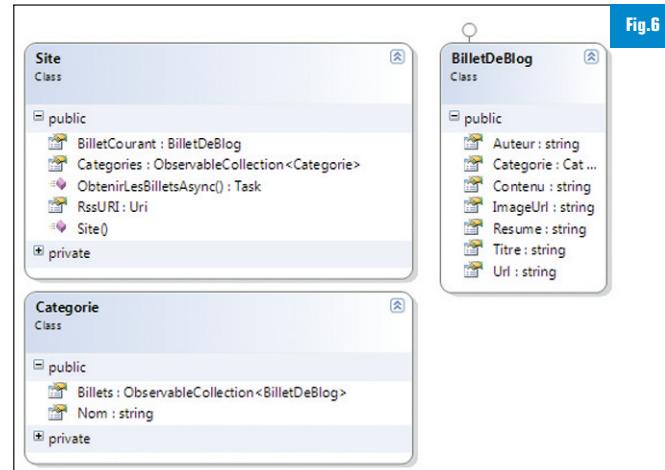


Fig.6

```
task.ContinueWith(_ => AfficherLaPagePrincipale(), TaskScheduler.FromCurrentSynchronizationContext());
}
```

Comme en WPF et en Silverlight, les modifications d'un contrôle graphique doivent être réalisées sur le Thread principal et c'est ce que l'on fait en fournissant le contexte courant d'exécution à la méthode ContinueWith.

Mise en place de l'interface graphique

Il est maintenant temps de créer des vues afin d'afficher ces informations.

La page principale

Cette page est créée par défaut grâce à l'assistant de Visual Studio et se nomme « *MainPage.xaml* ». Après l'avoir ouverte en édition, on constate que l'unique élément défini est son fond noir.

En premier, il faut afficher le titre de l'application. Afin de personnaliser cet affichage, un style est créé et placé dans le fichier *App.xaml*. Vous pouvez remarquer que le système de ressources d'une application Metro est le même que celui dont vous avez l'habitude en WPF ou en Silverlight. Tous les objets placés dans le fichier de ressources *App.xaml* sont accessibles via le système de résolution des ressources dans toute l'application. Ici nous utilisons l'extension de balisage « *StaticResource* », les ressources dynamiques n'étant pas présentes à l'heure actuelle.

```
<!-- le style du titre de l'application dans App.xaml-->
<Style x:Key="PageTitleStyle" TargetType="TextBlock">
    BasedOn="{StaticResource DefaultFontSize}">
    <Setter Property="Margin" Value="0,0,40,28" />
    <Setter Property="VerticalAlignment" Value="Bottom" />
    <Setter Property="TextWrapping" Value="NoWrap" />
    <Setter Property="FontSize" Value="53.333" />
</Style>
<!-- le titre de l'application-->
<TextBlock Text="Ma première application Metro" Style="{StaticResource PageTitleStyle}" />
```

Ensuite, nous affichons les catégories du site à l'aide d'un *ItemControls*. Nous définissons comme source d'éléments la propriété *Categories* du site. Cette écriture serait exactement la même dans une application WPF ou Silverlight. La liste de catégorie est placée dans un contrôle *ScrollViewer* car on ne peut pas prédire à l'avance le nombre de catégories du site et donc la taille prise en largeur par la liste.

```
<Style x:Key="ItemsControlFullStyle" TargetType="ItemsControl">
    <Setter Property="ItemsPanel">
        <Setter.Value>
            <ItemsPanelTemplate>
                <VirtualizingStackPanel Orientation="Horizontal" />
            </ItemsPanelTemplate>
        </Setter.Value>
    </Setter>
    <Setter Property="ItemTemplate" Value="{StaticResource CategorieDataTemplateFull}" />
</Style>
<ScrollViewer Grid.Row="1" VerticalScrollBarVisibility="Disabled">
```

```
    HorizontalScrollBarVisibility="Auto">
        <ItemsControl x:Name="_categorieListe" ItemsSource="{Binding Categories}">
            Style="{StaticResource ItemsControlFullStyle}" />
</ScrollViewer>
```

Comme vous pouvez le constater, la représentation d'une catégorie est indiquée par le *DataTemplate CategorieDataTemplate*. Il n'est pas possible actuellement d'utiliser le système de *DataTemplate* implicite et celui-ci est défini en ressources de la page afin d'améliorer la lisibilité du code. Le voici :

```
<DataTemplate x:Key="CategorieDataTemplate">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>
        <TextBlock Text="{Binding Nom}" VerticalAlignment="Center" TextAlignment="Center" Style="{StaticResource LargeContentFontStyle}" />
        <ScrollViewer HorizontalScrollBarVisibility="Disabled" VerticalScrollBarVisibility="Auto" Grid.Row="1">
            <ListBox ItemsSource="{Binding Billets}" Background="Transparent" SelectionChanged="CategorieItemList_SelectionChanged" ItemTemplate="{StaticResource BilletDeBlogTemplate}" ItemContainerStyle="{StaticResource ListBoxItemStyle}" />
        </ScrollViewer>
    </Grid>
</DataTemplate>
```

Une catégorie est représentée par son nom et la liste des billets qui la constitue est affichée en dessous à l'aide cette fois-ci d'une *ListBox*. On peut remarquer que, comme en WPF ou Silverlight, des gestionnaires d'événements peuvent être placés dans le XAML. Après avoir reproduit ce code XAML, cliquez-droit sur le gestionnaire d'événement « *CategorieItemList_SelectionChanged* » et cliquez sur « *Navigate to event handler* ». Cette dernière manipulation a pour effet de créer celui-ci s'il n'existe pas.

```
private void CategorieItemList_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    if (e.AddedItems.Count <= 0) return;
    App.AfficherBillet(e.AddedItems[0] as BilletDeBlog);
}
```

Aucune surprise non plus ici, la signature du gestionnaire d'événement est classique ainsi que l'argument passé. Dans son traitement, nous récupérons le billet sélectionné de l'argument et nous le définissons comme billet courant du Site. Nous verrons plus tard comment est implémentée la méthode *AfficherBillet*.

Le détail d'un billet

Cette page n'existe pas et il faut donc la rajouter. Pour cela cliquez-droit sur le projet, puis sur « *Add* » et finalement « *New Item* ». Plusieurs templates sont alors disponibles :

- Collection page : permet d'afficher une collection d'éléments ;
- Collection summary page : affiche un groupe d'éléments ainsi qu'un résumé de ses éléments ;
- Grouped collection page : affiche une collection d'éléments organisés par groupe ;
- Item detail page : affiche les détails d'un élément d'une collection et permet la navigation dans celle-ci ;
- Split page : affiche une liste d'éléments et les détails de celui sélectionné ;
- Blank page : une page blanche sans contenu prédéfini.

Sélectionnez cette dernière option et cliquez ensuite sur « Add » après l'avoir renommée « *BilletPage.xaml* ». Nous allons définir son contenu qui va rester très sobre et simple : un bandeau supérieur contenant le titre de l'article avec un bouton permettant le retour vers la liste des catégories et l'article lui-même en dessous. Voici un extrait du code XAML utilisé :

```
<Grid Grid.Row=>0</Grid>
<Grid.ColumnDefinitions>
  <ColumnDefinition Width=>Auto</ColumnDefinition>
  <ColumnDefinition Width=>*</ColumnDefinition>
</Grid.ColumnDefinitions>
<Button x:Name=>BackButton</Button> Click=>BackButton_Click Style=>
{StaticResource BackButtonStyle}</Button>
<TextBlock x:Name=>PageTitle</TextBlock> Text=>{Binding Titre}</TextBlock>
{StaticResource PageTitleStyle}</TextBlock> Grid.Column=>1</Grid>
</Grid>
<WebView x:Name=>webView</WebView> Source=>{Binding Url}</WebView> Grid.Row=>1</Grid>
```

Comme vous pouvez vous en douter, le composant *WebView* permet d'afficher une page web directement dans une application Metro. Il suffit de lui indiquer l'url de la page à afficher (locale ou distante) via sa propriété *Source*.

À ce stade, celui-ci souffre des mêmes soucis que son pendant dans WPF : impossible de jouer sur son opacité, de lui appliquer une matrice de transformation, etc. Cependant, il reste amplement suffisant pour nos besoins.

Utilisation du tactile et de la souris

L'utilisation de l'événement *Click* dans l'extrait de code précédent permet de soulever une question importante : comment sont gérés les périphériques d'entrée dans une application Metro ? Que cela soit une souris, un périphérique tactile, ou encore un stylet, tous sont considérés comme un objet abstrait : un pointeur qui possède toutes les informations intéressantes (position, les touches spéciales type Ctrl, Alt, Esc, etc.) ainsi que les points intermédiaires dans le cas où le périphérique d'entrée ne fournit pas les informations en temps réel. La propriété *PointerDeviceType* de l'argument de l'événement est de même en mesure d'indiquer la nature du périphérique d'entrée. Voici un exemple de gestionnaire d'événement de ce type :

```
private void UIElement_PointerPressed(object sender,
  Windows.UI.Xaml.Input.PointerEventArgs e)
{
  //traitement ici
  var periph = e.Pointer.PointerDeviceType;
}
```

Les applications Metro sont en mesure d'utiliser des périphériques multipoints et de faire un prétraitement fournissant des informations au travers d'un mécanisme de Manipulation similaire à celui de WPF.

Gestion de la navigation

La navigation au sein de l'application est quant à elle réalisée de manière très simple au travers de méthodes statiques de la classe *App*. La procédure est toujours la même :

- Créer le futur contenu de la fenêtre principale,
- Obtenir la fenêtre principale et lui assigner le contenu,
- Optionnellement, activer la fenêtre principale pour lui donner le focus.

Voici par exemple la méthode affichant les détails d'un billet de blog :

```
public static void AfficherBillet(BilletDeBlog billet)
{
  if (billet == null) return;
  Site.BilletCourant = billet;
  Window.Current.Content = new BilletPage();
  Window.Current.Activate();
}
```

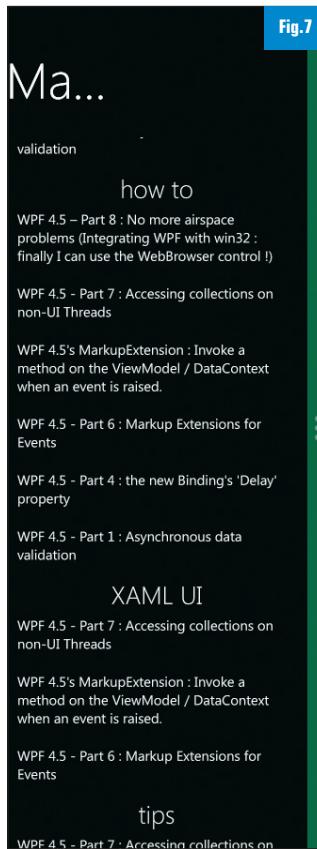


Fig.7

Les différents états d'une vue Metro

[Fig.7]. Une application Metro, en plus d'être soumise à l'orientation du périphérique, peut être affichée suivant plusieurs dispositions :

Full screen : c'est le plus courant, la fenêtre occupe tout l'écran,

Snapped : la fenêtre occupe un cinquième de l'écran,

Filled : la fenêtre occupe quatre-cinquième de l'écran.

Il est important d'afficher de façon différente l'application en fonction de ces modes. À tout moment, il est possible de connaître l'état de l'application à l'aide des classes statiques *ApplicationLayout* et *DisplayProperties* indiquant la disposition courante et l'orientation du périphérique (portrait, paysage, portrait inversé ou paysage inversé).

Le modèle par défaut de Visual Studio pour une page prend en compte ces différents modes et utilise le *VisualStateManager* pour modifier l'affichage. Plusieurs états visuels sont alors définis dans la page et le développeur est libre d'adapter l'affichage en fonction des différents cas.

Nous avions précédemment utilisé un style pour définir les différentes propriétés de l'*ItemControls* affichant les catégories de billets. Cela se retrouve être un choix judicieux car nous pouvons définir un style différent à utiliser lorsque l'application passe en mode « *Snapped* » afin d'afficher correctement les informations.

Nous mettons de même à jour les propriétés du ScrollViewer pour changer le sens de défilement d'horizontal pour vertical.

```
<VisualState x:Name=>Snapped</VisualState>
<Storyboard>
<ObjectAnimationUsingKeyFrames Storyboard.TargetProperty=>Style>
    Storyboard.TargetName=>_categorieListe</Storyboard>
    <DiscreteObjectKeyFrame KeyTime=>0</KeyTime> Value=>{StaticResource ItemsControlSnappedStyle}</DiscreteObjectKeyFrame>
</ObjectAnimationUsingKeyFrames>
<ObjectAnimationUsingKeyFrames Storyboard.TargetProperty=>VerticalScrollBarVisibility</ObjectAnimationUsingKeyFrames>
    Storyboard.TargetName=>_scrollViewer</Storyboard>
    <DiscreteObjectKeyFrame KeyTime=>0</KeyTime> Value=>Auto</Value>
</ObjectAnimationUsingKeyFrames>
<ObjectAnimationUsingKeyFrames Storyboard.TargetProperty=>HorizontalScrollBarVisibility</ObjectAnimationUsingKeyFrames>
    Storyboard.TargetName=>_scrollViewer</Storyboard>
    <DiscreteObjectKeyFrame KeyTime=>0</KeyTime> Value=>Disabled</Value>
</ObjectAnimationUsingKeyFrames>
</Storyboard>
</VisualState>
```

L'implémentation du partage d'information

Précédemment, nous avons déclaré l'application comme capable de partager des adresses internet. Il est à la charge de chaque page de fournir les informations qu'elle souhaite partager. Lorsque la barre de partage est affichée (aussi appelée « charms »), un événement est levé sur l'objet *DataTransferManager* correspondant à la vue courante. Voici le code permettant de s'y abonner dans le constructeur de la vue affichant un billet de blog :

```
public BilletPage()
{
    InitializeComponent();
    DataContext = App.Site.BilletCourant;

    //Abonnement aux demandes de partage
    DataTransferManager
        .GetForCurrentView().DataRequested += BilletPage_DataRequested;
}
```

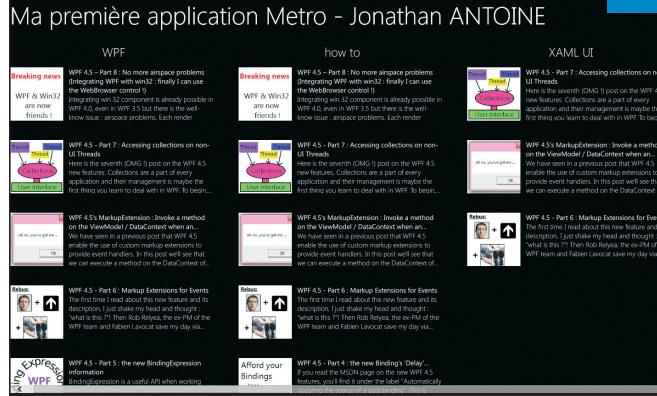


Fig.8

Le gestionnaire d'événement quant à lui se contente de fournir les informations nécessaires au travers de l'argument de type *DataRequestedEventArgs* qui lui est fourni. À noter qu'il est nécessaire de fournir à minima l'URL du blog sous peine de voir la barre de partage afficher une erreur.

```
void BilletPage_DataRequested(DataTransferManager sender,
    DataRequestedEventArgs args)
{
    //assigne un titre
    args.Request.Data.Properties.Title = App.Site.BilletCourant.Titre;
    //assigne une description
    args.Request.Data.Properties.Description = App.Site.BilletCourant.Resume;
    //Assigne une vignette
    args.Request.Data.Properties.Thumbnail =
        StreamReference.CreateFromUri(new Uri(App.Site.BilletCourant.ImageUrl));
    //Assigne la donnée réellement partagée l'URL du billet
    args.Request.Data.SetUri(new Uri(App.Site.BilletCourant.Url));
}
```

Une fois ce code mis en place, la barre de partage s'affiche correctement et affiche les applications capables de recevoir le type d'information fournie.

Il n'est cependant pas garanti que l'application sélectionnée par l'utilisateur utilise toutes les informations fournies par l'application source.

En résumé

Nous avons vu que les applications Metro sont d'un type nouveau de par l'ergonomie mise en place mais que le code reste très semblable à celui d'une application WPF ou Silverlight, excepté quelques spécificités.

Contrairement à ce que l'on pourrait imaginer en découvrant cette technologie, la plus grande difficulté va être de réinventer la façon dont sont conçues les applications d'un point de vue ergonomique dans un environnement mono-fenêtré où le design d'une application est primordial [Fig.8 et 9].

■ Jonathan Antoine – *Infinite Square*
jantoine@infinitesquare.com



Fig.9

Les nouveautés de Team Foundation Server 11

La version 2010 de Team Foundation Server a permis d'imposer un certain nombre de fonctionnalités innovantes dans la gamme de produits Microsoft. En effet, l'apparition de Microsoft Test Manager, outil de tests fonctionnels permet enfin de réconcilier testeurs et développeurs à travers un mode de collaboration outillé et fluidifié ! De son côté, la première version du Lab management apporte une nette amélioration de l'expérience de création d'environnements de tests avec un vrai module industrialisable. Ceci et tout un ensemble de nouvelles micro-fonctionnalités ont fait de TFS 2010 une version majeure du produit.

Il est beaucoup trop tôt pour connaître le nom final de la prochaine version du serveur de Visual Studio (nom de code : TFS 11), mais la mise à disposition d'une première version couplée aux annonces de l'événement BUILD permet d'obtenir une idée claire de la direction prise par l'outil.

Les maîtres mots sont : une interface web revue, un travail énorme au niveau de l'ergonomie de l'explorateur d'équipes, de nombreuses évolutions pour le contrôle de code source et Test Manager, l'apparition d'une notion de Feedback, une version hébergée dans Azure... bref ! La version 11 ne sera pas pauvre en nouveautés.

Cet article offre un aperçu des quelques nouveautés déjà dévoilées.

Contrôle de code source

Le contrôle de code source est une des fonctionnalités qui a, en apparence, subi le moins d'évolutions dans la version 2010, exception faite des interfaces graphiques de gestion des branches. Beaucoup d'améliorations de performances ont été réalisées depuis 2005, mais peu de fonctionnalités ajoutées.

La version 11 apporte, entre autres, une nouvelle approche pour gérer les espaces de travail. Pour rappel, il s'agit de la fonctionnalité qui permet notamment de définir les correspondances entre des réper-

toires du serveur et des répertoires en local sur le poste du développeur.

Par le passé, un espace de travail était géré par le serveur. Lorsqu'un développeur souhaite modifier un fichier, il est forcé de l'indiquer au serveur.

Bien que cela soit fait automatiquement par Visual Studio dans certains cas, il s'agit d'une contrainte que l'on ne peut pas contourner (ou difficilement, avec un mode déconnecté amélioré en 2010). L'avantage de ce mode est que l'on limite les échanges avec le serveur et que l'on gagne de ce fait en performances globales (très avantageux dans des contextes de site de production distant).

Avec la version 11 apparaît la notion d'espaces de travail locaux. Ce mode est choisi par défaut à la création de l'espace et offre la possibilité de ne pas avoir à indiquer à TFS que l'on souhaite modifier un fichier. On le modifie et TFS sera averti automatiquement au moment voulu. Cela simplifie un nombre important de scénarios basés sur des outils externes à Visual Studio pour de la modification de sources.

Le mode déconnecté a également été entièrement revu. Plus besoin de relancer

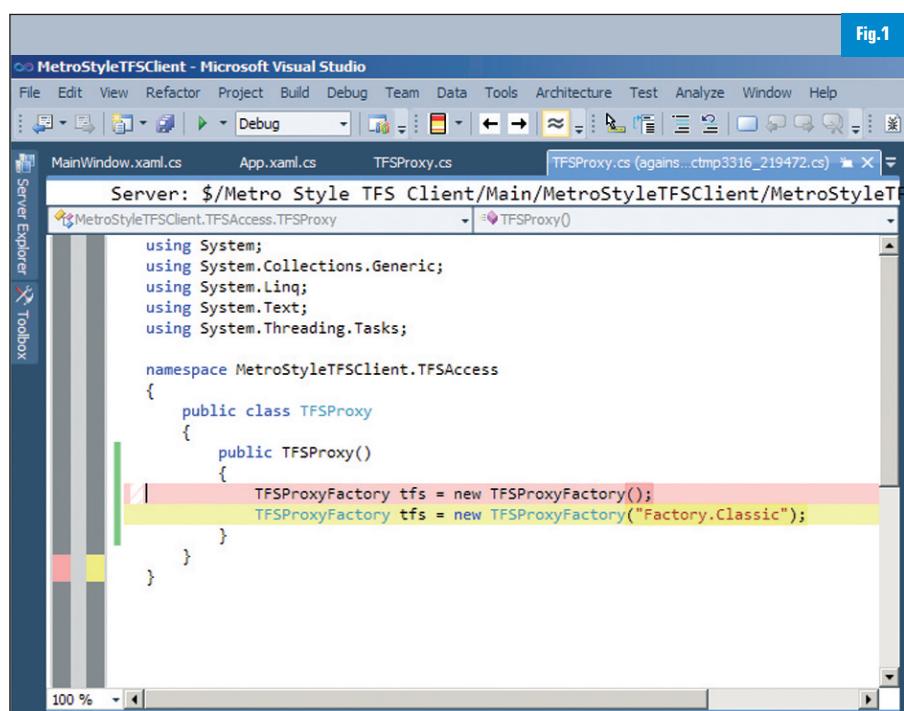
Visual Studio et modifier ses fichiers sources quand le serveur TFS est inaccessible ! Plus besoin de surcharger l'attribut de lecture seule... car il n'y en a plus ! On peut désormais effectuer une extraction, modifier les fichiers, effectuer des ajouts au contrôle de source, annuler nos modifications, comparer avec la version d'origine sans accéder au serveur.

Enfin, l'interface de comparaison de fichiers a aussi été redéveloppée (« enfin ! »). Elle permet maintenant de visualiser selon deux modes, dont un nouveau qui compare chaque ligne en les positionnant l'une en dessous de l'autre (un peu tel que le fait WinDiff) [Fig.1].

Team Explorer : une nouvelle expérience utilisateur

Fonctionnellement et visuellement, le Team Explorer (ou explorateur d'équipes) n'a également pas évolué depuis la première version. En 2010, c'est toujours la même arborescence qui permet d'accéder aux différentes fonctionnalités et d'interagir avec le serveur.

Avec TFS 11, il paraît relativement évident



que l'équipe produit mise beaucoup plus sur l'ergonomie de cet outil. Le Team Explorer n'est plus un point d'entrée vers un ensemble de fonctionnalités, mais un ensemble de pages au travers desquelles les utilisateurs naviguent pour effectuer leurs opérations quotidiennes. Plus que jamais, il devient l'outil qui accompagne les utilisateurs.

Le développeur est guidé à travers un flux de travail. Lorsqu'il débute une tâche, il est invité à la préciser. S'il souhaite mettre en pause son travail, l'interface lui permet de tout faire pour lui : changer l'état de l'élément de travail, réserver le code (le mettre sur étagère), annuler les modifications en local, etc. Lorsqu'il souhaite reprendre son travail, c'est la même chose mais dans le sens inverse : l'élément de travail repasse en cours de progression, le code est dé-réserve (et désormais fusionné avec les modifications en local !).

Aujourd'hui, trop peu d'équipes mettent en place des processus de revue de code. La raison est souvent liée à la complexité, au manque d'outillage et surtout à l'impression d'inutilité. TFS 11 outille enfin ce processus important en cycle projet et permet à chaque développeur d'effectuer une demande de relecture au travers de

Team Explorer. Le relecteur reçoit une demande via un élément de travail et, après acceptation, peut commenter le code. Chaque commentaire peut être visionné par le développeur à la source de la demande, qui peut y répondre, apporter les modifications nécessaires, et lorsque tous les deux sont d'accord, réellement archiver le code source [Fig.2].

Une nouvelle interface web

L'interface web de TFS existe depuis les toutes premières versions. A l'origine développée par une société externe, elle a été rachetée par Microsoft et intégrée au produit dans la version 2008. La version 2010 n'y a apporté que de légères modifications, surtout liée à son intégration dans SharePoint, la version 11 la révolutionne. De simple interface d'appoint, elle évolue vers le centre de communication de l'équipe et l'endroit où on organise le projet. Que ce soit la création des itérations ou des zones, la gestion des droits ou des équipes sont toutes réalisées uniquement par l'intermédiaire du site d'équipe dans la version disponible au moment où ces lignes sont écrites.

La notion d'équipe est certainement le changement le plus important. Au-delà d'être un simple groupe de sécurité tel qu'en version 2010, une équipe permet de mettre en place un espace privilégié pour un groupe de personnes qui travaillent ensemble avec un objectif commun. Le site permet de gérer son backlog, de visualiser un tableau permettant de voir les tâches à réaliser sous forme de post-it, de définir les dates de début et de fin des itérations, de définir les dates d'indisponibilité des

membres de l'équipe, et bien d'autres choses... [Fig.3 et 4].

Team Foundation Service

Team Foundation Server 11 n'apporte pas seulement de nouvelles fonctionnalités aux utilisateurs. La sortie de cette version rime également avec l'annonce de la version hébergée dans Azure. Microsoft propose actuellement, sur invitation, de tester le service d'hébergement de Team Foundation Server sur Azure : Team Foundation Service.

Pour faire simple, il s'agit des mêmes fonctionnalités que la version basique de TFS, celle comprenant les fonctionnalités de gestion de code source, d'éléments de travail et de builds mais ne disposant pas de portail SharePoint ou serveur de rapports. Les avantages complémentaires sont ceux liés à l'hébergement dans les nuages : une haute disponibilité garantie, plus besoin de se préoccuper de l'administration technique, la maintenance ou encore la gestion des sauvegardes et surtout, plus besoin d'installer le serveur.

Pour l'instant, aucune information sur le prix et le modèle de licences, mais il est certain que cela ouvre la porte à des scénarios très intéressants et aidera encore plus à accélérer l'adoption de Team Foundation Server.

Fig.2

Fig.3

Fig.4

Les nouveautés de Visual Studio 2011

Afin de permettre aux développeurs d'exploiter efficacement les nouveaux concepts de Windows 8, Microsoft a mis à disposition de ces derniers une première version de Visual Studio 11. Le but de cet article est donc de vous donner une vision globale des nouveautés apportées par cette version.

Visual Studio 11 permet le développement d'applications adaptées au nouveau modèle induit par Windows 8 : Metro Style Application vs Desktop Application. En effet, Windows 8 propose 2 environnements utilisateur, l'environnement « Desktop » relativement similaire à ce que propose aujourd'hui Windows 7 et l'environnement « Metro » destiné aux périphériques tactiles (type tablette) proposant une ergonomie adaptée à ce type d'utilisation.

Metro Style Application

Visual Studio 11 intègre un nouveau designer permettant la création d'applications. Alors que Visual Studio 2010 proposait un designer dédié : « Cider », qui pouvait être utilisé conjointement à Expression Blend,

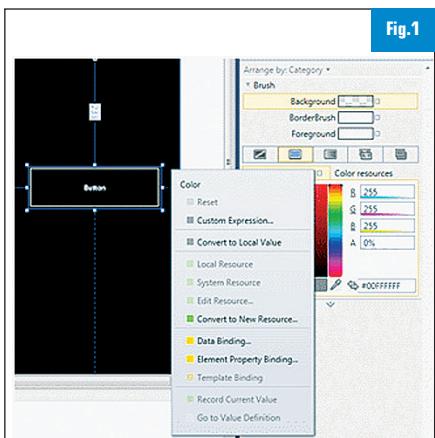
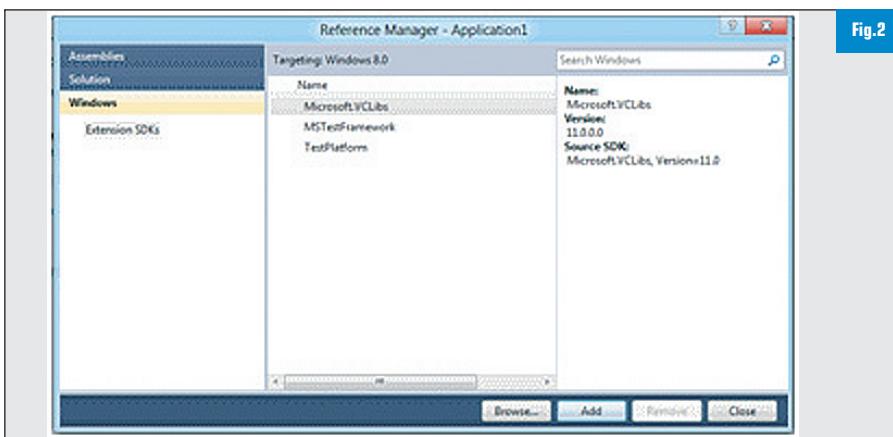


Fig 1

Visual Studio 11 embarque directement le moteur de design de la nouvelle version d'Expression Blend (outil de design permettant la création d'applications bâties sur le langage XAML ou HTML 5). Ceci permet d'homogénéiser l'expérience entre ces 2 produits et donc d'améliorer la productivité des développeurs (gestion du layout, édition des propriétés, XAML intelliSense, design time, etc.) [Fig.1]. L'une des plus importantes nouveautés apportées par Windows 8 est la possibilité de développer des applications « Metro style » en utilisant le couple « HTML 5 / JavaScript ». Alors que les versions précédentes d'Expression Blend étaient destinées exclusivement à l'édition du XAML, la version 5 gère la manipulation du HTML 5, permettant ainsi de conserver cette collaboration efficace entre les 2 produits, même pour les applications utilisant ce nouveau modèle de développement.

Productivité

Toujours dans un souci de productivité, Visual Studio 11 fournit une version améliorée de certains composants inhérents au développement de tout type d'application comme l'explorateur de solution, qui permet maintenant d'effectuer une recherche sur l'ensemble de la solution (classe, méthode, propriété, etc.) et d'y accéder directement depuis l'arborescence. Le gestionnaire de références a également été amélioré.



Ein 2

ment été revu afin de permettre ici aussi d'effectuer une recherche mais également de catégoriser les références disponibles. Nous avons également une nouvelle version de l'outil de recherche, de gestion des onglets, etc. De façon plus générale Visual Studio 2011 intègre nativement une extension très populaire de Visual Studio 2010 : les « Productivity Power Tools » [Fig.2 et 3].

Applications DirectX

L'expérience de développement d'applications DirectX a été également améliorée avec l'arrivée de nouveaux templates destinés aux développements de jeux dans un contexte « Metro Style ». De nouveaux outils de gestion de contenu (textures, etc.) sont intégrés pour faciliter le développement de ce type d'application ainsi que l'ajout du support du HLSL (High Level Shader Langage) au niveau de l'IDE (IntelliSense et Debugging) [Fig.4].

Développement web

Avec l'arrivée de ce nouveau couple « HTML 5 / JavaScript », Visual Studio 11 s'enrichit de toute une batterie de nouvelles fonctionnalités améliorant la productivité du développement web. Parmi celles-ci, on constatera une plus grande facilité à produire du code HTML 5 / CSS 3.0, notamment à l'aide de Code Snippets.

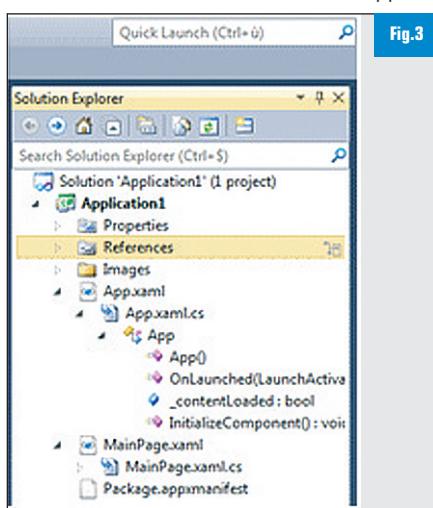


Fig. 2

mais également de nombreuses fonctionnalités liées à la qualité du code (meilleure indentation, mise à jour automatique de la balise fermante lors de l'édition de la balise ouvrante, outlining).

Les avancées concernant le développement en JavaScript ont été particulièrement mises en avant. Ainsi, Visual Studio intégrera dorénavant de nombreux outils déjà familiers des développeurs web sous Internet Explorer tel que l'explorateur de DOM (Document Object Model), qui permettra d'analyser non seulement la structure HTML d'une page mais également l'enchaînement des styles CSS, ou encore la console de JavaScript. L'IntelliSense JavaScript a également été grandement améliorée et l'on retrouvera de nombreuses fonctionnalités manquantes pour les habitués du développement C# comme par exemple la possibilité de naviguer depuis une méthode vers la définition de cette dernière. Au final, le développement JavaScript sous Visual Studio tirera parti du moteur JavaScript déjà construit pour Internet Explorer. Pour finir avec les nouveautés web de cette nouvelle mouture de Visual Studio, il ne faut pas oublier que de nombreux efforts ont été réalisés côté ASP.NET afin d'aider les développeurs à produire des interfaces web adaptées selon le type de navigation (ordinateur ou téléphone). A noter également, pour les applications mobiles ASP.NET, que ces dernières pourront maintenant utiliser la réduite (mais non moins efficace) bibliothèque JQuery mobile [Fig.5 et 6].

Gestion du cycle de vie des applications

Du côté de l'ALM (Application Lifecycle Management), la Build fut l'occasion de montrer plus en détail de nombreuses fonctionnalités déjà annoncées lors des derniers Microsoft TechEd aux USA. C'est

ainsi que l'on aura pu, entre autres, avoir une démonstration de la nouvelle interface du Visual Studio Team Explorer 11 qui se voudra encore plus intuitif en proposant une interface épurée et accessible.

On retrouvera ainsi directement dans ce nouvel affichage les principales informations nécessaires au développeur dans son quotidien : les fichiers ayant subi une modification depuis le dernier checkin, ou encore la récente partie « My Work » qui contiendra directement les Work Item assignés au développeur (ne nécessitant plus l'exécution d'une requête pour les afficher, comme cela pouvait être le cas dans Visual Studio Team Explorer 2010).

Une autre modification importante qui risque de chambouler les habitudes des développeurs sous Visual Studio 2010 est l'apparition d'une interface permettant de demander à un autre développeur de revoir le code produit, ou le contraire, revoir, annoter voire valider le code écrit par un autre développeur [Fig.6].

Si l'outil Code Analysis était déjà disponible depuis Visual Studio 2010, il le sera cette fois-ci dans toutes les versions de Visual Studio (y compris les versions Express) et sera doté d'une nouvelle fonctionnalité : Code Clone Analysis. Cette dernière permet de retrouver, de manière sémantique, certaines portions de codes afin de faciliter le refactoring ou la correction de bugs similaires. Enfin, pour finir, il faut souligner que les tests unitaires sous Visual Studio 11 sont dotés d'une nouvelle interface et ont été rendus plus extensibles : ils peuvent maintenant supporter des adaptateurs tiers de tests unitaires tels que xUnit.net.

Le framework .NET 4.5

Visual Studio, l'environnement de développement utilisé pour la conception d'applications a été mis à jour avec de nouvelles fonctionnalités, mais c'est également le

cas du Framework .NET qui passe maintenant en version 4.5.

Au programme, il n'y a pas encore de grandes révolutions mais diverses briques ont été mises à jour. C'est par exemple le cas de MEF (Managed Extensibility Framework), de WPF (Windows Presentation Foundation), qui intègre maintenant un ruban, qui propose une meilleure intégration entre les composants Win32 et les contrôles WPF. WCF (Windows Communication Foundation) et WF (Windows Workflow Foundation) ont aussi été mis à jour, que ce soit pour ajouter des fonctionnalités ou bien en améliorer d'autres, qui étaient déjà présentes mais pas forcément idéales, pour les développeurs.

La liste complète des nouveautés/évolutions apportées au framework peut être consultée directement à l'adresse suivante : [http://msdn.microsoft.com/en-us/library/ms171868\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/ms171868(v=vs.110).aspx)

Conclusions

Avec cette nouvelle version, Microsoft fournit d'une part tous les outils nécessaires aux développeurs pour une adoption efficace des nouveaux modèles apportés par Windows 8 mais également une nette amélioration de l'expérience et de la productivité de développement.

■ Arnaud **Auroux** – *Infinite Square*

aauroux@infinitesquare.com

■ Vivien **Fabing** – *Infinite Square*

vfabing@infinitesquare.com

Fig.6

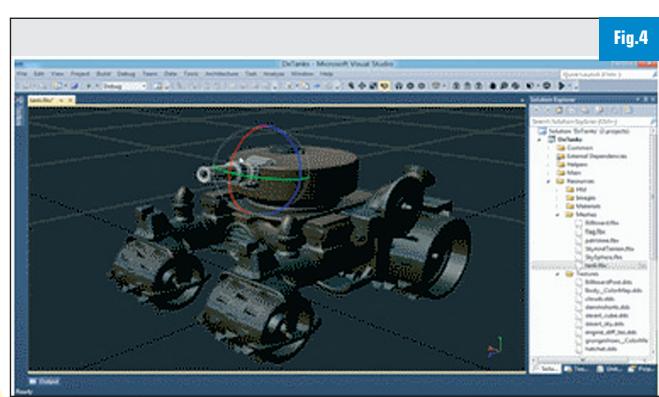
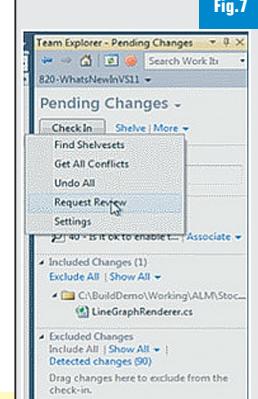
```
.page
{
    width: 960px;
    background-color: #fff;
    margin: 20px auto 0px auto;
    border: 1px solid #496077;
    -moz-border-radius: 15px;
    -webkit-border-radius: 15px;
    border-radius: 15px; /*border radius*/
}
```

Fig.5

```
.page
{
    width: 960px;
    background-color: #fff;
    margin: 20px auto 0px auto;
    border: 1px solid #496077;
}

.header
{
    border-bottom-left-radius: 15px;
    border-bottom-right-radius: 15px;
    border-radius: 15px;
    border-top-left-radius: 15px;
    border-top-right-radius: 15px;
    border-bottom: 1px solid #496077;
    border-top: 1px solid #496077;
    border-left: 1px solid #496077;
    border-right: 1px solid #496077;
}
```

Fig.7



Travailler avec les administrations et collectivités

Les marchés publics – Etat, santé et collectivités territoriales – représentent une opportunité d'affaires à ne pas négliger... à condition de respecter les règles du jeu de l'appel à projet public de la fonction publique.

5 millions de fonctionnaires – 1/2 Etat, 1/3 territoriaux, 1/5 hospitaliers – agissent, consomment, gèrent nos administrations, nos hôpitaux, nos collectivités territoriales. 90 000 acheteurs publics alimentent ainsi une commande publique qui s'est élevée en 2010 à 140 milliards d'euros ! Même si la politique de maîtrise des dépenses publiques dans un environnement économique plutôt défavorable s'impose comme une contrainte forte, les marchés publics représentent une opportunité que les TPE et PME des métiers de l'informatique ne peuvent négliger. La commande publique représente d'ailleurs environ un quart du chiffre d'affaires de ces entreprises et soutient leur activité. Pour autant, le contrat public n'est pas chose simple pour des entreprises plutôt habituées à la relation commerciale directe avec leurs congénères ou le grand public. Il est en effet soumis à des règles et

appels d'offres sont publiés au JOUE (Journal Officiel de l'Union Européenne) ou au BOAMP (Bulletin Officiel des Annonces des Marchés Publics). Nous noterons également la Procédure de Dialogue Compétitif, destinée aux marchés publics complexes, par exemple certains gros projets de développement logiciel, dont les difficultés associées à la définition des moyens techniques ou au montage juridique ou financier du pro-

jet imposent au pouvoir adjudicateur d'établir un dialogue avec les entreprises qui seront ensuite invitées à soumettre une offre.

Le CMP impose le respect de trois grands principes : la liberté

d'accès à la commande publique ; l'égalité de traitement des candidats, qui est de la responsabilité de l'acheteur public gestionnaire des fonds ; et la transparence des procédures (art. 1 du CMP). Toute entreprise est libre de soumissionner, à la condition expresse d'être en règle, c'est-à-dire d'avoir une existence légale et de respecter la fiscalité. Pour cela, lors du retrait de l'appel d'offres, même numérique, elle devra fournir des documents obligatoires.

La commande publique représente 31,8% du chiffre d'affaires "Travaux" 23,5% "Services" et 13,6% "Fournitures" des entreprises.



usages qui peuvent surprendre, jusqu'à paraître lourds. Comme le rappelle **Vincent Goujon**, Directeur des Achats et de l'Innovation du CNRS, travailler avec les administrations « nécessite de maîtriser l'environnement de la fonction publique ». Et de rappeler que les marchés publics sont financés par le contribuable.

Le processus d'achat de la fonction publique est codifié par le CMP, Code des Marchés Publics. Il prévoit qu'à partir de 4 000 € (art. 28 du CMP), l'achat passe par une procédure d'appel d'offres. De 4 000 € à 90 000 € pour les communes et organisations territoriales ou 125 000 € pour l'Etat, la procédure est dite « adaptée » (MAPA ou Marchés à Procédure Adaptée), ce qui se traduit par une obligation de publication. Au-delà de ces montants, qui sont révisés régulièrement par le Gouvernement, la procédure est « formalisée », c'est-à-dire que les

que le démarrage de la procédure en ligne définit l'heure du dépôt, il y a très peu de dossiers qui sont incomplets. Concrètement, cela signifie que la compétition peut être rude, et que si ce marché vous intéresse, vous allez devoir soigner votre rendu ! Il faut d'ailleurs compter une dizaine d'heures pour répondre à une consultation simple.

Le candidat peut retirer le DCE, Dossier de Consultation des Entreprises par voie électronique, et répondre par voie papier ou numérique. Dès le 1er janvier 2012, les marchés publics seront intégralement dématérialisés en émission comme en réception... Toutes les commandes publiques qui font l'objet d'un appel d'offres seront soumises à l'obligation d'être disponibles par voie électronique. Il est cependant peu probable que la majorité des collectivités territoriales et administrations seront prêtes à cette date ! Le cahier des charges est accompagné d'un règlement de la consultation et de son calendrier, il est évidemment indispensable d'en prendre connaissance dans les détails et de qualifier le projet afin d'apporter une réponse globale adaptée au DCE.

« Il est fondamental de maximiser les réponses, et pour cela de connaître le client afin de répondre aux attentes de l'acheteur

Proposer une réponse adaptée

Si sur les marchés généraux il n'est pas rare que plus de 50 entreprises répondent à un appel d'offres, la concurrence est

moindre sur les marchés informatiques. Selon **Jean-Alain Thiebaud**, de KBU Conseil et ex-acheteur et DSI de la Communauté d'Agglomération de Cergy Pontoise, la moyenne serait de 10 compétiteurs, voire de 2 à 6 réponses sur le progiciel. On notera également qu'en dehors du retard à la remise de l'appel d'offres, qui est immédiatement sanctionné – **Jacques Setton**, de WayCast Development, indique



Les bonus de la démat

La dématérialisation des appels d'offres est un avantage... pour les entreprises qui la pratiquent ! La réponse est plus rapide, plusieurs réponses peuvent être déposées, autant de fois que souhaité par l'entreprise, seule la dernière sera retenue ; elle nécessite en revanche de définir en interne une charte des processus de dématérialisation et de réponse en ligne. N'oubliez pas que vous devez disposer d'un certificat de signature électronique. Malheureusement, la majorité des établissements publics ne seront pas prêts au 1er janvier 2012...

Les PME face à l'administration

Les marchés publics ont la réputation de ne pas être accessibles aux TPE et PME... Lorsque l'on est petit, les chances de gagner un appel d'offres seraient inférieures à 10% ! Ainsi, les experts de la CICF conseillent à leurs adhérents de privilégier le statut de SARL à celui d'indépendant. Pourtant, la tendance est à la facilitation de l'accès aux domaines publics par les PME. Ainsi au CNRS, 65% des fournisseurs sont des PME. « *Il n'existe pas de critères qui pourraient favoriser une SARL au détriment d'un petite structure* », a assuré Vincent Goujon.

public, et d'apporter une valeur technique et fonctionnelle », nous confie Catherine Moreau, de CAM International, experte en ce domaine. Il est possible de prendre contact en amont avec l'acheteur, à condition que cette démarche soit justifiée. La question sera soumise par écrit, et la réponse sera communiquée à l'ensemble des candidats. Le questionnement est donc à utiliser avec parcimonie... Catherine Moreau invite également à gérer le closing, la phase finale de la négociation, celle où vous passerez le grand oral (éventuellement, car si l'appel d'offres est 'ouvert', il n'y a pas de négociation). « *L'architecture de la réponse permet de positionner le candidat. Vous devrez partir de la demande du client, présenter les avantages de votre société et démontrer les caractéristiques de votre offre.* »

Retenu ou rejeté ? Grandissez...

Pourquoi une offre est-elle retenue ? Pour Vincent Goujon, c'est d'abord l'adéquation du rapport qualité-prix qui va peser sur la balance.

Au sein d'une organisation comme le CNRS, les appels d'offres se comptent par milliers, et les acheteurs à l'unité.

Dans ces conditions, et pour la majorité des produits, le prix est le critère principal, mais pas forcément essentiel.... Viennent ensuite la connaissance du client, la relation avec lui et la projection vers futur.

Les critères écologiques, qui ne présentent pas d'intérêt particulier sur beaucoup de produits, peuvent parfois peser dans la prise de décision, surtout s'ils font l'objet d'une cotation dans le cahier des charges.

A l'inverse, le manque de connaissance du

besoin du client peut peser lourdement sur la prise de décision.

Jean-Alain Thiebaud invite tout particulièrement à lire le CCTP, Cahier des Clauses Techniques Particulières, qui correspond dans le langage courant au cahier des charges. Cette lecture définit la qualité de votre interlocuteur, et soulève parfois une interrogation : sait-il de quoi il parle ? En effet, l'acheteur occupe une place particulière dans la structure de la collectivité. Dépend-il du service achat ou d'un service marchés ? A-t-il la compétence suffisante pour analyser les aspects techniques, voire sociaux du projet ? Les acheteurs sont humains, ils ont également leurs faiblesses, et vous pouvez s'ils le souhaitent participer à leur compréhension. Rappelons également la présence du CCAP, le Cahier des Clauses Administratives Particulières, qui détaille par des clauses la partie administrative et juridique du marché.

Votre proposition n'a pas été retenue ? Ne vous formalisez pas, la collectivité ciblée reste un prospect ! En revanche, c'est le moment de progresser et d'apprendre. A la réception de la notification du rejet de l'offre, demandez à ce que l'on vous transmette le Rapport d'Analyse des Offres, que l'organisation a l'obligation de vous fournir. Ce document classe les offres à partir des critères définis dans le DCE, dans la limite du secret industriel et commercial. Si vous êtes en bonne place, c'est encourageant ; si au contraire vous figurez en queue de peloton, vous devrez revoir votre copie pour votre prochaine candidature. Vous disposez également d'un délai de 2 mois pour transmettre un recours éventuel auprès du Tribunal, délai suspendu si l'acheteur ne transmet pas le Rapport d'Analyse dans les 15 jours qui suivent le rejet. La prudence s'impose cependant si vous souhaitez engager une telle procédure, certains appels d'offres peuvent parfois respecter une procédure imposée mais le marché est acquis par un prestataire dont l'action s'inscrit dans le temps (par exemple l'éditeur d'une solution spécifique) et les procédures admi-

nistratives sont particulièrement longues pour un résultat aléatoire.

Se faire payer

Le paiement par les administrations a mauvaise presse... Elles sont cependant soumises à la loi, qui leur impose un délai de paiement de 30 jours. De l'avis des experts de la CICF, ce délai est généralement respecté. Il faut rappeler en particulier que la collectivité est composée de trois entités : les élus, qui ne sont pas toujours décisionnaires ; l'administration, pilotée par la DGS (Direction Générale des Services), avec ses Directions opérationnelles ; et le comptable public, dont l'agent comptable est externe aux collectivités (c'est l'Etat qui paie...). Cette structuration peut être à l'origine de légers décalages dans les paiements.

Les administrations n'échappent pas à la vague du numérique et de nombreux chantiers présentent une réelle opportunité pour tous les acteurs des technologies informatiques. A condition d'en connaître et respecter les règles et procédures, TPE et PME peuvent s'y faire une place, et être payées en retour... Quelques conseils : effectuer une lecture exhaustive du cahier des charges ; faciliter le travail des acheteurs ; mettre en avant ses avantages et bénéfices ; reprendre toutes les caractéristiques au niveau des lecteurs ; inscrire la réponse dans la durée ; et inspirer confiance.

Article réalisé en collaboration avec la CICF

La CICF – Chambre de l'Ingénierie et du Conseil en France –, qui fêtera son centenaire en 2012, est composée de 12 syndicats professionnels, dont l'informatique, l'industrie, le management, l'ergonomie, etc. Elle est représentée en région au travers de 16 entités. Elle fédère les ingénieurs et consultants dits « de proximité » et à ce titre est l'équivalent du Syntec pour les métiers de l'informatique, mais à destination des TPE et PME.

Les chantiers modernes de la commande publique

Vincent Goujon, Directeur des Achats et de l'Innovation du CNRS, évoque les quatre tendances qui pilotent l'innovation administrative de la commande publique, ou plus directement des achats de l'administration :

- La réduction des dépenses publiques, principalement des frais généraux ;
- Les objectifs d'achats durables, tant pour l'environnement que pour le social ;
- La modernisation administrative via la dématérialisation fiscale complète ;
- La mesure de la performance.

CES ENTREPRISES QUI RECRUTENT

Ingénieurs et administrateurs Systèmes et Réseaux

CELESTE, fournisseur d'accès Internet Haute Disponibilité pour les entreprises, recherche des profils techniques pour ses services technique opérationnel et de R&D. www.celeste.fr

Informaticiens

Le Groupe ADDITEAM, SSII indépendante, recrute 30 informaticiens : Analystes d'Exploitation, qui intègrent les applications développées aux Etudes dans les univers de Production Informatique ; Techniciens Support de premier et second niveaux ; Techniciens et Administrateurs Réseaux ; Ingénieurs Systèmes ; Gestionnaires de Bases de Données. www.additeam.com

Informaticiens

Le groupe Acteos, éditeur de progiciels d'optimisation des flux logistiques, recrute des profils multiples : Chefs de projet TMS, WMS, et approvisionnement / prévisions ; Chefs de projet R&D / ingénierie logicielle ; Consultants Supply Chain ; Formateurs / Chargés de support client ; Equipes de test / déploiement / qualité ; Architectes techniques SaaS, Architectes réseaux & systèmes. www.acteos.com

Consultants

Acial, société de conseil et de services spécialisée dans le test logiciel et de la qualité du SI, recrute 35 collaborateurs en Ile de France et en Pays de la Loire (Nantes, Niort, Angers) : ingénieur d'Affaires ; Consultant senior assurance qualité ; Consultant qualification logiciel (junior ou confirmé) ; Consultant Outils de Test ; Consultant Assistance MOA ; Consultant Assistance MOA Banque/Finance ; Chef de projet Junior/Senior. www.acial.fr

Infogérance

Computacenter, services et solutions informatiques, recrute 150 collaborateurs supplémentaires afin de répondre au développement de son activité de Services, lié notamment au gain de plusieurs contrats d'infogérance bureautique. **Profils débutants et expérimentés** : Technicien informatique ; Superviseur ; Team Leader ; Coordinateur / Planificateur de ressources ; Gestionnaire de parc informatique ; Technicien d'exploitation ; Technicien informatique itinérant ; Packageur ; Technicien help-desk. **Profils expérimentés** : Service Delivery Manager ; Service Manager ; Ingénieur poste de travail ; Superviseur help desk ; Architecte ; Ingénieur d'exploitation ; Technicien commercial. www.computacenter.fr

Ingénieurs et techniciens

Econocom Managed Services, société de conseil et de services en informatique, recrute 200 nouveaux collaborateurs, sur des profils d'ingénieurs de niveau bac +4/5 et de techniciens de niveau bac +2/3. www.solucom.fr

Ingénieurs et grandes écoles

Ozitem, société de services spécialisée dans le domaine de l'infrastructure systèmes et réseaux, recrute 130 collaborateurs sur Paris et Toulouse d'ici fin 2012, profils débutants issus de grandes écoles de commerce et d'ingénieurs, et profils expérimentés venant de SSII ou d'éditeurs. Postes techniques (infrastructure, production), postes fonctionnels (AMOA, gestion du changement, ITIL), postes de management (team leader, chef de projet, responsable opérationnel dédié), postes commerciaux. www.ozitem.com

Les acheteurs publics face à la complexité grandissante des TIC



Entretien avec Louis-Aimé de Fouquières, Administrateur du Conseil National des Ingénieurs et Scientifiques de France, Associé de SXP Organisations

La fonction achat du secteur public est-elle en difficulté ?

Le secteur public a une problématique dans ses procédures

d'achat, qui sont souvent prises pour une contrainte. Certes il y a les grands principes, comme l'accès au marché et la transparence, mais il y a également des facilités pour réaliser de l'achat professionnel, sans acheter moins bien sous prétexte que l'on est sous contrainte. Les réformes récentes, le nouveau Code des achats publics version 2006, sa version 2005 qui s'applique aux activités contrôlées par les structures administratives, et dernièrement la refonte et l'harmonisation du Cahier des Clauses Administratives Générales, donnent un cadre général à l'achat public et les outils de base sur certains types d'achats, dont un CCAG TIC pour l'achat informatique, avec notamment citée la réversibilité, une notion clé sur l'achat de prestations informatiques.

Elle peine cependant à externaliser...

Le secteur public est contraint par la RGPP d'être toujours plus efficace et de dépenser à bon escient. S'il ne peut se passer d'informatique, s'en munir devient complexe et nécessite une part importante de ressources extérieures. Aujourd'hui, se développe une offre toujours plus professionnelle, qui a commencé par le plus simple, l'infrastructure, et qui monte vers des applications standardisées et externalisées, comme les ressources humaines. La difficulté est d'en dessiner avec précision les services attendus, les niveaux convenables face à une attente raisonnable. La question des spécifications du service est assez différente de celle du cahier de charges fonctionnelles. Les questions portent sur le niveau de service, la pérennité, le partage des bénéfices du progrès technique et l'apprentissage. Et plus spécifiquement pour le secteur public sur la question essentielle de l'hébergement géographique des données. C'est un sujet qui fait débat.

Pourtant, la contractualisation est possible ?

Oui, mais comment passer un contrat de service informatique ou d'externalisation ? Il faut réaliser une sélection progressive, et lorsqu'on a choisi

un prestataire, on commence à écrire le contrat. Ce n'est pas compatible avec les règles du secteur public, qui voudraient que le contrat soit écrit avant la conclusion. Grâce au CCAP et à des structures adaptées, ce n'est plus insurmontable, mais ils obligent le secteur public à une gymnastique pour obtenir des propositions et des projets de contrats. Signer un contrat demande beaucoup d'efforts. Le fournisseur a des devoirs vis-à-vis de son client, mais le client également, pour lui expliquer ce qu'il attend, lui indiquer les changements d'environnement qui vont entraîner des évolutions dans le périmètre des services. Il faut que se crée une relation de confiance et tous les référentiels, qui sont des garde-fous, n'y pourront rien.

Reste le Dialogue Compétitif...

Les acteurs publics se retrouvent souvent dans une relation quelque peu conflictuelle, contraints par les règles des marchés publics qui les poussent à choisir le soumissionnaire qui a les meilleures notes, mais pas toujours le mieux adapté à leurs attentes. Quand le contrat est signé, il faut avoir un minimum de confiance et aborder les difficultés, qui sont inévitables, d'une manière positive. Cela se fait notamment par une structure de gouvernance, et par une relation stratégique une fois l'an afin de pouvoir

mettre en oeuvre et exploiter les progrès informatiques.

C'est là que le secteur public a des difficultés. On peut pourtant demander à obtenir un résultat sans être précis sur les moyens, cela s'appelle en effet le Dialogue Compétitif, qui est une source de création de valeur et de maturité entre le fournisseur et le client. On peut également créer des

contrats où la responsabilité de l'infrastructure et des postes de travail est chez le fournisseur, cela s'appelle le Partenariat Public Privé. Compte tenu de la complexité des projets, cela demande aux acteurs d'achats de concevoir autrement la relation qu'ils veulent passer avec leurs fournisseurs... Mais les citoyens seront contents, car ils auront un meilleur service sans augmentation de leurs impôts !

■ Yves Grandmontagne

Les règles des marchés publics poussent les acteurs à choisir le soumissionnaire qui a les meilleures notes, mais pas toujours le mieux adapté à leurs attentes.

Git : boostez votre gestion de configuration

1^{re} partie

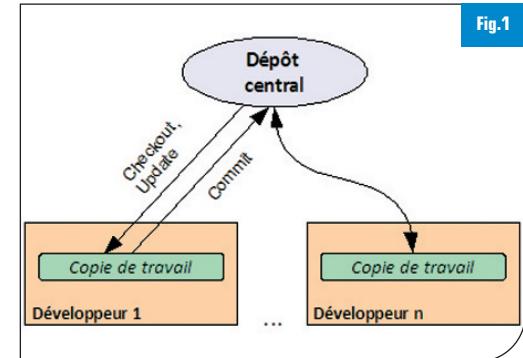
Enjeu majeur du cycle de vie d'une application, la gestion de configuration permet principalement de stocker et de tracer les modifications affectant son code source au cours du temps.

La gestion de configuration a toujours été une problématique centrale au sein des projets informatiques et dont la complexité est directement liée à la taille d'un projet. Plus celui-ci devient important, plus la gestion de configuration peut s'avérer périlleuse. De fait, les innovations dans ce domaine viennent souvent de projets open-source de grande envergure qui par le nombre important de développeurs engagés et leur répartition aux quatre coins du globe ont dû mettre en place des solutions particulières afin d'obtenir une gestion de configuration performante et flexible à la fois. C'est dans ce contexte que Linus Torvalds a lancé le logiciel Git afin de doter la communauté de développement du noyau Linux d'un gestionnaire de configuration répondant à ses besoins particuliers. En effet, si la première décennie de développement du noyau Linux avait vu les développeurs travailler de manière désordonnée via l'échange de patchs ou d'archives de fichiers, le projet s'était organisé de façon plus pérenne dès 2002 avec le basculement vers le logiciel de gestion de configuration distribué (DVCS) propriétaire, BitKeeper. Les tensions entre la communauté et l'éditeur de ce dernier qui remit en cause la gratuité de l'outil en 2005 poussèrent ainsi Linus Torvalds et la communauté du noyau Linux à développer Git pour s'affranchir de BitKeeper. La première version de Git se basait tout naturellement sur l'architecture de ce dernier en corrigeant le tir sur les différentes limites ressenties par les développeurs de la communauté avec BitKeeper. Les objectifs suivis étaient avant tout de fournir un outil simple de conception et totalement distribué qui supporterait les développements non linéaires pouvant compter des milliers de branches parallèles, tout en garantissant une vitesse d'exécution et une compacité des données optimales pour des projets de grande envergure tels que le noyau Linux.

Modèle centralisé

Pour véritablement apprécier les possibilités amenées par les DVCS, et donc Git, il est important d'avoir une connaissance minimale du modèle utilisé par les systèmes de gestion de configuration actuels comme SVN qui se basent sur un modèle centralisé. Un tel modèle implique la présence sur un serveur d'un dépôt central unique [Fig.1] qui possède l'ensemble des fichiers, de leurs versions et des différentes branches éventuelles pouvant exister dans le cas où plusieurs versions du projet doivent être générées en parallèle.

Afin de travailler, les développeurs récupèrent une copie du projet sur leur poste de travail depuis le dépôt central. Cette version récupérée via la commande `checkout` est la copie de travail du développeur sur laquelle il peut travailler et effectuer des modifications. Au quotidien, le développeur va pouvoir mettre à jour sa copie de travail depuis le dépôt central en utilisant la commande `update`. L'envoi des modifications effectuées par un développeur vers le serveur central est réalisée via la commande `commit`. Ce mode de fonctionnement offre aux développeurs la possibilité de travailler en parallèle sur un projet en partageant leurs modifications et en versionnant les fichiers qu'ils modifient. Bien que fonctionnel, ce schéma de travail présente de nombreux défauts. Le plus criant étant que le dépôt central est situé sur un serveur unique. De fait, une panne de ce dernier, ne serait-ce que d'une heure, va empêcher les différents développeurs de collaborer ou bien d'enregistrer leurs modifications locales sur le dépôt central. La communication avec le serveur étant le point d'entrée à la mise jour des modifications effectuées, les développeurs sont contraints de travailler en étant connectés au réseau. En outre, une corruption du disque dur du serveur entraînera, s'il n'y a pas de sauvegarde régulière, une perte totale de l'historique du projet. De fait,



Modèle centralisé de gestion de version.

le modèle centralisé est jugé dépassé par un grand nombre d'entreprises et de développeurs depuis quelque temps maintenant.

Modèle distribué

Devant les limites de la gestion de version centralisée, le modèle distribué sur lequel les DVCS sont fondés a été mis en lumière comme solution et connaît depuis un bel essor dans le sillage de Git. L'absence de dépôt central unique sur un serveur est le fondement de ce modèle où chaque développeur possède son propre dépôt local [Fig.2]. Ainsi, chaque développeur possède sur son poste de travail en local un historique complet des modifications d'un projet. Avec le modèle distribué, la création d'un dépôt local par copie d'un dépôt distant d'un autre client devient une opération assez lourde puisqu'il est nécessaire de copier l'ensemble de l'historique et une grande partie des informations du dépôt distant, là où le modèle centralisé s'attèle seulement à copier le contenu du projet présent sur le dépôt central. De fait, chaque copie de travail est considérée comme un fork du projet du dépôt initial. Cette petite lenteur initiale devient par la suite un atout indéniable, puisqu'à l'instar de la consultation de l'historique du dépôt ou de la réalisation d'un commit, près de 95% des opérations possibles avec Git sont des opérations locales et donc extrêmement rapides pour le développeur. L'époque où les développeurs passaient un temps fou à effectuer des recherches au

sein de l'historique du dépôt devient un lointain et mauvais souvenir avec les gestionnaires de version distribués. Le commit étant une opération locale, le développeur va pouvoir dissocier le versionning de ses modifications du partage de celle-ci. Ainsi, il pourra effectuer des commits plus réguliers étant donné que ceux-ci n'auront pas d'impact sur les autres développeurs du projet et qu'il n'aura pas la crainte de partager une version non entièrement stabilisée d'une fonctionnalité notamment. En outre, les branches locales sont autorisées puisque chaque développeur possède son propre dépôt local. Une nouvelle méthode de travail se met ainsi progressivement en place, à savoir le développement de fonctionnalités au sein de branches spécifiques. Un outil distribué comme Git qui permet par exemple de gérer de manière très rapide des milliers de branches, ce qui est un besoin essentiel du noyau Linux, encourage grandement cette pratique qui évite d'avoir des versions macroscopiques contenant plusieurs fonctionnalités et éventuellement la correction de plusieurs bugs. S'y retrouver là dedans en cas de problème devient par la suite une opération pénible et fastidieuse. Une fois les fonctionnalités associées à ces différentes branches, le développeur va pouvoir appliquer ses modifications sur son dépôt local avant de pousser ses modifications vers le dépôt distant d'un autre client. Le must apporté par Git dans cette opération est la possibilité de ne publier qu'une seule version de modifications à partir de plusieurs versions locales de modifications réalisées. On le voit clairement, le modèle distribué permet un travail déconnecté du réseau puisqu'il n'est nécessaire d'échanger vers d'autres dépôts que pour un nombre très restreint d'opérations. Ainsi, plus de pro-

blèmes de disponibilité en cas de défaillance du serveur central puisque celui-ci n'est en rien obligatoire. En outre, chaque copie d'un dépôt pour création d'un dépôt local devient une sauvegarde du dépôt initial avec tout l'historique en son sein. La corruption d'un disque dur n'est ainsi plus une crainte. Enfin, le mode distribué offre des possibilités quasiment infinies quant à la mise en place de modèles de collaboration entre développeurs. Ces modèles peuvent être des plus simples sans serveur faisant office de dépôt central comme on peut le voir sur la [Fig.2] ou bien beaucoup plus complexes avec des niveaux hiérarchiques par exemple comme cela est nécessaire pour la gestion de version du noyau Linux.

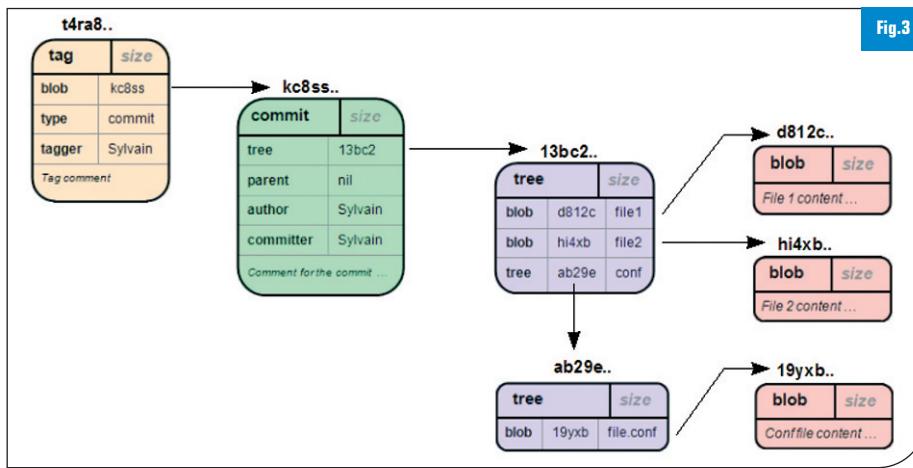
Architecture de Git

Techniquement parlant, le cœur de Git est une simple base de données de type clé-valeurs contenant l'ensemble des informations nécessaires à la gestion des versions et de l'historique d'un projet. L'ensemble des informations de cette base est stocké au sein d'un unique répertoire caché, nommé .git, qui est présent à la racine d'un projet géré par Git. Là où SVN pollue inutilement votre projet en plaçant un dossier spécifique de configuration pour chacun de ses répertoires, Git va puiser au sein de ce seul répertoire l'ensemble des informations nécessaires à la gestion de votre projet tant au niveau de son historique que des objets versionnés, sans oublier la gestion des branches. Le contenu du projet est stocké sous forme d'objets adressables qui peu-

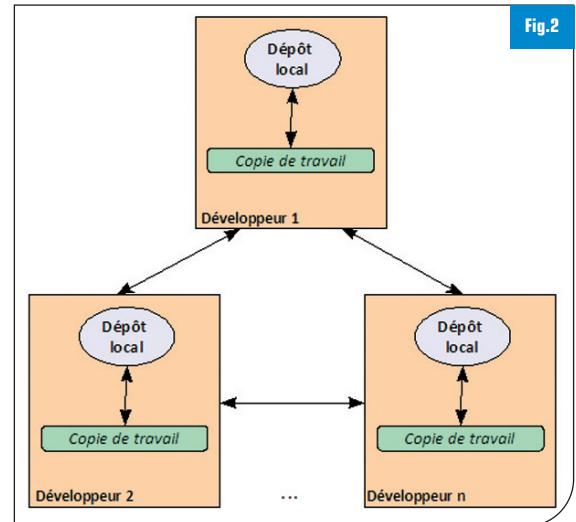
vent être de 4 types différents mais possédant tous 3 informations de base que sont un type, une taille et un contenu [Fig.3]. Ces types sont les suivants :

- Blob, ce type d'objet représente le contenu d'un fichier de manière binaire. En interne, Git compresse ces objets afin de gagner en espace mémoire et de réduire la consommation de bande passante.
- Tree, qui représente le contenu d'un unique répertoire et qui contient une liste de pointeurs vers d'autres objets trees ou blobs (sous-répertoires et fichiers).
- Commit, qui pointe vers un unique objet tree et le marque afin de représenter le projet à un instant t dans le temps. Il lie l'état physique d'un tree à la description de la manière et de la raison de son arrivée dans cet état.
- Tag, qui permet de marquer un commit spécifique d'un projet, ce qui peut correspondre par exemple à la version spécifique d'un projet.

Afin de pouvoir adresser directement chacun de ces objets, Git associe à chacun d'entre eux un identifiant unique calculé à l'aide de la fonction cryptographique SHA-1 à partir du contenu de l'objet en question. De fait, chaque objet possède un identifiant unique de 40 caractères qui lui est propre et qui va grandement faciliter et accélérer les opérations de comparaison d'objets notamment. En effet, pour savoir si 2 objets sont identiques il suffira à Git de comparer leurs identifiants respectifs. Dans le cas d'objets trees représentant une hiérarchie complexe de répertoires, le gain de temps est énorme puisqu'il n'y aura aucun parcours récursif à effectuer pour comparer le contenu. En outre, l'intégrité d'un fichier

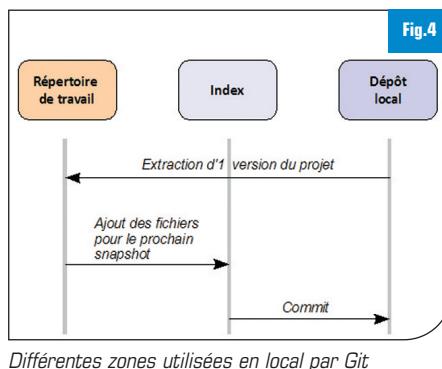


Vue de la base objet de Git à partir d'un tag



Modèle distribué de gestion de version.

devient facilement vérifiable puisqu'il est possible à tout moment de comparer son contenu à son identifiant. Ces différents identifiants sont stockés comme des références par Git afin de pouvoir adresser l'ensemble des informations qu'il gère. Si ces références sont efficaces d'un point de vue technique pour Git, elles ne sont pas forcément pratiques pour le développeur qui les utilise. Afin de faciliter leur utilisation, Git les reconnaît également de manière abrégée via les premiers caractères d'un identifiant. Mieux encore, le concept de références symboliques est introduit qui offre la possibilité de nommer une référence via une chaîne de caractères. Par défaut, Git propose ainsi le raccourci HEAD qui va pointer sur la branche courante de travail. De nombreux autres raccourcis sont proposés et leur utilisation est très puissante pour l'utilisateur. Ainsi, Git reconnaît par exemple une expression telle que HEAD~2 comme la désignation du "grand-parent" du commit pointé par HEAD. Sur le poste client en local, Git considère 3 zones distinctes afin de gérer les fichiers d'un projet [Fig.4]. La première d'entre-elles est le répertoire de travail qui contient une version temporaire du projet sur laquelle le développeur travaille en modifiant ses fichiers jusqu'au prochain commit sur le dépôt local. Ce dernier se charge de stocker les différentes versions des fichiers d'un projet ainsi que l'historique des modifications associées. Enfin, entre ces deux zones se trouve une zone d'assemblage, également connue sous le nom d'Index, rattachée uniquement au dépôt local et utilisée pour construire un ensemble de modifications qui seront commitées ensemble par la suite. Cette zone est un atout indéniable de Git puisqu'elle permet une séparation claire entre la modification des fichiers sur l'espace de travail du développeur et le commit de certains de ces fichiers sur le dépôt local. Lors de l'ajout d'un fichier au sein de l'index, Git va y placer un snapshot du contenu du fichier à l'instant précis de cet ajout. Ainsi lors d'un commit, Git va transférer les snapshots de fichiers présents dans l'index vers le dépôt local et non le contenu de ces fichiers dans le répertoire de travail à l'instant précis où le commit est réalisé. Cette notion de snapshot est essentielle puisque contrairement à un outil comme SVN, Git ne va pas gérer les différentes versions de fichiers en stockant l'information comme des modifications de fichiers mais bien comme des snapshots du projet au cours du temps.



Installation

Les fondements théoriques essentiels de Git mis en place, nous passons maintenant à l'installation de notre DVCS. Disponible évidemment en natif sur les systèmes de type Linux, Git bénéficie cependant d'un client Windows de bonne qualité bien que moins complet. Néanmoins, ces manques ne se font ressentir que dans des utilisations très poussées et spécifiques de Git, ce qui n'est pas notre cas dans le cadre de cet article. Nommé MsysGit, ce client est disponible à l'adresse suivante : <http://code.google.com/p/msysgit/> et c'est lui que nous allons utiliser pour prendre en main Git. Au cours de l'installation, le client propose d'intégrer les outils de ligne de commande de type Unix directement au sein du terminal de Windows pour en faciliter l'utilisation mais cela n'est pas une nécessité. Une fois ce choix fait, l'installation se termine sans réelles difficultés. L'utilisation de Git se fait alors via le terminal Windows ou via un outil ligne de commande de type Unix installé par MsysGit.

Premiers pas

Au sein de l'outil ligne de commande, Git est utilisable via l'exécutable éponyme git. Celui-ci propose une auto-complétion des commandes associées facilitant ainsi l'utilisation de Git au quotidien. En outre, la commande help fournit une aide complète sur chacune des commandes disponibles via la syntaxe suivante : git help <command>. Git possède un certain nombre d'options de configuration qu'il est possible de renseigner de manière globale afin que chacun des projets que vous utilisez sur votre poste local prenne en compte celles-ci. La mise en place de ces options se fait via la commande config de Git. Cette dernière se charge ensuite de stocker vos préférences au sein d'un fichier .gitconfig, présent dans votre dossier utilisateur, sous la forme de couples clés-valeurs. Pour commencer, nous allons donc setter un nom d'utilisateur et un email spéci-

fiques pour notre utilisation de Git comme suit :

```
git config --global user.name "sylvain
.saurel"
git config --global user.email "sylvain
.saurel@gmail.com"
```

Ces informations renseignées, nous allons enfin pouvoir créer notre premier dépôt local. Cette création peut se faire de 2 manières différentes. La première consiste à cloner un dépôt distant grâce à la commande git clone. Il peut être intéressant de tester ce mode de création mais dans notre cas, nous créerons plutôt notre propre dépôt local pour un projet partant de zéro afin de mieux mettre en évidence le démarrage d'un projet Git. Dans un second temps, il deviendra ensuite intéressant de regarder du côté d'un site tel que GitHub par exemple qui héberge un grand nombre de projets open source stockés chacun sur leur propre dépôt. Vous pourrez ainsi en choisir un et le forker via un clone afin de vous faire la main sur un projet possédant déjà un historique complet. Pour créer un dépôt local vierge, nous nous plaçons au sein du répertoire vierge de notre projet et nous exécutons la commande git init. Cette dernière a pour effet la création du répertoire de configuration .git à la racine de notre projet. Ce répertoire contient, entre autres, un dossier *objects* qui contiendra les futurs objets mis en gestion de version, un dossier *refs* qui contiendra les futures références de notre projet ainsi qu'un fichier HEAD dont le contenu pointe sur la branche courante du projet. Une fois un premier fichier README créé à la racine du projet, nous exécutons la suite de commandes :

```
git add .
git commit -m 'Premier commit du projet'
```

La commande *add* va permettre d'ajouter le contenu du répertoire courant à l'index Git à l'instant où l'exécution de la commande est réalisée. Ainsi, les éventuels changements du contenu des fichiers du répertoire sans nouvel ajout à l'index ne seront pas commités. C'est ici que rentre en jeu la notion de snapshot évoquée précédemment dans cet article. Enfin, la seconde commande réalise le commit des fichiers ajoutés dans la zone d'index. Suite à l'exécution d'un commit, Git détaille un certain nombre d'informations telles que la branche sur laquelle le commit a été réalisé et des statistiques sur les fichiers affectés par ce commit.

■ Sylvain Saurel
Ingénieur d'Etudes Java / JEE
sylvain.saurel@gmail.com

Devenir un développeur iOS

3^e partie

Dans nos précédents articles, nous avons vu comment créer le squelette d'une application et comment coder ses premières fonctionnalités en nous basant sur l'un des contrôles les plus utilisés au sein d'iOS, à savoir « UITableView » et son contrôleur associé « UITableViewcontroller ».



APPLICATION : iOS

LANGAGE : Objective-C

SOURCE : Non

Notre application est pour le moment capable de lister des informations et de les afficher sous forme d'un tableau mais il nous manque encore le principal, à savoir, être capable d'ajouter ou modifier des données au moyen de nouveaux écrans, vers lesquels il devra être possible de naviguer de manière simple et rapide. Le but de ce 3^e volet va donc être de se concentrer sur l'implémentation de plusieurs écrans au sein de notre application, d'échanger des données entre ces écrans et d'agrémenter tout cela de quelques effets visuels (également appelés transitions).

IMPLÉMENTATION DE LA NAVIGATION

Avant de commencer à implémenter nos écrans, commençons par ajouter une première manière de naviguer au sein de notre application, notamment en permettant à l'utilisateur de pouvoir ajouter des données au sein de notre application de gestion de contacts.

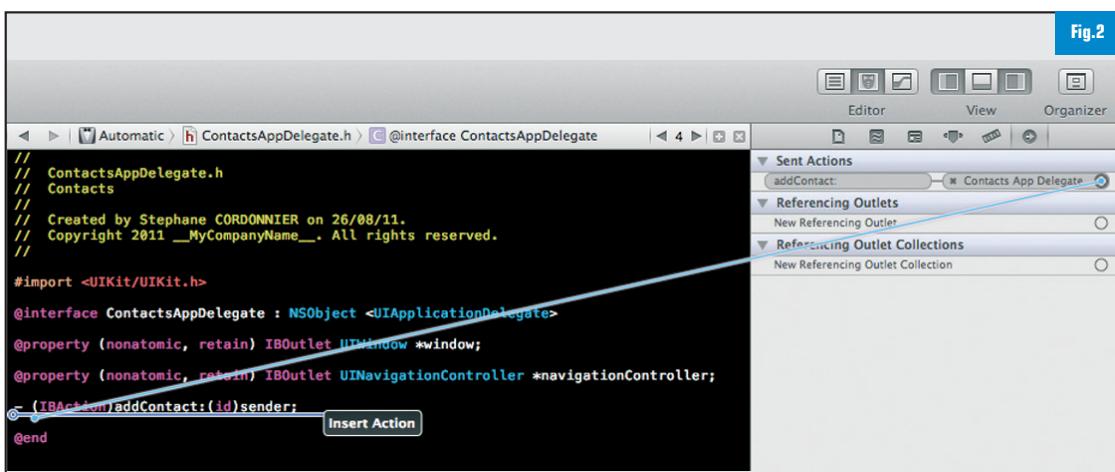
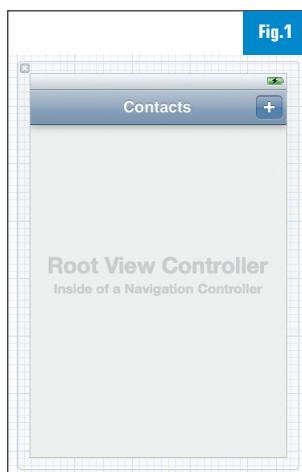
Pour cela, ouvrons la vue nommée « **MainWindow.xib** » et ajoutons un contrôle de type « **Bar Button Item** » au sein de la barre de navigation supérieure de notre application. Dans l'inspecteur de propriétés de ce bouton (accessible via le menu « **View > Utilities > Show Attributes Inspector** »), sélectionnons « **Bordered** » pour la propriété « **Style** » et « **Add** » pour la propriété « **Identifier** ».

Ces propriétés permettent de jouer sur le rendu du bouton et plus particulièrement sur le type de bordure, la couleur et l'icône apparaissant dans celui-ci. Vous pouvez vous amuser à tester différentes combinaisons pour voir les styles prédéfinis au sein d'iOS.

Le résultat auquel vous devriez arriver doit ressembler à ceci : [Fig.1].

Ajoutons ensuite un gestionnaire d'événements pour le bouton. Pour cela et toujours en sélectionnant le bouton, affichons l'inspecteur de

Notre application est pour le moment capable de lister des informations et de les afficher sous forme d'un tableau mais il nous manque encore le principal, à savoir,



connexions ainsi que l'assistant d'édition (aller dans le menu « **View > Assistant Editor > Show Assistant Editor** »).

Dans l'inspecteur d'actions, cliquons sur le rond à côté de l'entrée « **Selector** » et tout en maintenant le bouton de la souris enfoncé, tirons un trait jusqu'à la partie droite fenêtre d'édition du code. Lorsque nous relâchons le bouton de la souris, une fenêtre s'ouvre nous invitant à saisir le nom du gestionnaire d'évènements que nous voulons créer.

Dans le cas de notre application, utilisons « **addContact** » comme nom de méthode, ce qui est parlant pour l'utilité que nous voulons donner à notre bouton [Fig.2]. Si nous ouvrons le fichier qui va nous permettre d'ajouter le code de notre gestionnaire d'évènements (normalement le fichier « **ContactsAppDelegate.m** »), vous devriez voir apparaître l'implémentation vide de notre gestionnaire.

Ajoutons un bout de code relativement simple pour nous assurer que la liaison entre le bouton et le code fonctionne correctement. Ce code est juste chargé d'afficher une boîte de dialogue avec un message à l'intérieur.

```
- (IBAction)addContact:(id)sender
{
    UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"Contacts" message:@"Add Contact" delegate:nil cancelButtonTitle:@"OK" otherButtonTitles:nil, nil];
    [alert show];
    [alert release];
}
```

Si nous exécutons l'application et que nous cliquons sur le bouton que nous avons ajouté à notre barre de navigation, la boîte de dialogue s'affiche correctement à l'écran et nous pouvons cliquer sur le bouton « **OK** ».

IMPLÉMENTATION DE L'ÉCRAN D'AJOUT DE DONNÉES

Maintenant que notre navigation principale fonctionne, passons à l'étape suivante, à savoir, ajouter un écran à notre application, écran qui sera affiché lorsque l'utilisateur cliquera sur le bouton que nous venons d'ajouter à l'étape précédente.

Pour cela, commençons par ajouter un nouveau fichier au sein de notre projet en nous rendant dans le menu « **File > New > New File** ». L'assistant d'ajout de fichier s'ouvre alors et nous invite à sélectionner le type de fichier que nous souhaitons ajouter. Puisque nous souhaitons ajouter un écran et que nous travaillons dans le modèle MVC, il faut ajouter un « **UIViewController subclass** » et cliquer sur « **Next** ». Il faut ensuite indiquer que nous voulons gérer la vue de manière visuelle en cochant la case « **With XIB for user interface** ». Notre application étant destinée à être utilisée sur un iPhone, nous pouvons décocher la case « **Targeted for iPad** ». Indiquons ensuite la valeur « **ViewContactController** » comme nom de classe et validons nos choix.

Une fois tout cela effectué, 3 fichiers ont dû être ajoutés à notre projet à savoir :

- **ViewContactController.h**
- **ViewContactController.m**
- **ViewContactController.xib**

Commençons par éditer le fichier correspondant à l'interface utilisateur à savoir le fichier XIB. Ajoutons au sein de ce fichier les élé-

ments qui nous serviront pour saisir des contacts à savoir des « **Labels** » et des « **Text Field** ». Une fois les contrôles ajoutés et la mise en page effectuée, changeons quelques propriétés de ces contrôles, notamment sur les libellés pour leur donner des intitulés « parlant ». Pour cela, allons dans l'inspecteur de propriétés utilisé précédemment et changeons la propriété « **Text** ».

Comme nous pouvons le constater dans l'inspecteur de propriétés, de très nombreuses propriétés sont à notre disposition pour influer sur le comportement, le rendu visuel, etc. de nos contrôles.

Vous pouvez jouer avec les différentes propriétés à votre disposition pour comprendre l'utilité de chacune d'entre elles, et les impacts qu'elles peuvent avoir sur le rendu visuel des contrôles. Il est bon de noter que chaque contrôle possède des propriétés qui lui sont propres, même si vous constaterez que certaines propriétés peuvent être retrouvées sur les mêmes contrôles.

Le rendu final de notre écran est visible avec 3 labels et 3 zones de saisie qui nous serviront à saisir le nom de famille, le prénom et le numéro de téléphone de nos contacts [Fig.3].

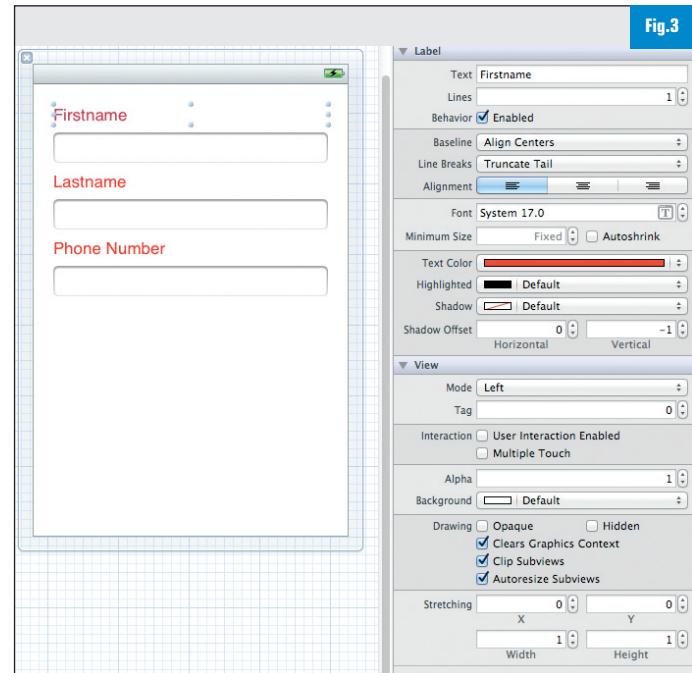
Les contrôles sont positionnés dans notre écran, mais le travail n'est toutefois pas terminé pour autant. En effet, nous allons avoir besoin de manipuler certains contrôles dans notre code, notamment les labels, afin de récupérer les valeurs qui auront été saisies par l'utilisateur. Pour cela il est donc nécessaire de faire en sorte que le contrôle soit associé à une variable qui permettra de manipuler les contrôles par programmation. Pour cela, nous allons réutiliser la même méthode que pour le gestionnaire d'évènements que nous avons mis en place précédemment.

Faisons donc apparaître l'assistant d'édition et le gestionnaire de connexions. Sélectionnons ensuite les 3 labels un par un et dans la fenêtre de connexion, effectuons un glisser/déplacer depuis la ligne « **New Referencing Outlet** » vers la fenêtre de code.

Pour chacun des 3 labels, donnons leurs des noms tels que :

- **firstNameLabel**
- **lastNameLabel**
- **phoneNumberLabel**

Les 3 lignes suivantes doivent avoir été ajoutées au sein du fichier



« **ViewContactController.h** ». Ces lignes permettent de déclarer de nouvelles propriétés au sein du contrôleur associé à notre vue, pour permettre à celui-ci de pouvoir manipuler les contrôles et notamment récupérer les valeurs qui auront été saisies par l'utilisateur.

```
@property (retain, nonatomic) IBOutlet UITextField *firstNameLabel;
@property (retain, nonatomic) IBOutlet UITextField *lastNameLabel;
@property (retain, nonatomic) IBOutlet UITextField *phoneNumLabel;
```

Dans le fichier « **ViewContactController.m** », 3 autres lignes ont été automatiquement ajoutées. Ces lignes permettent de ne pas avoir à implémenter de manière manuelle les accesseurs de nos propriétés précédentes.

```
@synthesize firstNameLabel;
@synthesize lastNameLabel;
@synthesize phoneNumLabel;
```

Si vous regardez attentivement le fichier « **ViewContactController.m** », vous constaterez que d'autres modifications sont intervenues, notamment dans les méthodes concernant la gestion de la mémoire. Xcode a ajouté automatiquement le code permettant de libérer les ressources allouées aux propriétés qui avaient été déclarées préalablement.

```
- (void)viewDidLoad
{
    [self setFirstNameLabel:nil];
    [self setLastNameLabel:nil];
    [self setPhoneNumLabel:nil];
    [super viewDidLoad];
}

- (void)dealloc
{
    [firstNameLabel release];
    [lastNameLabel release];
    [phoneNumLabel release];
    [super dealloc];
}
```

Il est important de noter que cette manière de gérer la mémoire dans iOS 4 est appelée à disparaître dans iOS 5 avec « **Automatic Reference Count** ». Cette notion permettra de faire en sorte que tous les mécanismes de « **Retain/Release** » pour gérer la mémoire, soient faits de manière automatique. Il ne faut pas confondre ceci avec le « **Garbage Collector** » de .NET mais cela simplifiera grandement la vie des développeurs qui n'auront plus (en théorie) à gérer manuellement les allocations que nous pouvons voir dans notre application présente.

AFFICHAGE DE L'ÉCRAN D'AJOUT DE DONNÉES

Maintenant que notre écran d'ajout de données est terminé d'un point de vue esthétique (nous implémenterons le fonctionnel ultérieurement), voyons comment afficher celui-ci au sein de notre application lorsque l'utilisateur cliquera sur le bouton d'ajout que nous avons créé précédemment. Pour cela, commençons par supprimer le code faisant apparaître la boîte de dialogue que nous avions créée

pour faire un premier test tout à l'heure. Remplaçons ce code par celui qui nous permettra d'afficher notre écran, tout en gérant la barre de navigation supérieure de notre application (le « **UINavigationController** » intégré dans notre article précédent). Si nous regardons attentivement les déclarations présentes dans le fichier « **ContactsAppDelegate.h** », nous constaterons qu'une propriété nommée « **navigationController** » avait été automatiquement ajoutée lorsque nous avions créé notre projet. Cette propriété va nous permettre de pouvoir demander au contrôleur de la barre de navigation de pousser une autre vue en lieu et place de notre tableau de contacts, pour pouvoir saisir une donnée lorsque l'utilisateur cliquera sur le bouton d'ajout. La démarche pour réaliser cela est assez simple et se passe en 3 étapes :

- 1 Déclarer une variable et instancier le contrôleur associé à notre vue.
- 2 Demander au contrôleur de la barre de navigation d'afficher la vue.
- 3 Libérer la mémoire allouée.

Dans la pratique et si l'on traduit tout ceci par du code, cela donne ce qui suit :

```
- (IBAction)addContact:(id)sender
{
    ViewContactController *vcc = [[ViewContactController alloc] init];
    [self.navigationController pushViewController:vcc animated:YES];
    [vcc release];
}
```

Comme nous pouvons le constater, la partie la plus importante pour afficher la vue concerne l'appel à la méthode « **pushViewController** ». Cette méthode prend en paramètre le contrôleur que l'on souhaite afficher à l'écran.

Plus précisément, ce sera la vue par défaut du contrôleur qui sera affichée mais c'est le contrôleur qui doit être passé en paramètre. Il est aussi possible de spécifier si l'affichage de la vue se fait de manière animée ou non (mettez « **animated :NO** » si vous voulez désactiver l'animation de transition lors de l'affichage de la vue). Si nous exécutons notre application dès à présent, nous pouvons constater que celle-ci est désormais fonctionnelle et que lorsque nous cliquons sur le bouton d'ajout, la vue créée précédemment avec nos 3 labels, apparaît à l'écran.

Le fait d'utiliser la barre de navigation principale (et la méthode « **pushViewController** »), fait qu'automatiquement, l'utilisateur peut revenir à l'écran précédent en cliquant sur la flèche présente en haut à gauche de cette barre de navigation.

AFFICHAGE DE CONTRÔLEURS/VUES

Faisons juste un petit aparté concernant l'affichage de vues au sein des applications iOS. Nous venons de voir un exemple basé sur la barre de navigation (UINavigationController) mais il est possible d'afficher des écrans d'une autre manière au sein des applications iOS. En effet, chaque contrôleur qui dérive de la classe « **UIViewController** » peut afficher une vue en utilisant la méthode « **presentModalViewController** », qui a la particularité d'afficher une vue en plein écran, afin que l'utilisateur ne puisse interagir avec rien d'autre que la vue souhaitée (d'où la notion de modale).

La signature de la méthode en question est la suivante :

```
- (void)presentModalViewController:(UIViewController *)modalViewController animated:(BOOL)animated
```

Là aussi, l'affichage d'une vue peut être réalisée de manière animée

mais les styles à disposition des développeurs sont plus nombreux qu'avec l'utilisation de la barre de navigation. Les styles à disposition sont au nombre de quatre :

- `UIModalTransitionStyleCoverVertical`
- `UIModalTransitionStyleFlipHorizontal`
- `UIModalTransitionStyleCrossDissolve`
- `UIModalTransitionStylePartialCurl`

En cas d'exécution sur un iPad, la présentation des vues peut même être réalisée de différentes manières parmi les suivantes :

- `UIModalPresentationFullScreen`
- `UIModalPresentationPageSheet`
- `UIModalPresentationFormSheet`
- `UIModalPresentationCurrentContext`

Il existe également d'autres possibilités sur l'iPad comme l'utilisation des `UIPopoverController` pour afficher des vues au sein de fenêtre flottantes, en fonction de l'endroit où l'utilisateur clique sur l'écran. Ces différents modes de présentation pouvant faire l'objet d'articles en entier pour aborder leur présentation et mode de fonctionnement, nous ne nous y attarderons pas plus ici.

AFFICHAGE DES DONNÉES SÉLECTIONNÉES DANS LE TABLEAU

Maintenant que notre écran s'affiche lorsque nous cliquons sur le bouton d'ajout présent dans la barre de navigation, voyons comment implémenter un comportement similaire lorsque l'utilisateur sélectionne une ligne dans le tableau présent sur la page d'accueil de l'application (pour en afficher les détails par exemple).

La première étape va consister à déterminer, quelle ligne du tableau a été sélectionnée par l'utilisateur. Pour cela, rien de plus simple puisque le protocole « `UITableViewDelegate` » de notre tableau nous donne cette information via la méthode « `didSelectRowAtIndexPath` ». La signature de cette méthode est la suivante :

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath
```

Comme nous le constatons, celle-ci nous offre un paramètre « `indexPath` » qui nous indique le numéro de section du tableau, ainsi que le numéro de ligne au sein de la section, qui a été sélectionnée. A partir de ces informations, il suffit de se référer à la source de données du tableau, pour récupérer l'information voulue pour la traiter de la manière adéquate.

Dans notre application, nous n'avons pas encore de source de données servant à alimenter notre tableau (nous le ferons dans le prochain article), aussi nous utiliserons simplement le numéro de ligne pour simuler le fait que nous avons récupéré la bonne valeur, celle qui correspond à la cellule sélectionnée par l'utilisateur.

Une fois la cellule déterminée et la valeur trouvée, il ne reste plus qu'à demander à notre application d'afficher la vue créée précédemment, mais en effectuant quelques modifications sur le comportement de celle-ci. En effet, vu que nous souhaitons uniquement faire de la consultation de données, il ne faut pas que l'utilisateur puisse saisir des valeurs dans les zones de textes. Nous allons donc retirer les bordures de celles-ci et désactiver la possibilité pour l'utilisateur de pouvoir interagir avec elles.

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath
```

```
{
    ViewContactController *vcc = [[ViewContactController alloc] init];
    [self.navigationController pushViewController:vcc animated:YES];

    vcc.firstNameLabel.borderStyle = UITextBorderStyleNone;
    vcc.firstNameLabel.text = [NSString stringWithFormat:@"FirstName #%@", indexPath.row];
    vcc.firstNameLabel.enabled = NO;

    vcc.lastNameLabel.borderStyle = UITextBorderStyleNone;
    vcc.lastNameLabel.text = [NSString stringWithFormat:@"LastName #%@", indexPath.row];
    vcc.lastNameLabel.enabled = NO;

    vcc.phoneNumberLabel.borderStyle = UITextBorderStyleNone;
    vcc.phoneNumberLabel.text = [NSString stringWithFormat:@"PhoneNumber #%@", indexPath.row];
    vcc.phoneNumberLabel.enabled = NO;

    [vcc release];
}
```

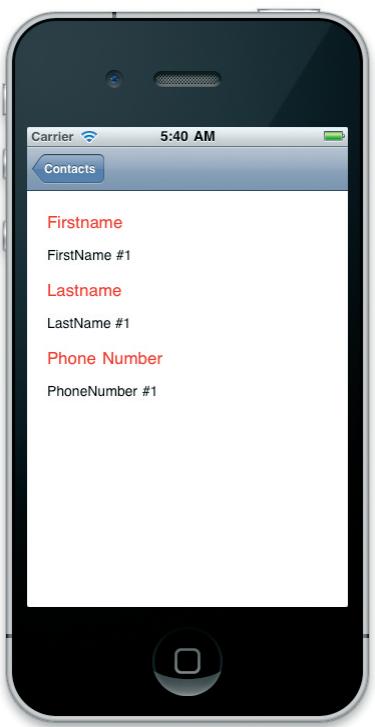
Si nous exécutons notre application, nous constatons que celle-ci est désormais entièrement fonctionnelle à savoir que nous pouvons afficher des données dans le tableau, sélectionner une cellule pour en consulter les détails (en lecture seule) ou bien demander d'ajouter un nouveau contact dans la liste.

Pour l'instant, l'écran permettant d'ajouter de nouvelles données ne permet pas d'enregistrer celles-ci au sein de l'application mais nous verrons dans le prochain article comment réaliser cette opération.

CONCLUSION

Notre application commence à prendre la forme que nous voulions lui donner mais il reste encore le plus important à voir, à savoir aller lire et écrire les données correspondant à nos contacts (nom de famille, prénom et numéro de téléphone) au sein d'une base de données afin de les rendre persistantes dans notre application.

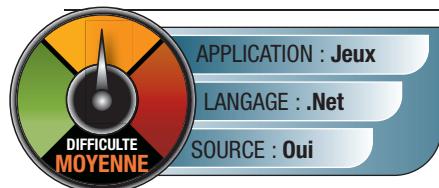
Cette étape fera l'objet du dernier article de cette série dans lequel nous verrons les différentes possibilités pour stocker des données en utilisant CoreData ou SQLite comme référentiel de données.



■ Stéphane Cordonnier
Directeur – iTouch
<http://www.itouchconsulting.com>

XNA sur tous les fronts

Après avoir découvert les fondamentaux de la 3D dans un précédent article (Initiation à la 3D avec Direct3D 11 - *PROgrammez !* 143), nous allons nous pencher sur la technologie XNA qui va nous permettre de mettre en application nos connaissances sur de nombreuses plateformes.



pour la 3D en environnement managed [Fig.1]. Ainsi XNA 4.0 se décline sur Windows, sur Xbox 360, Windows Phone 7 (et Zune) et même sur Silverlight 5.

Sur chacune de ces plateformes le principe de fonctionnement reste le même. Chaque environnement apportera son lot de limitations par rapport à la version Windows qui naturellement est la plus riche mais le portage de l'un vers l'autre est extrêmement aisé (voire parfois automatique).

XNA s'appuie sur la technologie DirectX 9 qui est présente sur chacun des systèmes cibles. Il s'appuie également sur le framework .NET, ce qui facilite grandement la tâche des développeurs qui n'ont pas besoin de faire du C++ pour atteindre les API graphiques.

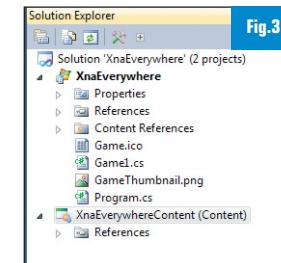
Le développement avec XNA se fait via Visual Studio 2010 (que ce soit en version standard ou en version express gratuite) ou l'on trouve les modèles de projets adéquats : [Fig.2].

Nous allons donc dans un premier temps faire un projet « hello world » sur Windows puis nous verrons comment le porter vers les autres systèmes.

Mise en place sur PC

Une fois notre modèle de projet choisi, nous nous retrouvons devant une solution composée de deux éléments :

- Notre projet principal : C'est ici que nous ferons notre développement avec notamment la boucle de rendu principale
- Le projet de contenus (assets) : Ce projet a pour but de produire les ressources (graphiques, fontes, sons, shaders) nécessaires au projet principal [Fig.3].



Cette dualité peut paraître étrange mais à l'usage c'est une mécanique qui s'avère redoutable. En effet, le projet de contenus se chargeant de préparer les données dans un format spécialisé pour XNA, le chargement des ressources lors de l'exécution de notre application va se révéler très rapide. Dans

les faits, ce projet est un projet « de compilation » en ce sens qu'il est exécuté à la compilation pour rajouter des données dans le projet principal et n'a aucune existence lors du runtime.

De plus, il ne sera pas nécessaire d'embarquer dans notre application tout le code requis pour charger les formats divers et variés que l'on utilise puisque tout aura été préparé à la compilation.

Définition des assets

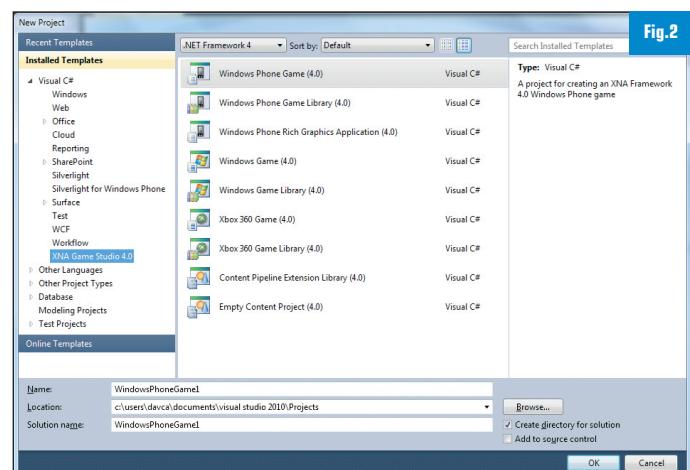
Nous allons donc rajouter (par simple glisser/déposer) dans notre projet de contenus, les assets (les ressources) que nous souhaitons utiliser. Elles se composent de :

- Un modèle 3D au format FBX (format d'Autodesk très répandu) : *Ship.fbx*
- La texture du modèle : *ShipDiffuse.tga*

Notre projet de contenu va alors se charger d'ouvrir le format FBX et va produire un fichier utilisable par notre projet principal. Il se chargera également de récupérer la texture et l'affectera à l'objet. Pour cela, il va copier des fichiers au format .xml et .xbn dans les répertoires de compilation du projet principal qui se chargera de les embarquer en tant que ressources : [Fig.4]

Chargement des ressources

Nous allons donc maintenant passer dans notre code pour récupérer ces informations. Le projet principal s'architecture autour d'une classe qui hérite de Game et qui représente le squelette de notre application. Dans notre cas, notre classe se nomme par défaut



Game1 et surcharge un certain nombre de méthodes pour nous permettre de faire ce que nous souhaitons :

LoadContent : Cette méthode sera utilisée pour charger nos ressources produites par le projet de contenus. XNA nous fournit pour cela une classe : le ContentManager dont une instance (Content) est présente sur notre classe Game1.

- **UnloadContent** : On peut se servir de cette méthode pour libérer des ressources que nous aurions créées sans passer par le ContentManager

- **Update** : Cette méthode servira à mettre à jour les informations de notre application pour la rendre dynamique

- **Draw** : Cette méthode permettra de dessiner nos modèles

Ainsi notre méthode LoadContent va se présenter ainsi :

```
protected override void LoadContent()
{
    model = Content.Load<Model>("Ship");
}
```

Notre objet model est de classe Model et sert à stocker toutes les informations pour afficher un modèle 3D. L'objet Content se charge donc de récupérer la ressource .xnb associée à notre modèle et génère un objet de classe Model à partir de ces informations.

Animations

Pour que notre rendu ne soit pas statique nous ferons tourner notre objet sur lui-même sur son axe des Y (l'axe vertical). Pour cela, nous utiliserons un variable angle que nous ferons progresser avec le temps. Le meilleur endroit pour gérer ce genre de modifications est la méthode Update. En effet, cette dernière est fournie pour permettre au développeur de gérer sa logique de fonctionnement. Elle est cadencée de pair avec la méthode Draw qui permet de faire le dessin de notre application.

Si la propriété IsFixedTimeStep de notre classe Game1 est vraie, la méthode Update sera appelée à la fréquence définie par TargetElapsed (1/60^e de seconde par défaut). Après avoir appelé Update, s'il n'est pas encore temps d'appeler Update, le système appellera Draw. Cela garantit que la méthode Update est bien appelée selon la fréquence voulue. Si le système n'a pas le temps d'appeler Draw (et donc s'il supprime certains rendus), il positionnera la variable IsRunningSlowly à vrai.

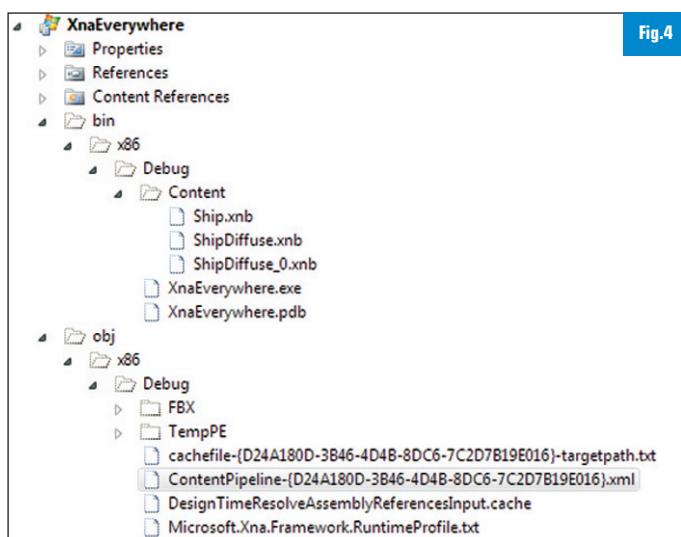


Fig.4

Dans notre cas, pas de risque là-dessus, puisque la méthode est très simple :

```
protected override void Update(GameTime gameTime)
{
    // Allows the game to exit
    if (GamePad.GetState(PlayerIndex.One).Buttons.Back == ButtonState.Pressed)
        Exit();

    angle += 0.01f;

    base.Update(gameTime);
}
```

Nous vérifions que l'utilisateur ne veut pas quitter et sinon nous avançons la valeur de notre angle.

Rendu

La phase de rendu est donc effectuée dans la méthode Draw. C'est dans cette dernière que nous appellerons le code suivant :

```
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.Gray);

    GraphicsDevice.BlendState = BlendState.Opaque;
    GraphicsDevice.DepthStencilState = DepthStencilState.Default;
    GraphicsDevice.SamplerStates[0] = SamplerState.LinearWrap;

    foreach (ModelMesh mesh in model.Meshes)
    {
        foreach (BasicEffect effect in mesh.Effects)
        {
            effect.World = Matrix.CreateRotationY(angle);
            effect.View = Matrix.CreateLookAt(new Vector3(0, 0, 2000), Vector3.Zero, Vector3.Up);
            effect.Projection = Matrix.CreatePerspectiveFieldOfView(0.8f, GraphicsDevice.Viewport.AspectRatio, 0.1f, 10000.0f);
            effect.EnableDefaultLighting();
        }

        mesh.Draw();
    }

    base.Draw(gameTime);
}
```

Nous demandons pour chaque sous-partie de notre modèle (les meshes) à définir plusieurs valeurs (pour de plus amples informations sur les matrices, je vous renvoie à l'article précédent (*Initiation à la 3D avec DirectX 11*) :

- La matrice monde (World) : C'est ici que nous précisons notre souhait que l'objet tourne sur lui-même en fonction de l'angle
- La matrice de vue (View) : Nous définissons notre point de vue avec notre position, le point que l'on regarde et vers où se situe le haut de notre monde
- La matrice de projection (Projection) : Elle définit comment passer du monde 3D au monde 2D de notre écran

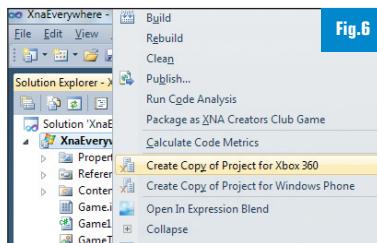
- Le modèle d'éclairage : Pour ne pas passer du temps à définir une source lumineuse, nous pouvons demander à l'effet de définir une modèle d'éclairage standard.

Finalement, il ne reste plus qu'à dessiner chaque mesh via leur méthode Draw.

Résultat

Au final, nous nous trouvons avec une fenêtre dans laquelle tourne un superbe vaisseau spatial texturé et éclairé ! [Fig.5].

Nous allons donc maintenant porter notre projet vers d'autres plateformes pour leur faire profiter de cette superbe réalisation.



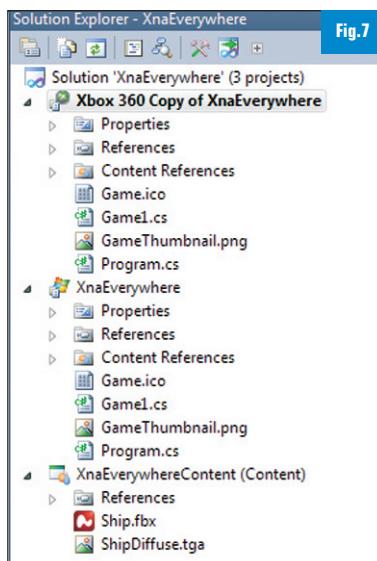
XNA sur Xbox 360

XNA fut à la base conçu pour faire du développement croisé Windows/Xbox 360.

C'est donc sans surprise que le portage vers la Xbox 360 va se faire de manière automatique.

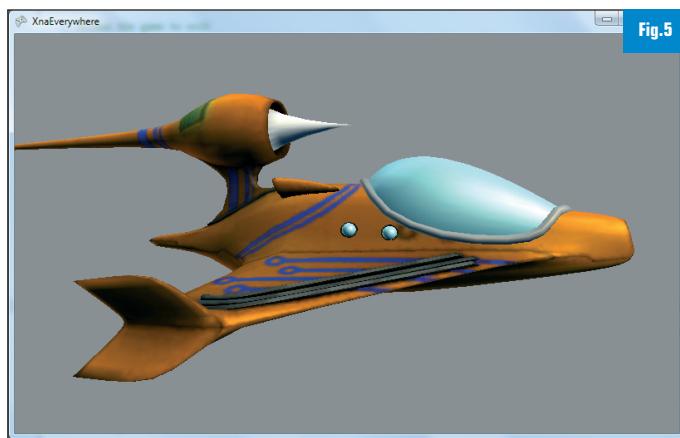
Pour cela, il suffit de faire un clic droit sur notre projet principal et de sélectionner le menu « Create Copy of Project for Xbox 360 » : [Fig.6]. Ce qui fait donc apparaître un nouveau projet avec une jolie icône Xbox 360 : [Fig.7].

A noter qu'il n'est pas nécessaire de faire un portage du projet de contenus puisque ce dernier n'est utile que lors de la compilation. Ainsi sans changer une ligne de code, nous disposons d'une version Xbox



360 de notre application qui pourra par la suite être déployée sur la console pour tester et débugger puis qui pourra finalement être mise sur le App Hub de la Xbox 360.

Un autre point est important à signaler : il n'y a pas de copie des fichiers entre les projets. Ainsi par exemple le fichier Game1.cs n'existe qu'une fois et est référencé par les deux projets (Windows



et Xbox 360. Si vous le modifiez, vous le modifiez pour les deux environnements. C'est pourquoi il est utile de faire appel aux directives de compilations qui permettent de garder un seul fichier mais avec certaines parties uniquement visibles pour un projet :

```
#if WINDOWS || XBOX
    static class Program
    {
        static void Main(string[] args)
        {
            using (Game1 game = new Game1())
            {
                game.Run();
            }
        }
    }
#endif
```

Ici la classe statique Program n'existera que pour Windows et la Xbox. De ce fait, quand nous ajouterons un projet Windows Phone 7, ce dernier ne verra pas cette classe car dans son cas elle n'est pas nécessaire.

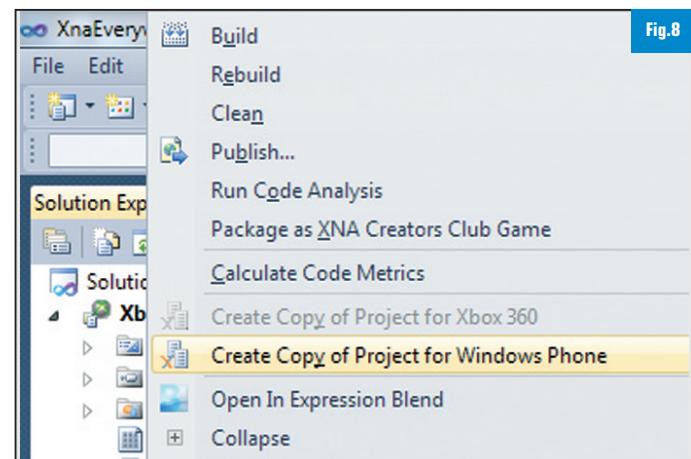
XNA sur Windows Phone 7

Le portage vers Windows Phone 7 va s'avérer également très simple puisque de la même manière que pour la Xbox 360, il suffit d'appeler un menu spécifique : [Fig.8].

Ce qui fait apparaître un nouveau projet dans notre solution : [Fig.9].

Ce projet est immédiatement utilisable sur l'émulateur : [Fig.10].

Grâce aux directives de compilations, nous pourrons spécifier des comportements différents sur Windows Phone 7 si nous le souhaitons.



tons pour, par exemple, nous adapter à la moindre puissance du matériel (changer les shaders, réduire la complexité des objets, etc.)

XNA sur Silverlight pour Windows Phone 7.1

La nouvelle version de Windows Phone (nommée 7.1 pour le moment) rend possible de faire du XNA non pas uniquement en mode exclusif mais plutôt hébergé dans une application Silverlight. Ainsi au sein d'une application Silverlight standard, nous allons pouvoir avoir des pages .xaml dont le rendu sera pris en charge par XNA et non plus par le moteur de dessin de Silverlight.

Cela signifie que nous allons pouvoir profiter à la fois de la puissance de rendu 3D de XNA tout en nous servant des éléments d'interfaces et d'animations de Silverlight. Pour cela, il faut, au niveau d'une page .xaml modifier le code behind pour basculer le moteur de rendu :

```
protected override void OnNavigatedTo(NavigationEventArgs e)
{
    SharedGraphicsDeviceManager.Current.GraphicsDevice.SetSharingMode(true);

    model = content.Load<Model>("Ship");

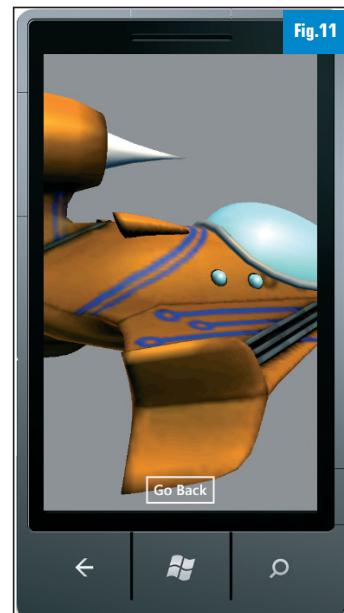
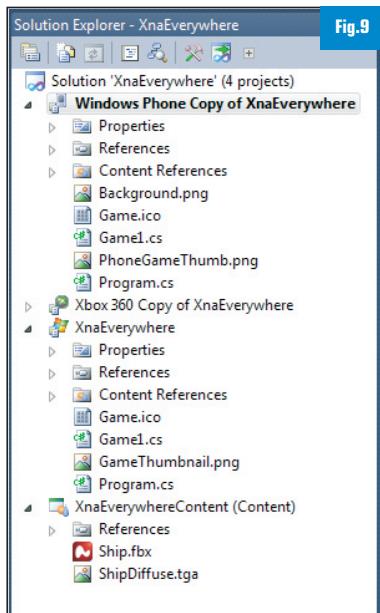
    spriteBatch = new SpriteBatch(SharedGraphicsDeviceManager.Current.GraphicsDevice);

    // Start the timer
    timer.Start();

    base.OnNavigatedTo(e);
}
```

Comme on peut le constater, lors de l'événement OnNavigatedTo, il faut faire appel à la méthode SetSharingMode de l'objet SharedGraphicsDeviceManager.Current.GraphicsDevice pour indiquer que l'on passe en mode « partagé ».

Lors de la construction de la page, nous en profiterons pour récupérer notre ContentManager qui est référencé par le projet Silverlight :



```
public GamePage()
{
    InitializeComponent();

    // On récupère le content manager general de l'application
    content = (Application.Current as App).Content;

    // Création du timer pour la page
    timer = new GameTimer();
    timer.UpdateInterval = TimeSpan.FromTicks(333333);
    timer.Update += OnUpdate;
    timer.Draw += OnDraw;

    LayoutUpdated += GamePage_LayoutUpdated;
}
```

Lors de la construction, nous déclarons également un GameTimer qui jouera le rôle de séquenceur comme notre classe Game en XNA pur. Il se chargera d'appeler de manière cohérente nos méthodes OnUpdate et OnDraw.

Lors de la mise à jour du layout (lorsque le XAML a été traité et a généré son arbre visuel), il faut instancier un objet particulier qui se chargera de faire le dessin de ce layout pour XNA :

```
void GamePage_LayoutUpdated(object sender, EventArgs e)
{
    if (ActualWidth > 0 && ActualHeight > 0 && elementRenderer == null)
    {
        elementRenderer = new UIElementRenderer(this, (int)ActualWidth, (int)ActualHeight);
    }
}
```

OnUpdate se chargera toujours de faire progresser notre angle. OnDraw quant à elle doit toujours faire le rendu XNA mais en plus, elle doit demander le dessin du XAML dans une texture pour pouvoir la dessiner par-dessus la 3D :

```
private void OnDraw(object sender, GameTimerEventArgs e)
{
    // Dessin du xaml dans une texture
    elementRenderer.Render();

    // 3D
    GraphicsDevice device = SharedGraphicsDeviceManager.Current.GraphicsDevice;

    device.Clear(Color.Gray);

    device.BlendState = BlendState.Opaque;
    device.DepthStencilState = DepthStencilState.Default;
    device.SamplerStates[0] = SamplerState.LinearWrap;

    foreach (ModelMesh mesh in model.Meshes)
    {
        foreach (BasicEffect effect in mesh.Effects)
        {
            effect.World = Matrix.CreateRotationY(angle);
        }
    }
}
```

```

        effect.View = Matrix.CreateLookAt(new Vector3(0,
0, 2000), Vector3.Zero, Vector3.Up);
        effect.Projection = Matrix.CreatePerspectiveFieldOfView(0.8f, device.Viewport.AspectRatio, 0.1f, 10000.0f);
        effect.EnableDefaultLighting();
    }

    mesh.Draw();

}

// Dessin de la texture issue du xaml
spriteBatch.Begin();
spriteBatch.Draw(elementRenderer.Texture, Vector2.Zero, Color.White);
spriteBatch.End();

}

```

Grâce à ce fonctionnement, nous obtenons notre interface Silverlight au-dessus de notre rendu (libre à nous par la suite de faire plus audacieux en utilisant la texture XAML sur nos objets par exemple) : [Fig.11].

XNA sur Silverlight 5

Pour conclure notre visite au pays de XNA, il est obligatoire de parler de la dernière version de Silverlight. Cette version numérotée « 5 » intègre entre autres nouveautés le support de XNA. Ce support est différent de celui fourni par Windows Phone puisqu'ici Silverlight 5 fournit un contrôle nommé « DrawingSurface » qui va lever un événement OnDraw dans lequel nous pouvons faire du XNA sur la surface du contrôle.

Par ce biais, le rendu XNA est totalement intégré à l'interface et ce de manière simplissime :

```
<Controls:DrawingSurface Name=>drawingSurface Draw=>DrawingSurface_Draw SizeChanged=>drawingSurface_SizeChanged/>
```

En effet, lors de l'événement OnDraw, nous allons exécuter le même code que si nous étions dans un projet pur XNA :

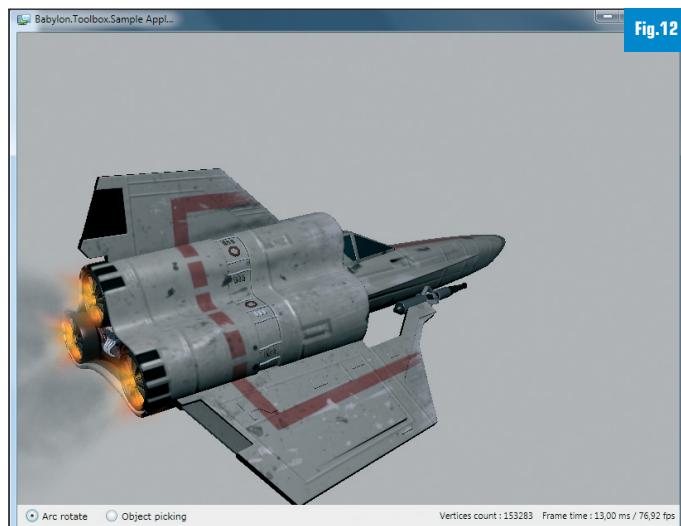


Fig.12

```

private void DrawingSurface_Draw(object sender, DrawEventArgs e)
{
    e.GraphicsDevice.BlendState = BlendState.AlphaBlend;

    // Draw
    e.GraphicsDevice.Clear(ClearOptions.Target | ClearOptions.DepthBuffer, new Color(0, 0, 0, 0), 1.0f, 0);

    foreach (ModelMesh mesh in model.Meshes)
    {
        foreach (BasicEffect basicEffect in mesh.Effects)
        {
            basicEffect.SceneAmbientColor = new Color(0.3f, 0.3f, 0.3f, 0);

            basicEffect.World = Matrix.Identity;
            basicEffect.View = camera.View;
            basicEffect.Projection = camera.Projection;
            basicEffect.LightPosition = camera.Position + new Vector3(1, 5, 3);
        }
    }

    model.Draw();

    e.InvalidateSurface();
}

```

On peut constater que l'événement OnDraw nous fait passer un argument de type DrawEventArgs dans lequel on va trouver une référence vers notre GraphicsDevice.

Grâce à lui nous pourrons donner nos ordres de rendus. Il ne faudra pas oublier d'appeler e.InvalidateSurface() à la fin pour redemander le dessin d'un image sinon Silverlight ne rappellera pas l'événement OnDraw [Fig.12].

Le problème pour le moment est que XNA pour Silverlight 5 n'intègre pas plusieurs points importants de XNA :

- Pas de ContentManager
- Pas de classe Effect
- Pas de classe Model

En attendant, un projet sur Codeplex permet de remplacer les services manquants (Effect, Model, chargement depuis des formats comme le .OBJ) : <http://babylontoolkit.codeplex.com>. A vous les joies de la 3D dans les navigateurs !

Conclusion

Tout au long de cet article, nous avons pu voir comment XNA permet de faire de la 3D facilement sur de nombreux environnements Microsoft. Hormis Silverlight 5 qui est encore en bêta et ne dispose pas de toutes les options, nous avons pu constater comment le portage peut s'avérer simple entre les environnements.

Si vous désirez aller plus loin ou simplement télécharger les outils de développement, une seule adresse importe : <http://create.msdn.com>.

■ David Catuhe / Microsoft

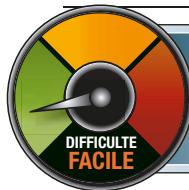
Relations techniques avec les développeurs

Blog : <http://blogs.msdn.com/b/eternalcoding>

Mail : davca@microsoft.com

LISP : Hors du temps

Il est un des tous premiers langages et il ne laisse personne indifférent. On le déteste ou on l'adore. Il traverse les âges et les modes. C'est Lisp.



APPLICATION : Tout

LANGAGE : Fonctionnel

SOURCE : Non

Les deux premiers langages informatiques sont, c'est bien connu, Fortran et Lisp. Ils furent si différents que cela a créé une sorte de fracture dans le monde de la programmation, fracture qui ne s'est jamais vraiment refermée. Lisp est le tout premier langage fonctionnel. Inventé dans les années 1950 par McCarthy au MIT, c'était alors un langage fonctionnel pur. Il était très en avance sur son temps, et d'une certaine façon, l'est toujours aujourd'hui. Très vite, il s'est taillé la réputation du langage de l'intelligence artificielle.

Ceci, non parce qu'il était conçu pour cela spécifiquement, mais parce qu'il était un langage procurant un haut niveau d'abstraction, et comme tel, permettant d'aborder l'écriture de programmes d'intelligence artificielle.

EN AVANCE SUR SON TEMPS

Lisp a tout inventé avant tout le monde. Ainsi McCarthy qui a écrit son premier interpréteur Lisp en Fortran (il n'avait que cela sous la main) s'est vite rendu compte des faiblesses de ce langage (nous parlons du Fortran de l'époque. Fortran a lui-même beaucoup évolué) et il a inventé la construction if-then-else qui faisait défaut. McCarthy a fait de Lisp le premier langage doté d'un ramaressement pour libérer le programmeur de la gestion de la mémoire. McCarthy a créé les premiers systèmes 'time-shared' ou système à temps d'exécution partagé, ce qui a abouti à la possibilité de charger du code à chaud.

Principe repris par un autre langage fonctionnel génial: Erlang. Lisp c'est le premier langage typé dynamiquement, le premier langage à permettre la métaprogrammation, le premier langage capable de faire des calculs sur des entiers sans limitation de grandeurs et de gérer des tableaux redimensionnables dynamiquement. Le premier, Lisp traite les fonctions comme un type de données et supporte la récursivité.

Enfin le premier bootstrap de l'histoire a été fait en Lisp. En effet, John McCarthy a écrit son premier compilateur Lisp en Lisp. Lisp est donc le premier langage capable de se compiler lui-même. Lisp a inventé l'interpréteur interactif, et un interpréteur interactif en Lisp, ça s'écrit en une ligne de code Lisp :

```
(loop (print (eval (read))))
```

TOUJOURS EN AVANCE

Aujourd'hui Lisp a beaucoup évolué. Il n'est plus du tout un langage fonctionnel pur. Il est tout : impératif, objet, fonctionnel... D'autres langages le sont aussi. Mais Lisp a encore beaucoup plus, et on pourrait le résumer en une phrase de John Foderaro : Lisp est un *langage de programmation programmable*. Ainsi quand le paradigme objet est apparu, Lisp n'a pas été modifié dans son corps pour autant. Son orientation objet a été écrite en Lisp.

A LA BASE UNE IDÉE TOUTE SIMPLE

Dit succinctement, pour Lisp, tout ce qui n'est pas un atome est une liste. Une liste a une tête, le premier élément de la liste, ou *car*, et une queue, ou *cdr* qui est le reste de la liste. Quand le premier élément d'une liste est une fonction ou un opérateur, il est appliqué au reste de la liste. Ceci implique que Lisp utilise la notation préfixée. Par exemple

```
(+ 1 2 3)
```

fait la somme des nombres un, deux et trois, et s'évalue à 6. Cette facilité est bien entendu impossible en notation infixe. Avec ce qui vient d'être dit, vous savez l'essentiel et vous pouvez commencer à coder en Lisp ! De ce qui vient d'être dit, on déduit qu'un programme Lisp présente un nombre très élevé de parenthèses. Beaucoup trop élevé disent ses détracteurs qui souvent le surnomment *Lot of Insipid and Stupid Parenthesis* :) Il est certain que la syntaxe de Lisp, qui rend extrêmement simple l'analyse syntaxique pour le système, a de quoi désarçonner le programmeur habitué à autre chose. Cependant, c'est un faux procès. On s'y fait très vite, surtout lorsqu'on suit les conventions d'écriture et qu'on utilise un éditeur dédié ou mieux encore, un IDE complet comme Allegro CL ou Lispworks. Rapidement ces parenthèses, on ne les voit même plus. Un peu comme les nombreuses accolades de C, Java, PHP, ne sont pas un problème avec un éditeur qui va bien, et souvent un problème sans lui. Aventurons-nous à regarder un peu de code Lisp. L'exemple ci-dessous est le jeu ultra-connu et absolument sans intérêt C+/C- L'ordinateur tire au sort un nombre entre 1 et 100, et le joueur doit essayer de deviner ce nombre avec le moins d'essais possibles.

```
;;; Une implémentation du jeu C+/C- en LIsp

(defun GénérerNombre()
  (random 100))

(defun CplusCmoins()
  (let ((nombre (GénérerNombre)))
    (format t "~-~%Jouons ensemble à C+/C-!~%")
    (format t "J'ai choisi un nombre entre 0 et 99~%")
    (format t "Essayez de le deviner, ou tapez 'q' pour quitter
la partie à n'importe quel moment.")
    (Jouer nombre 1)))

(defun Jouer(nombre tentatives)
  (let ((essai (read)))
    (cond
      ((or (eq essai 'q) (eq essai 'Q))
       (format t "~-~%Vous avez quitté la partie.~%")
       (format t "Merci d'avoir joué avec moi.~%"))
      (t
       (format t "~-~%Tentez de deviner le nombre.~%")
       (format t "Le nombre est : ~d~%" nombre)
       (Jouer nombre (1- tentatives))))
```

```

((> essai nombre)
 (format t "Trop grand. Essayez à nouveau.~%")
 (Jouer nombre (+ 1 tentatives)))
((< essai nombre)
 (format t "Trop petit. Essayez à nouveau.")
 (Jouer nombre (+ 1 tentatives)))
((= essai nombre)
 (format t "Félicitations!~%")
 (format t "Vous avez deviné le nombre en ~d tentatives~%"
 tentatives)
 (JouerEncore)))))

(defun JouerEncore()
 (format t "Voulez vous rejouer (o/n)")
 (let ((reponse (read)))
 (cond
  ((or (eq reponse '0) (eq reponse 'o))
   (CplusCmoins))
  (t
   (format t "Merci d'avoir joué avec moi.~%")))))

```

Il est certain que si, sous l'effet de surprise, on regarde ce code en essayant de suivre les parenthèses, on va vite attraper mal à la tête. Mais si on ne les regarde pas, tout devient très facile. Ainsi la première fonction du code se lit :

```

defun GernerNombre
  random 100

```

L'on comprend tout de suite que nous avons une fonction qui retourne un nombre aléatoire. Ensuite :

```

let nombre GernerNombre
  format t "~%~%Jouons ensemble à C+/C-!~%"
  format t "J'ai choisi un nombre entre 0 et 99~%"

```

Nous déclarons une variable affectée du nombre aléatoire généré par notre fonction, puis nous imprimons des messages sur la console. Ensuite :

```

let essai read
cond
  or eq essai 'q eq essai '0

```

Nous lisons une fonction sur la console et nous affectons une variable essai. Puis nous entrons dans une suite de tests, qui commencent par regarder si essai vaut q ou Q. Avec un bon éditeur, qui va placer les parenthèses et indenter automatiquement, lire du Lisp n'est finalement pas plus difficile que de lire autre chose. Des déclarations de templates C++ par exemple ;-)

UN LANGAGE QUI RESTE UTILE

Puisque nous en sommes aux reproches, les détracteurs se demandent à son sujet: à quoi ça sert ? Il ne sert pas à tout, notamment parce qu'un environnement Lisp est assez volumineux. Mais il servira dans le cas d'applications très complexes. Ainsi c'est en Lisp qu'a par exemple été écrite l'application *Recife* qui sert d'aide à la décision du routage des communications chez France Telecom. Une des raisons du choix de Lisp pour cette application a été qu'il permet le chargement de code à chaud, opportunité qui n'est pas si courante. C'est en Lisp que Paul Graham a écrit Viaweb, le premier fournis-

seur d'applications, revendu plus tard à Yahoo pour près de 50 millions de dollars. Paul Graham est aussi l'auteur d'un ouvrage très célèbre: *On Lisp*. Dans cet ouvrage, Paul explique que Lisp est excellent pour le développement en approche montante (bottom-up) une approche où l'on étend le langage de base avec des concepts facilitant l'expression du problème. Contrairement à l'approche descendante (top-down), plus classiquement employée, et dans laquelle le problème à résoudre est décomposé jusqu'à pouvoir être exprimé dans les termes du langage de base. Autrement dit, en programmation courante on plie le problème pour faire avec le langage, alors qu'en bottom-up on plie le langage pour faire avec le problème. Lisp n'est pas le seul langage à permettre le bottom-up, c'est évident, mais il est reconnu comme le plus adapté à cela, le plus souple. Ce qui fait dire à Robert Strandh, un guru français de Lisp: "*Il est courant dans l'industrie qu'on choisisse un langage connu par les programmeurs (afin d'éviter le coût de l'apprentissage), mais si peu adapté à la tâche qu'on est finalement perdant.*" Lisp est aussi remarquable comme langage d'extension. Là non plus il n'est pas le seul. Mais il va bien. Deux exemples archi-connus : Emacs et AutoCAD. Enfin si Lisp a traîné pendant des années une réputation de lenteur, ceci est oublié depuis bien longtemps. Malgré son typage dynamique, les meilleures implémentations présentent des performances proches de C. Lisp est encore couramment utilisé dans de grosses applications bancaires. En outre Lisp est standardisé ANSI et il est le seul, à ma connaissance, à avoir un GUI également standardisé.

ATTENTION...

Ceux qui apprécient Lisp l'apprécient vraiment. Ainsi Gregory Chaitin dit de lui : « *Malheureusement, alors que les langages de programmation gagnent en sophistication, ils sont de plus en plus le reflet de la complexité de la société humaine et du monde immense des applications logicielles. Ainsi, ils deviennent d'énormes boîtes à outils, comme des garages et des greniers chargés de plus de trente ans d'histoire ! A contrario, Lisp est un langage de programmation d'une grande beauté mathématique ; il ressemble plus à un scalpel de chirurgien ou à un diamant affûté qu'à un garage à deux places encombré de bricolages, où il ne reste plus de place pour une voiture.*

Lisp a un petit nombre de concepts élémentaires puissants, et tout le reste est construit au-dessus de ça, ce qui correspond à la façon de travailler des mathématiciens ; c'est à ça que ressemblent les théories mathématiques. Ces théories, les bonnes théories, consistent à définir quelques nouveaux concepts clés, et à partir de là le feu d'artifice commence : elles révèlent de nouvelles allées, elles ouvrent la porte à des mondes radicalement nouveaux. Lisp est comme ça aussi ; il est plus proche des maths que la plupart des langages de programmation. Du moins si vous éliminez les parties utiles qui ont été ajoutées, les ajouts qui ont fait de Lisp un outil pratique. Ce qui reste si vous faites cela, c'est le Lisp original, le cœur conceptuel de Lisp, un cœur qui est un joyau de beauté mathématique et de beauté intellectuelle austère. » Vous pouvez découvrir Lisp avec les éditions personnelles de Allegro (www.franz.com), Lispworks (www.lispworks.com) Corman Lisp (www.cormanlisp.com) ou encore la remarquable implémentation open source Steel Bank Common Lisp (SBCL www.sbcl.org) Mais attention! On devient très vite totalement accro... :-)

■ Frédéric Mazué - fmazue@programmez.com

Haskell : élégance, pureté, beauté

Si la programmation fonctionnelle est peu populaire en France, le langage Haskell y est lui, totalement inconnu, ou quasiment. Pourtant que de qualités dans ce langage très en avance sur son temps ! Découvrons-le.



APPLICATION : Tout

LANGAGE : Fonctionnel

SOURCE : Non

Tous les langages de programmation ont pour vocation de procurer un niveau d'abstraction "facilitant" la programmation et affranchissant le programmeur des détails les plus techniques comme la taille des cases mémoire ou les instructions CPU. Certains n'y parviennent pas très bien. Ainsi la syntaxe d'un Fortran, langage qui reste très populaire, est en liaison directe avec la structure physique des cartes perforées de la préhistoire. D'autres y parviennent sans y parvenir, comme C++ avec son haut niveau d'abstraction et ses constants problèmes dans le bas niveau. D'autres encore y parviennent assez bien, comme Java avec son ramasse-miettes et sa machine virtuelle. Mais toujours quelque part, ces langages font partie du problème de la programmation. Les langages fonctionnels ont, parmi d'autres bonnes intentions, celle d'abstraire complètement de la machine. Avec eux, un programme est un ensemble de fonctions au sens mathématique du terme et c'est le système qui s'occupe de l'évaluation d'un ensemble d'expressions, pas le programmeur. Pour ce faire, ces langages s'appuient sur un runtime très évolué. C'est ainsi que Lisp, le premier des langages fonctionnels, et créé il y plus de cinquante ans, a inventé le ramasse-miettes que Java a repris à son compte bien plus tard, à grand renfort de communication...

Haskell, un langage à la pointe du progrès

Dans la lignée des langages fonctionnels, Haskell est encore tout récent. Ses spécifications ont été fixées en 1998. Celles-ci ne sont toutefois pas un standard hélas. Haskell est un langage à la pointe du progrès, doté d'une syntaxe épurée, fonctionnel pur (sans aucun effet de bord), totalement paresseux et doté à la fois d'un polymorphisme ad hoc et d'un polymorphisme paramétrique. Voilà pour la définition technique. Nous n'entrerons pas dans ces détails abscons. Disons plus simplement que tout avec Haskell est fonction, y compris les entrées/sorties qui sont effectuées dans un contexte purement fonctionnel, la Monade. Haskell c'est l'homogénéité. Les constructeurs de types sont fonctionnels, les types intégrés et les types utilisateurs se comportent de la même façon, etc.

Cette homogénéité et cette absence d'effet de bord apportent un niveau d'abstraction incroyable ainsi que nous le verrons plus loin. Encore ignoré en France, pays où l'on a toujours un train de retard, Haskell est de plus en plus utilisé à l'étranger, notamment aux Etats-Unis, au Canada et en Pologne, par des industriels qui ont compris les avantages qu'ils pouvaient en tirer en termes de temps de développement et de sûreté du code.

C'est que, comme avec beaucoup de langages fonctionnels, la correction d'un programme Haskell peut être prouvée. En comparaison, les tests en C, Java, C++ ne font au mieux que mettre des bugs en évidence, mais hélas ne prouvent pas leur absence... Ainsi,

comme nous l'avons dit plus haut, les défauts inhérents à tous les langages impératifs font du langage lui-même une partie du problème à résoudre plutôt qu'une partie de la solution. Le reste de cet article, tout en présentant Haskell, va essayer de montrer que celui-ci ne fait pas partie du problème de la programmation, mais est réellement un moyen d'en exprimer la solution, avec concision qui plus est, et sans même devoir réfléchir au déroulement du programme.

Haskell n'est pas un langage à tout faire

Hélas, en ce monde nul n'est parfait ! Doté de librairies standard riches: réseau, graphiques, OpenGL, concurrence, etc., Haskell n'a toutefois pas la prétention de savoir tout faire. S'il est question de piloter un périphérique c'est évidemment C (ou C++ ou Pascal ou Ada, etc.) que l'on retiendra puisque c'est sa vocation. Par contre, s'il est question d'écrire une application scientifique, bancaire, réseau, exemples non limitatifs, Haskell convient. On peut également trouver des défauts à Haskell. Ses performances excellentes sont en dessous de C en raison notamment du mécanisme d'évaluation paresseuse. Il a tout ce qu'il faut pour supporter Unicode mais les librairies sont encore en cours d'écriture.

Le débogage peut s'avérer difficile en raison de l'évaluation paresseuse. Mais Haskell est tellement sûr, qu'un programmeur n'éprouve le besoin de déboguer que très exceptionnellement. Le problème majeur est l'apprentissage. Ce n'est pas que Haskell soit difficile à apprendre ou à manipuler par lui-même.

La difficulté réside dans les habitudes prises par les programmeurs rompus aux langages impératifs. Aborder un langage qui ne connaît pas l'affectation comme Haskell à la manière d'un langage impératif, c'est se heurter à un mur. Il est nécessaire de l'aborder avec l'esprit neuf et de prendre le temps d'en saisir la philosophie. J'ai commencé ma carrière en programmant des automates et de l'assembleur. Alors ces difficultés je les connais bien :)

Elles ne sont finalement pas si difficiles à surmonter. Et avec le recul, souvent le résultat laisse songeur. Ainsi, si après avoir pratiqué Haskell quelque temps, vous êtes amenés à revenir à C, C++, Java, ou encore d'autres, vous vous demandez immanquablement comment vous avez pu, par le passé, travailler avec ces tas de silex. Je vous propose maintenant de voir ce que abstraction veut dire avec Haskell.

Un peu de code, beaucoup d'abstraction

Pour bien évaluer un langage, rien de tel que d'écrire un peu de code et comparer avec le code équivalent écrit avec d'autres langages. Posons un problème simple. Nous voulons une fonction qui recevant une liste (ou équivalent selon le langage), construise et retourne une liste telle que ses éléments soient ceux de la liste reçue en argument, additionnés à une valeur également reçue en argument.

L'idéal est que la fonction soit générique. Nous choisissons de travailler d'abord C++ et nous écrivons d'abord un premier jet, qui travaillera sur des entiers :

```
#include <vector>
#include <algorithm>
#include <functional>

std::vector<int> f(const std::vector<int>& v, int valeur)
{
    std::vector<int> nv(v.size());
    std::transform(v.begin(), v.end(),
                  nv.begin(),
                  std::bind1st(std::plus<int>(), valeur));
    return nv;
}
```

Nous avons 10 lignes de code pas vraiment faciles à lire. Ne parlons pas de concision. Voici maintenant la fonction écrite en Haskell :

```
f valeur liste = map (+ valeur) liste
```

Une seule ligne de code. Et encore, fort verbeuse... :-) Car un haskeller averti, appliquera la règle du currying et écrira seulement :

```
f valeur = map (+ valeur)
```

Nous pouvons maintenant parler de concision :-) Le code C++ fait-il au moins ce qu'on lui demande ? Rien n'est moins sûr. Nous y reviendrons, mais écrivons d'abord des fonctions génériques, pour travailler sur tout type supportant l'opération d'addition. En C++ :

```
#include <vector>
#include <algorithm>
#include <functional>

template <class T> std::vector<T> ft(const std::vector<T>& v, T valeur)
{
    std::vector<T> nv(v.size());
    std::transform(v.begin(), v.end(),
                  nv.begin(),
                  std::bind1st(std::plus<T>(), valeur));
    return nv;
}
```

Le code C++ n'a pas gagné en clarté. En Haskell maintenant :

```
f valeur = map (+ valeur)
```

Et oui, c'est exactement le même que plus haut. Notre ligne de code, toute simple, était déjà générique de facto. Essayons un peu notre code. En C++ :

```
int data[] = {1, 2, 3};
std::vector<int> v(data, data+3);

std::vector<int> nv = f(v, 10.5);
std::vector<int> nvt = ft<int>(v, 10.5);
```

A l'issue de quoi les vector nv et nvt contiendront tous deux les valeurs 11, 12 et 13. Seulement voilà 1+10.5 cela ne fait pas 11. Que s'est il passé ? Nous avons dit à C++ que nous travaillons avec des int. Or nous lui passons une valeur de type double (10.5). Qu'à cela ne tienne, tout générique qu'il est, C++ fait entrer "au chausse-pied" la valeur dans le type int. Peut être le compilateur émettra-t-il un avertissement selon les options de compilation sélectionnées, mais rien, dans le standard C++, ne l'oblige à le faire. Peut-être que le résultat de l'opération est bien conforme à ce que nous souhaitons, ou peut-être avons nous simplement perdu des données à la suite d'une erreur de programmation. Ce genre d'erreur d'inattention se produit fréquemment et C++ valide sans état d'âme, et ni l'un ni l'autre de nos exemples de codes C++ ne se comporte correctement. Bon admettons-le, nous avons triché. Nous disons au compilateur que nous travaillons avec des valeurs entières, travaillons donc effectivement avec des valeurs entières :

```
unsigned int data[] = {1, 2, 3};
std::vector<unsigned int> v(data, data+3);

std::vector<unsigned int> nvt = ft<unsigned int>(v, 4294967294U);
```

Cette fois, nvt contient les valeurs 4294967295, 0 et 1. En raison d'un débordement de capacité d'un registre 32 bits, notre calcul est encore faux! Les programmeurs expérimentés que vous êtes savent bien que ce type d'erreur est hélas lui aussi très fréquent. En raison de son manque de sûreté, nous concluons que C++ fait partie de notre petit problème, pas de sa solution. Voyons avec Haskell :

```
f 10.5 [1, 2, 3]
[11.5, 12.5, 13.5]
```

Nous n'avons rien dit à Haskell quant au type. De lui-même il infère le bon type, le type Float. Nous n'avons pas perdu de valeur. Mais peut-être voulions nous travailler avec des Int. Voyons comment cela fonctionne: notre fonction est générique. Voici sa signature :

```
f :: (Num a) => a -> [a] -> [a]
f valeur = map (+ valeur)
```

Cela signifie que f fonctionne pour tout type de la classe Num(érique), c'est-à-dire ici et concrètement tout type, y compris un type utilisateur, supportant l'opération d'addition. Mais nous pouvons souhaiter spécialiser la fonction à des valeurs entières :

```
f :: (Integral a) => a -> [a] -> [a]
f valeur = map (+ valeur)
```

A partir de maintenant, l'appel f 10.5 [1, 2, 3] sera rejeté par le compilateur. Et si nous donnons :

```
f 4294967294 [1, 2, 3]
```

nous obtenons le résultat correct :

```
[4294967295, 4294967296, 4294967297]
```

Nous constatons qu'en plus d'être concis, Haskell dispose d'un système de types extrêmement sûr. En béton armé même :-) Nous concluons que Haskell fait, lui, partie de la solution à notre petit pro-

blème :-) Quand nous écrivons notre fonction `f` en Haskell, nous n'avons à nous soucier ni du déroulement de son code ni d'éventuelles pertes de valeurs. Nous sommes affranchis des tracas inhérents à la machine, à la taille des types, etc.

TOUJOURS PLUS D'ABSTRACTION

Ecrivons maintenant des fonctions factorielles en Java. Une fonction est récursive et l'autre non. Pour ne pas alourdir le discours, nous admettons que les fonctions seront appelées avec un entier strictement positif en argument.

```
public class Facs {
    public static int facRec(int valeur) {
        if(valeur==1)
            return 1;
        else
            return valeur*facRec(valeur-1);
    }
    public static int fac (int valeur) {
        int result = 1;
        for(int i=2; i<= valeur; i++)
            result *= i;
        return result;
    }
}
```

Ce code verbeux fonctionne-t-il ? Nullement. L'appel `facRec(15000)` provoque une exception `StackOverflowError`. L'appel `fac(15000)` retourne la valeur zéro, ce qui ne semble pas très exact. Fondamentalement nous avons des erreurs dans l'esprit de celles rencontrées plus haut avec C++. La tracasserie du débordement de pile est due à la machine, que celle-ci soit affublée du nom de virtuelle n'y change rien. Quant à la valeur fausse, c'est le même débordement de capacité que celui rencontré précédemment. Comparons maintenant notre code Java avec Haskell d'un autre point de vue. Dans notre code Java, nous sommes extrêmement dirigistes. En plus d'exprimer l'algorithme proprement dit, nous avons codé la séquence d'exécution dans le détail. Voyons comment cela peut se passer avec Haskell :

```
facrec valeur =
  if valeur == 1 then 1
  else valeur* facrec (valeur-1)
```

Ce code, extrêmement maladroit, a au moins le mérite de fonctionner et est capable de retourner le résultat de `facrec 15 000`, nombre astronomique certes, mais calculé. Pourquoi ce code est-il maladroit ? Parce qu'il est écrit dans un style impératif, comme si nous voulions avoir la main mise sur le flux d'exécution, ce qui est contraire à la philosophie du langage. Haskell est un langage déclaratif. Déclarons donc que dans le cas où valeur vaut 1, notre fonction s'évalue à 1 et que sinon elle s'évalue à valeur multipliée à elle-même invoquée avec `valeur-1` en argument :

```
facrec 1 = 1
facrec valeur = valeur * facrec (valeur-1)
```

Plus de `if` ni de `else`. Nous nous sommes concentrés uniquement sur la définition de la fonction et nous laissons au compilateur le soin

de construire la séquence d'exécution. Il y a toutefois bien mieux à faire. Quand, humainement, on pense "factorielle n", on pense au produit de (la liste de) tous les entiers de 1 à n. Tel on le pense, tel on l'écrit en Haskell :

```
fac n = product [1..n]
```

Nous avons là un grand moment d'informatique. D'abord la concession. Ensuite, aucune contrainte due à la machine, aucune gestion de la séquence d'exécution, rien. Nous avons simplement exprimé la chose telle qu'elle est, en une ligne, et c'est réglé. Techniquement cela est possible grâce à l'évaluation paresseuse de Haskell, qui ne construira jamais la liste des entiers de un à n mais évaluera ceux-ci au fur et à mesure des besoins. A nouveau, le langage fait partie de la solution au lieu de faire partie du problème. Il est aussi, accessoirement, une solution au problème des barrettes de Ram ;)

INFINIMENT PLUS D'ABSTRACTION

Haskell permet d'aller infiniment plus loin, c'est le cas de le dire, que ce que nous venons de voir avec notre fonction `fac`. En effet, grâce à l'évaluation paresseuse, nous pouvons manipuler des structures de données infinies. Pour illustrer cela, posons-nous un autre petit problème. Nous voulons travailler avec une liste d'entiers tels que ceux-ci soient tous les carrés d'entiers impairs. Cette liste est donc infinie. Nous voulons être capables d'afficher par exemple les 5 premiers éléments de cette liste, nous voulons être capables de tester si une valeur quelconque fait partie de cette liste et nous voulons pouvoir afficher par exemple le 1000e élément de cette liste. Voici le code, simple, qui fait tout cela :

```
ensembleN = [0,1...]
impairs = filter odd ensembleN
impairs_carres = [x*x | x <- impairs]
est_impair_carre n (x:xs)
  | n < x = False
  | n == x = True
  | otherwise = est_impair_carre n xs
main = do
  putStrLn $ (show $ take 5 impairs_carres) ++ "\n"
  putStrLn $ (show $ est_impair_carre 121 impairs_carres) ++ "\n"
  putStrLn $ (show $ est_impair_carre 83 impairs_carres) ++ "\n"
  putStrLn $ (show $ impairs_carres !! 999) ++ "\n"
```

A l'exécution (fonction `main`) ce code affiche :

```
[1,9,25,49,81]
False
True
3996001
```

Ce code est fascinant. La première ligne déclare rien moins que l'ensemble des entiers naturels. D'ailleurs la déclaration `ensembleN = [0..]` aurait suffit :-) Toujours est-il que `ensembleN` est un ensemble de données infini. Pourtant, nous pouvons passer `ensembleN` en

argument à une autre fonction comme si de rien n'était. C'est d'ailleurs ce que nous faisons dans la fonction "impairs" qui filtre, au moyen du prédictat odd tous les entiers impairs. "impairs" est donc l'ensemble, infini lui aussi, des entiers impairs. Enfin nous déclarons un troisième ensemble, lui aussi infini, celui des carrés des nombres impairs. Pour ce faire nous utilisons une liste compréhensive, qui construit la liste de tous les carrés des valeurs prises dans l'ensemble des valeurs impaires. Nous manipulons ainsi, l'air de rien, pas moins de trois ensembles infinis. Enfin nous déclarons une fonction "est_impairs_carres" qui teste si un nombre passé en argument appartient à la liste impairs_carres également passée en argument. Nous procédons en comparant le nombre à la tête de la liste. Pour cette fonction, nous déclarons simplement que si le nombre est inférieur à la tête de la liste, il n'appartient pas à cette liste. Nous déclarons que si le nombre est égal à la tête de la liste, il appartient à la liste, et dans tous les autres cas nous déclarons devoir tester notre nombre sur la queue (ou reste) de la liste.

UN EXEMPLE DE ROT13

Toujours au chapitre des types récursif/ infinis, et toujours dans l'intention d'illustrer combien il est aisé de coder en Haskell, voici un programme de ROT13, qui se base sur une déclaration originale de l'alphabet: un alphabet récursif :

```
alphabet = ['A'..'Z'] ++ alphabet
```

Nous avons une fonction récursive non terminante qui s'évalue sous la forme d'une liste qui se concatène à elle-même autant que de besoin, mais pas plus, toujours en vertu de mécanisme d'évaluation paresseuse. Avec notre alphabet, notre programme sera un ROT13 ou un ROT39 ou un ROT65 tout aussi bien :) Pour écrire un programme complet nous devons encore accéder à un élément de la liste (notre alphabet). Cela se fait en Haskell au moyen de l'opérateur (!!). Voici le code qui parle maintenant de lui-même :

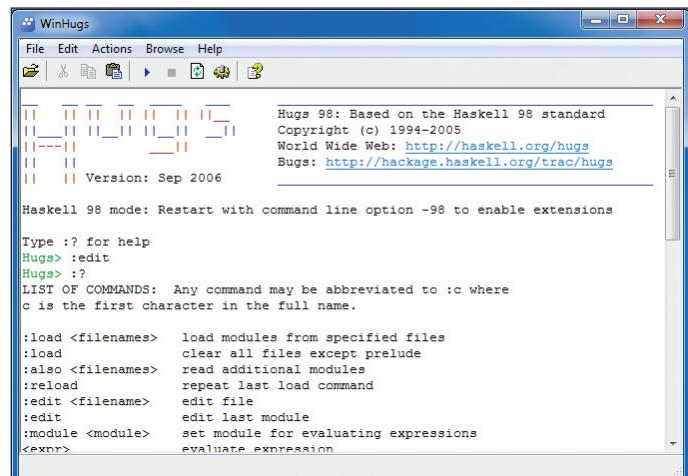
```
import Char

alphabet = ['A'..'Z'] ++ alphabet

rot13 c =
  let uc = toUpper c
  ordA = ord 'A'
  in alphabet !! (ord uc - ordA + 13)

main = do putStrLn $ map rot13 "Programmez"
```

Après avoir déclaré notre alphabet, nous déclarons une fonction rot13 qui code un seul caractère à la fois. Toutefois nous voulons coder des chaînes de caractères. En Haskell une chaîne est une liste de caractères. Il suffit donc d'appliquer notre fonction rot13 à chaque élément de la liste. En Haskell comme avec tous les langages fonctionnels, il existe une fonction dite d'ordre supérieur qui prend en argument une fonction et une liste, applique la fonction à chacun des éléments de la liste, et retourne la liste des résultats. Cette fonction s'appelle map. Il est maintenant aisé de comprendre la fonction main de notre programme. Dans cette ligne, le signe \$ signifie que la fonction putStrLn, c'est- à dire la sortie sur console d'une ligne, sera appliquée au résultat de map rot13 "Programmez" Il n'aura pas échappé au lecteur attentif que le nom de son magazi-



Hugs 98 est un interpréteur pour le langage Haskell

ne préféré s'écrit avec un point d'exclamation. En l'état, notre fonction rot13 ne marche pas si elle est appliquée à autre chose qu'une lettre. Nous la ré-écrivons donc en utilisant une syntaxe particulière d'Haskell (syntaxe courante avec les langages fonctionnels): les gardes. Voici notre nouveau programme :

```
import Data.Char

alphabet = ['A'..'Z'] ++ alphabet

rot13 c
| (c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') =
  let uc = toUpper c
  ordA = ord 'A'
  in alphabet !! (ord uc - ordA + 13)
| otherwise = c

main = do
  let s = map rot13 "Programmez!"
  putStrLn s
  putStrLn $ map rot13 s
```

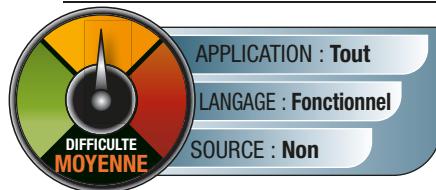
Ce qui est remarquable, avec tout ce code, c'est qu'à aucun moment nous ne nous sommes inquiétés de contraintes imposées par la machine. Nous ne nous sommes pas inquiétés le moins du monde du déroulement de ce programme. Totalement affranchis, libres, nous avons directement manipulé des concepts, des idées. Tout ceci peut-il trouver des applications pratiques ? Bien sûr. Imaginons simplement un nombre astronomique de possibilités que l'on manipulera comme une seule entité, que l'on filtrera pour examen dans un logiciel d'aide à la décision, dans un jeu de stratégie, etc. Avec Haskell l'ordinateur s'efface et la créativité du programmeur s'exprime à volonté. Si vous souhaitez travailler avec ce langage hors du commun, rendez-vous à <http://haskell.org> pour y trouver quantité de ressources. Vous pouvez utiliser l'interpréteur Hugs 98 (Figure ci-dessus) pour vous faire la main.

Puis rapidement vous préférerez travailler avec le compilateur open source GHC (Glasgow Haskell Compiler), qui est l'implémentation de référence du langage Haskell 98.

■ Frédéric Mazué - fmazue@programmez.com

Caml : méconnu mais brillant

Largement méconnu en France alors qu'il mérite un cocorico, CAML est le langage qui est utilisé lorsque performances et sûreté absolues sont requises.



en 1985 à l'INRIA, de la rencontre entre le langage ML et une machine abstraite : CAM. L'implémentation la plus aboutie s'appelle Objective Caml. Nous la devons à Xavier Leroy de l'INRIA, et comme son nom l'indique, cette implémentation ajoute une couche objet qui toutefois ne nous intéresse pas dans le cadre de ce dossier sur les langages fonctionnels. (Objective) Caml est un langage créé pour l'industrie, où il peine à s'imposer, sans doute par manque de communication.

INJUSTEMENT MÉCONNNU MAIS JUSTEMENT COPIÉ

Caml est un langage injustement relégué au monde universitaire. Cependant Microsoft ne s'y est pas trompé, lorsqu'il a été question de doter la plateforme .NET d'un langage fonctionnel, c'est Objective Caml qui a été choisi comme modèle, pour donner finalement F#. Caml a tout ce qu'il faut pour séduire l'industrie, car il allie la sûreté à la performance.

En tant que langage fonctionnel, il permet d'écrire des programmes prouvés, et il est rapide. Selon le type de code, sa rapidité est de 1 à 2 fois l'équivalent en C, en dépit de la réputation de lenteur que traînent les langages fonctionnels. Pour arriver à cela, le code CAML est compilé en code natif. Des implémentations sont disponibles pour Windows, Linux, Mac OS X et Solaris parmi d'autres.

Le compilateur effectue toute la panoplie des optimisations classiques, et tire aussi parti du fait que les informations de type sont totalement déterminées lors de la compilation, ce qui affranchit de tout test de typage au moment de l'exécution.

En outre, Caml est doté d'un ramasse-miettes spécialement conçu pour la programmation fonctionnelle et optimisé pour un rythme rapide d'allocation/libération de petits objets.

Il est même configurable pour garder son efficacité dans les situations atypiques. Créé pour l'industrie, (Objective) Caml vient avec des modules permettant de travailler avec les sockets, les entrées/sorties, la persistance, etc.

Le programmeur a à sa disposition tous les outils classiques, dont un interpréteur interactif pour se faire la main sur de petits morceaux de code, et un débogueur qui permet même le retour en arrière. Il est encore possible de compiler le code Caml en code-octet exécutable sur tout système qui dispose d'une implémentation, à la manière du byte-code de Java. Le code natif est quant à lui totalement autonome et ne nécessite pas une imposante plate-forme pour fonctionner, contrairement au F# de Microsoft. Enfin un toolkit graphique de base est proposé.

Des bibliothèques tierces, plus complètes ou plus spécifiques, pour OpenGL par exemple, sont disponibles. Vous pouvez télécharger Objective Caml à <http://caml.inria.fr/ocaml>.

En France, la programmation fonctionnelle n'est pas populaire. Pourtant c'est dans notre pays qu'est né un des meilleurs langages fonctionnels : Caml. Il est né

CE TYPE EST ABSOLUMENT SÛR

Caml est statiquement et fortement typé, et inféré. Son système de type provient du système de Robert Milner et de son langage ML. L'inférence de type, tout comme avec Haskell, est une faculté du compilateur à rechercher le bon type, ce qui dispense le développeur de devoir l'indiquer explicitement dans le code. Le système de typage ML supporte le polymorphisme paramétrique, c'est-à-dire des types. Ceci offre la générativité au langage. Prenons un exemple pour nous clarifier les idées, à la fois en ce qui concerne l'inférence et le polymorphisme paramétrique. Soit l'exemple classique d'une fonction qui calcule la longueur d'une liste qu'elle reçoit en argument. Il s'agit d'une fonction récursive. Voici le code :

```
let rec longueur = function
  [] -> 0
  | h::t -> 1 + (longueur t);;
```

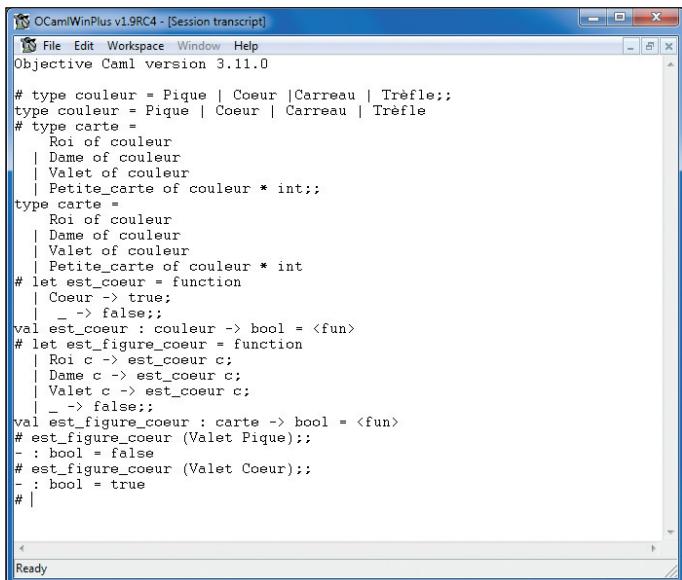
Nulle part il n'est dit explicitement que la fonction reçoit une liste en argument, et il est encore moins dit une liste de quoi. Le code ne donne pas non plus le type qui est retourné par la fonction. Le compilateur sait se débrouiller tout seul et si jamais on tentait d'utiliser cette fonction autrement que ce pour quoi elle s'évalue, c'est-à-dire un entier, le compilateur se plaindrait immédiatement. Dans ce code nous voyons également une caractéristique très importante des langages fonctionnels modernes: le filtrage de motif. Ici on filtre si l'on reçoit une liste vide et dans ce cas on retourne une valeur nulle, ou si l'on reçoit une liste non vide, auquel cas, la liste est décomposée en sa tête et sa queue, puis on ajoute 1 au résultat retourné par notre fonction appelée sur cette queue de site. On voit pour cet exemple que l'inférence de type n'est pas difficile pour le compilateur. Quand au type contenu par la liste reçue en argument, il n'a jamais besoin d'être connu :-)

ET ON PEUT COMPTER SUR LUI

Parce que Caml est rapide et extrêmement sûr, malgré son peu de popularité, il est utilisé lorsque la plus grande rigueur est requise. Microsoft, par exemple l'utilise, pour vérifier la correction des pilotes Windows écrits intra-muros. Airbus l'utilise dans son projet ASTREE, pour Analyse Statique de logiciels Temps-Réel Embarqués, pour la vérification statique des systèmes embarqués dans ses avions. Et effectivement, on préfère que tout fonctionne bien :-) Toujours chez Airbus, C'est avec Caml qu'a été écrit le compilateur du langage Lustre, utilisé pour les systèmes critiques tels que les systèmes d'avionique. Lustre est aussi utilisé dans les systèmes de contrôle de certaines centrales nucléaires. Lexifi, une entreprise française, spécialisée dans la conception de langages de programmation dédiés à la finance, utilise également Caml.

LES TYPES ALGÉBRIQUES

Les quelques exemples d'utilisation que nous venons de voir mettent en évidence que Caml brille lorsqu'il s'agit de créer des langages et d'écrire des compilateurs. Ceci est dû à ses types algébriques, dont



```

# type couleur = Pique | Coeur | Carreau | Trèfle;;
type couleur = Pique | Coeur | Carreau | Trèfle;;
# type carte =
  Roi of couleur
  | Dame of couleur
  | Valet of couleur
  | Petite_carte of couleur * int;;
type carte =
  Roi of couleur
  | Dame of couleur
  | Valet of couleur
  | Petite_carte of couleur * int
# let est_cœur = function
  | Coeur -> true;
  | _ -> false;;
val est_cœur : couleur -> bool = <fun>
# let est_figure_cœur = function
  | Roi c -> est_cœur c;
  | Dame c -> est_cœur c;
  | Valet c -> est_cœur c;
  | _ -> false;;
val est_figure_cœur : carte -> bool = <fun>
# est_figure_cœur (Valet Pique);;
- : bool = false
# est_figure_cœur (Valet Coeur);;
- : bool = true
# |

```

Haskell et d'autres disposent aussi d'ailleurs. Un type algébrique est un type de données dont chacune des valeurs est une donnée d'un autre type enveloppée dans un constructeur du type. Toutes les données enveloppées sont des arguments du constructeur. Contrairement à ce qu'on voit avec des langages comme C++, un constructeur n'est ici jamais exécuté. Un tel type est déclaratif seulement. On agit sur une donnée ainsi typée seulement en enlevant son constructeur dans un filtrage par motif. Prenons un exemple, que vous pouvez saisir dans l'interpréteur interactif de (Objectif) Caml : voir figure ci-dessus.

```
type couleur = Pique | Coeur | Carreau | Trèfle;;
```

Nous déclarons ainsi les couleurs de cartes à jouer. Voici maintenant le type carte dont la déclaration s'appuie sur celle qui précède :

```
type carte =
  Roi of couleur
  | Dame of couleur
  | Valet of couleur
  | Petite_carte of couleur * int;;
```

Nous pouvons maintenant déclarer une fonction retournant un booleen selon qu'une couleur est un cœur ou non :

```
let est_cœur = function
  | Coeur -> true;
  | _ -> false;;
```

On remarque au passage le motif universel `_` qui permet d'éviter d'écrire tous les cas. Et on apprécie la sûreté du langage qui vous avertira si un cas n'est pas traité. Pour le constater, retirez la dernière ligne de cette déclaration. Maintenant nous pouvons écrire une fonction testant si une carte est une figure de cœur :

```
let est_figure_cœur = function
  | Roi c -> est_cœur c;
  | Dame c -> est_cœur c;
  | Valet c -> est_cœur c;
  | _ -> false;;
```

On voit ici dans le filtrage de motif que si le constructeur est celui d'une figure, nous ôtons simplement celui-ci pour tester la couleur avec la fonction `est_cœur`. Vous pouvez ensuite essayer le code comme ceci, toujours dans l'interpréteur :

```
# est_figure_cœur (Valet Pique);;
- : bool = false
# est_figure_cœur (Valet Coeur);;
- : bool = true
```

De ce qui précède, on voit encore que les types algébriques de Caml sont potentiellement récursifs. Cela permet notamment de déclarer des arbres :

```
type tree = Empty
  | Leaf of int
  | Node of tree * tree
```

DES TYPES INÉDITS

OCaml est un langage qui cache quelques curiosités. Ainsi, outre les types classiques de la programmation fonctionnelle, ce langage permet de travailler avec des matrices non rectangulaires.

```
# let t = []
  [|1|];
  [|1; 2|];
  [|1; 2; 3|];
  [|1; 2; 3; 4|]
  () ;;
val t : int array array = [| [|1|]; [|1; 2|]; [|1; 2; 3|]; [|1; 2; 3; 4|] |]
```

UN SYSTÈME DE MODULES TRÈS ABOUTIS

Le Mécanisme de modules d'OCaml témoigne de la volonté de ses concepteurs d'en faire un langage industriel, qui permet le développement en équipe et favorise au maximum la réutilisation du code. Ainsi un développeur qui écrit du code qui doit utiliser les fonctionnalités d'un module n'est jamais contraint de disposer du code de celui-ci. Pour obtenir cela, tout module O Caml est fait de deux parties : une interface et une implémentation. Seule l'interface doit être connue par les utilisateurs du module lors du développement, et de son côté l'implémentation peut cacher du code au regard indiscret, si ce code n'est pas déclaré dans l'interface. Au chapitre de la réutilisation du code viennent les modules paramétrés. Ceux-ci sont aux modules ce que les fonctions sont aux valeurs. Tout comme une fonction construit une nouvelle valeur à partir des valeurs qu'elle reçoit en paramètres, un module paramétré permet de construire un nouveau module à partir de modules déjà construits. De tels modules prennent encore le nom de foncteurs.

On reproche parfois à OCaml de ne pas être capable d'exploiter les processeurs multicoeurs. Ceci est dû à la conception de son rama-sse-miettes, tourné vers la performance mais qui ne supporte pas la concurrence. Il existe toutefois des extensions pour contourner ce problème. La plus connue est sans doute jocaml, un autre projet de l'INRIA, qui rend OCaml apte à la programmation concurrente et distribuée.

■ Frédéric Mazué - fmauzé@programmez.com

100 % SÉCURITÉ

La sécurité dans le cloud



Difficulté : ***

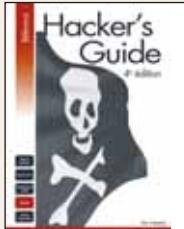
Editeur : Pearson

Auteur : Vic Winkler

Prix : 37 €

Le cloud pas sécurisé ? Comme une légende urbaine, on a beaucoup raconté sur les problèmes de sécurité du cloud, et parfois, d'énormes erreurs. Mais le danger est réel. Après avoir présenté les caractéristiques essentielles d'un Cloud, dont le modèle des services SPI (SaaS, PaaS et IaaS), les quatre types de déploiement possibles (public, privé, hybride ou communautaire), l'ouvrage décrit les points sensibles sur lesquels les attaques peuvent se focaliser : l'infrastructure, le réseau de communication, les données ou les services. La sécurité sur le cloud impose de revoir une partie des politiques de sécurité, de remettre à niveau les méthodes, les processus, les outils. Sans être exhaustif, cet ouvrage vous donnera les bases fondamentales pour comprendre la sécurité propre au cloud. Incontournable !

Hacker's guide 4e édition



Difficulté : ***

Editeur : Pearson

Auteur : Eric Charlton

Prix : 37 €

Cette édition mise à jour vous apporte de nouvelles informations sur les outils logiciels de sécurité et expose les principes de la stéganographie. Elle présente également les coffres-forts numériques, les logiciels d'anonymat, le phénomène des botnets ainsi que les méthodes de hacking dans les réseaux sociaux et micro-réseaux. Enfin, elle fait le point sur le droit et le piratage en 2011, prend en compte les dernières trouvailles technologiques, du carding au malware, et vous donne les moyens de vous en protéger. L'auteur revient sur la manière de reconnaître une attaque courante et savoir comment y remédier. Au-delà des techniques « classiques » des attaques et des contre-mesures, comment assurer une meilleure sécurité de vos données ou encore éviter les failles de TCP/IP ? Une mine d'or à (re)découvrir.

IOS

Concevoir des sites web pour iPhone et iPad pour les Nuls



Difficulté : **

Editeur : First

Auteur : collectif

Prix : 24,90 €

Il n'y a pas que les applications natives iOS pour faire le bonheur des développeurs et des utilisateurs. Le site web spécialement adapté est un autre must, parfois oublié. Les terminaux iOS imposent certaines technologies et des contraintes : HTML 5, CSS3, Ajax, jQuery, taille de l'écran. Ensuite, il faut tester le site, le mettre en production, le faire vivre. Simple d'accès, il est abondamment illustré et ponctué de nombreux exemples.

ANDROID

L'art du développement Android 3e édition



Difficulté : **

Editeur : Pearson

Auteur : Mark Murphy

Prix : 32,30 €

Cette nouvelle édition de l'art du développement Android supporte la v3, la v4 étant trop récente. L'auteur y aborde l'installation du SDK, les structures des applications Android, comment concevoir une bonne interface, les API, le support de HTML5, l'utilisation de PhoneGap. La seconde partie explique l'ajout d'interactivité avec les éléments d'interface (widgets et conteneurs), l'intégration d'un navigateur Web (via Webkit) et explore dans le détail les activités (activities) et les intentions (intents), composants applicatifs de base. Un bon ouvrage pour débuter.

agenda \\\\

NOVEMBRE

- Le 4 novembre 2011, Paris 10e, La Ruche. **Hack Day Paris** : Un marathon de 40 heures de R & D <http://hackdayparis.org/>
- Le 8 novembre 2011, Pavillon Royal Avenue de Suresnes 75116 Paris, **Eclipse Day Paris 2011**, organisé par les sociétés Zenika et ProxiAD. Plus d'informations. <http://www.eclipsedayparis.com/2011/>
- Du 8 novembre au 6 décembre 2011, **Tour De France** des versions 17 de WINDEV, WEBDEV et WINDEV Mobile. <http://www.pcsoft.fr/pcsoft/tdf-com/2011/index.html>
- Le 10 novembre 2011, Maison des associations, Paris 13e première édition de **Drupagora 2011**. Le

SAVOIR

Tout sur le développement logiciel 2e édition



Difficulté : *

Editeur : Dunod

Auteur : collectif

Prix : 12 €

Finalement, c'est quoi le développement d'un logiciel ? Comment le comprendre, le maîtriser ?

L'objectif de cet ouvrage est de permettre à tous de comprendre ce qu'est le développement logiciel, d'acquérir le vocabulaire et les compétences pour s'initier à n'importe quel langage et être en mesure de choisir un langage de programmation convenant à des besoins particuliers. Dans cette édition, les auteurs abordent les compilateurs open source ou encore le mapping relationnel objet.

SAVOIR

Ergonomie des interfaces



Difficulté : ***

Editeur : Dunod

Auteur : collectif

Prix : 35 €

L'ergonomie est un art difficile à maîtriser. Car un logiciel, une interface, sans ergonomie ne servent à rien, ou presque. Avec l'évolution rapide des terminaux et des systèmes, l'interface a évolué en 10 ans et de nouveaux modes d'interaction ne manquent pas de modifier l'ergonomie et la manière de penser celle-ci. Le tactile mais aussi les nouvelles métaphores et demain les outils vocaux tels que Siri d'Apple. L'ouvrage est illustré par des cas pratiques de conception d'interface pour lesquels les auteurs ont appliqué la démarche ergonomique. Ils témoignent concrètement du processus mis en œuvre pour réaliser une interface intuitive.

premier événement Européen sur Drupal dédié aux chefs de projets et aux directeurs des systèmes d'information.

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

• Le 24 novembre 2011, Paris La Défense, **MD DAY 2011**, la journée du Model Driven. Présentations d'experts (MDA, UML, SOA, ...), exposés de solutions « Model Driven », retours d'expériences, conclusions, perspectives d'avenir. <http://www.mdday.fr/>

• Du 24 au 25 novembre 2011, Lille, première édition du **PHP Tour**, organisé par l'AFUP. Développeurs, utilisateurs, prestataires et passionnés de PHP seront tous là. Plus d'informations : <http://afup.org/pages/phptourlille2011/inscription.php>

Formation *pour dirigeants et équipes IT*



- ✓ Gérer le **TEMPS**
- ✓ **MANAGER** les *projets*
- ✓ **COMMUNIQUER**

www.know-formation.com

CONTACT :

Stéphanie Khalif-Vennat : stephanie@know-formation.com - Tél. 01 74 70 48 91 - Fax 01 41 39 00 22

Know-Formation - Tour Albert 1^{er} - 65, avenue de Colmar - 92500 Rueil Malmaison



The place
to be*

Soyez parmi les millions
d'applications téléchargées
chaque jour sur Nokia Store.

Rendez-vous sur www.nokia.fr/developpeurs

NOOKIA
Connecting People