

OPEN SOURCE

Le monde s'ouvre à vous !

- *OpenData : la révolution des données ouvertes*
- *Créer son package de développement Java*
- *Mageia : au cœur du fork de Mandriva*
- *Non, MySQL n'est pas mort !*

**MySQL****mageia****Webmaster**

SketchFlow : créer la maquette d'un site web en quelques clics

SharePoint

*Il sait tout faire... ou presque :
Architecture, développement,
Powershell, sandbox : découvrez la
puissance de SharePoint au quotidien !*

Carrière

Le défi de la formation

**Microsoft Mix 2011****Silverlight 5 bêta dévoilé****Tag****Découvrez MS Tag****Web****Utiliser les API de Google Earth****iOS**

Maîtriser UITableView

Cloud

Déployez SVN sur Windows Azure

Langage

Mettez-vous à Purebasic

Windows

TPL Dataflow : la programmation concurrente par flux de données

M 04319 - 141 - F: 5,95 €



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

N°1 EN FRANCE

- Crée des applications Windows, Linux, Mac, ... compatibles Internet, Intranet, Android, Windows Phone 7 & Mobile
- Gestion complète du cycle de vie (ALM)
- Descripteur de données (UML, Merise, ...)
- RAD : créez une application d'un clic
- Lien avec toutes les bases de données (Lien natif, ODBC, OLE DB): Oracle, SQL Server, AS/400, Informix, DB2, MySQL, PostgreSQL, SQLite, ...
- HyperFileSQL: puissante base de données Client/Serveur gratuite incluse
- Générateur d'états PDF, Codesbarres
- Accès natif SAP R/3, Lotus Notes, ...
- Gestion de planning, des Exigences
- Audit de vos applications
- Langage de 5^e génération (LSG)
- SNMP, Bluetooth, TAPI, OPC, FTP, HTTP, Socket, Twain, API, DLL, XML, ...
- Domotique • Liaisons série et USB
- Débogage à distance, Profiler
- Multilingue (64 langues)
- Installateur 1-clic & Push
- Gestionnaire de versions
- Gestion de l'Infrastructure
- Tout en français
- etc...

Support technique gratuit

WINDEV®

16

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

VOTRE CODE EST MULTI-PLATEFORMES

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

GÉREZ LE «CYCLE DE VIE» DE VOS APPLICATIONS (ALM)

DÉVELOPPEZ

CONCENTREZ-VOUS SUR LES BESOINS MÉTIER

Elu «Langage le plus productif du marché»

VERSION EXPRESS GRATUITE

Téléchargez-la !

100 TÉMOIGNAGES SUR SIMPLE DEMANDE



996 NOUVEAUTÉS



Fournisseur Officiel de la Préparation Olympique

► DEMANDEZ LE DOSSIER GRATUIT

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

www.pcsoft.fr

sommaire

\\ actus

En bref	6
Les nouveautés de Silverlight 5	10
Agenda	16
Hardware	16

\\ sgbd

SkySQL renforce son offre	13
---------------------------------	----

\\ gros plan

C++ le langage incontournable	
C++ : soyez le maître de la jungle	18
Qt : l'interface C++ par excellence	20

\\ webmaster

Conception et prototypage d'un lecteur multimédia avec SketchFlow	22
---	----

\\ gros plan

Au cœur de SharePoint

Visual Studio 2010 et SharePoint 2010 : le couple détonnant.....	27
Les solutions sandbox : le développement sans empreinte serveur	30
Connecter Sharepoint 2010 à une source de données externe	33
Administration PowerShell pour SharePoint	36
SharePoint : un marché qui explose	38

\\ carrière

Formation des développeurs : un pari sur l'avenir

\\ dossier

Open Source : le monde s'ouvre à vous

Des données aux licences ouvertes

Open Source : défis et nouveaux marchés	45
MySQL : l'écosystème, le produit, La véritable histoire.....	46
Mageia : du fork à la réalité du projet	49
Les licences logicielles	51
Un socle applicatif Java sur des briques open source.....	53
Open (Source)Data)	57

\\ code

Click : le Web facile et léger !	60
La programmation avec PureBasic.....	63
Subversion dans les nuages	66
Apprendre à construire et à utiliser des tables views.....	69
Connectez le monde au web avec Microsoft Tag	72
Mettre en œuvre l'API Google Earth.....	74
TPL Dataflow, la programmation concurrente par flux de données.....	78

\\ temps libre

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



16



16



27



40



45



74



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.

DEVELOPPEMENT

Applications de Business Intelligence riches en WPF et Silverlight



Nos grilles de pivot pour WPF et Silverlight permettent à vos utilisateurs d'analyser leurs données et prendre des décisions clés. Visitez infragistics.com pour les essayer dès aujourd'hui!



NetAdvantage®
for Silverlight Data Visualization



NetAdvantage®
for WPF Data Visualization

Infragistics®

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



Innovation : du vide ou du concret ?

Le monde informatique pour continuer sa croissance, pour vendre, bref pour vivre, doit innover toujours et encore. Le terme « innovation » est cependant suffisamment flou et général pour y mettre ce que l'on veut dedans. Et depuis quelques années, on pourrait croire que seul le monde open source serait capable d'innover. Bien entendu, cela est totalement faux.

Comment dire qu'une nouveauté est une véritable innovation ? Le fait, par exemple, d'introduire le multiprocess dans les navigateurs web est-il une « simple » nouveauté ou une réelle innovation ? Pour notre part, nous serions tentés de penser qu'il s'agit d'une nouveauté. Par contre, le multitouch est une véritable innovation.

L'open innovation est souvent mise en exergue par certains penseurs libres, par certains projets tout en fustigeant l'informatique propriétaire et fermée. Une fois de plus nous revenons à la "guéguerre" aussi inutile que puérile entre le monde ouvert et le monde fermé. L'open source apporte des idées, des concepts, des améliorations qui permettent de faire avancer les logiciels, voire de pousser les éditeurs propriétaires à bouger. Nous l'avons vu avec Internet Explorer par exemple. Cette concurrence est saine et profitable aux utilisateurs. Nous l'avons aussi vu dans les formats documentaires avec ODF puis OpenXML. C'est une avancée réelle et permettant d'entrevoir la gestion du patrimoine documentaire sur le long terme, même si ces formats ouverts posent de nombreux problèmes d'implémentations.

Le cœur de l'innovation informatique provient des laboratoires et centres de recherches. Ce sont eux qui pensent, réfléchissent à des idées, des concepts qui seront, ou non, utilisés un jour. Les Mozilla Labs regroupent quelques dizaines de développeurs, de penseurs technologiques. Les idées fusent et elles sont développées. Ensuite, le tri est fait. Microsoft concentre sa recherche fondamentale autour de MS Research regroupant plusieurs centaines de chercheurs. Là, des technologies, des techniques sont lentement mûries et mettent parfois 5 ou 7 ans à émerger, comme ce fut le cas pour la table Surface. Et c'est là que le cœur des prochains Windows est élaboré. IBM, Oracle, Apple... font de même. Parfois, ces constructeurs et éditeurs s'associent avec des centres de recherches nationaux, comme l'INRIA en France, pour explorer des pistes complexes et totalement nouvelles comme l'informatique prédictible, la bio-informatique, la robotique de demain, les futurs langages de développement. Sans recherche fondamentale, l'informatique de demain n'existera pas.

Deux problèmes émergent : trouver des personnes capables de penser l'innovation et le financement. L'innovation ne s'improvise pas. On a beau mettre une communauté en action ce n'est pas pour autant qu'elle innove réellement. Elle peut trouver des astuces, des améliorations mais pas une innovation au sens strict du terme, la remarque vaut pour tous les éditeurs. L'innovation doit au contraire être structurée dans des laboratoires autour de chercheurs, de « penseurs ». Comme toujours, un problème trivial survient : le financement. Eh oui, car tout cela coûte cher, très cher. Et n'oublions pas que les consortiums, les principaux projets ouverts sont aidés, parfois financés par les éditeurs commerciaux.

Mais la révolution informatique peut aussi venir d'un simple garage ou dans sa chambre, avec une idée « simple » et géniale. Hier ce fut la micro-informatique, internet, aujourd'hui le cloud computing, demain ce sera la robotique.

■ François Tonic

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

Rédaction : redaction@programmez.com

Directeur de la Rédaction : Jean Kaminsky.

Rédacteur en Chef : François Tonic - ftonic@programmez.com. Ont collaboré à ce numéro : F. Mazué, S. Saurel, C. Calais, C. Peyrusse. Experts : T. Lebrun, A.D. Robison, T. Hanna, J-F. Brind'Amour, A. Nouvel, G. Bouveret, C. Su, J. Chomar, K. Arno, A. Nicolas, R. D'Alverny, P. Ognibene, F. Esnault, T. Parisot, P. Georges, A. Weinbach, C. Maneu.

Illustrations couverture : ©iStock/evirgen, ©iStock/CostinT

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 libre Component Source sur les abonnés et une partie du tirage et un encart Libre Linux Expo sur les abonnés.

Abonnement : ProgrammeZ 22, rue René Boulanger, 75472 Paris Cedex 10

Tél. : 01 55 56 70 55

abonnements.programmez@groupe-gli.com

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

**L'INFO
PERMANENTE**
WWW.PROGRAMMEZ.COM



PROCHAIN NUMÉRO
N°142 juin 2011
parution 28 mai

✓ Webmaster

Accélération matérielle sous IE 9

✓ Dossier spécial "TESTS"

Pourquoi et comment tester ?
Améliorer la sécurité grâce aux tests.

✓ SGBD

Faut-il encore utiliser MySQL ?

✓ Cloud computing

L'architecture multitenante
est-elle la meilleure ?

■ **AdaCore** a annoncé la version 2.0 de CodePeer, un outil d'analyse de code source capable de détecter statiquement des erreurs de logique et des erreurs d'exécution dans un programme Ada. CodePeer 2.0 inclut également plusieurs outils d'analyse communs avec la technologie GNAT Pro, un analyseur sémantique et un générateur de documentation.

■ La convergence entre le Web et la télévision a été le point de mire du deuxième atelier du **W3C** consacré au Web et à la télévision, qui s'est déroulé à Berlin en février. Plusieurs priorités ont été émises : Utilisation de HTTP pour une diffusion de contenu audio et vidéo, réseau domestique et interactions entre plusieurs écrans, extensions à HTML5, profils de tests.

■ Vous êtes étudiant ? Ingénieur jeune diplômé ? Le développement logiciel open source vous passionne ? L'**INRIA** a créé le concours "Boost your code", l'opportunité pour vous de présenter votre projet à un jury de scientifiques et de professionnels. Avec à la clé, un contrat d'un an pour développer votre projet au sein de leur institut. <http://www.inria.fr/actualite/actualites-de-l-inria/boost-your-code>

■ **Opera Software** vient de sortir la version finale de la mouture 11.10 de son navigateur. Celui-ci voit un moteur de rendu amélioré, avec un support de HTML5 étendu. Opera 11.10 supporte WebP, le format d'image de Google, ainsi que les polices de caractères au format WOFF. Enfin sous Windows 7 Opera 11.10 est annoncé comme réduisant sa consommation d'énergie pour les portables tournant sur la batterie.

JavaFX : Oracle n'a pas dit son dernier mot sur les RIA



Depuis le rachat de Sun par Oracle, JavaFX n'a guère fait parler de lui. Et pour cause : Sun n'a jamais su mettre en avant cette plateforme riche de type RIA face à Flex / Flash et Silverlight.

JavaFX devait fonctionner aussi bien sur mobile que sur desktop, avoir un modèle de développement simple et cohérent basé sur Java. Mais Adobe et Microsoft n'ont jamais cessé d'améliorer leur proche plate-forme et aujourd'hui, JavaFX ne présente pas de réel intérêt pour les développeurs. Oracle a décidé de remobiliser les équipes, les développeurs et le projet JavaFX a dévoilé un agenda complet, avec la liste des fonctions attendues et prévues dans les futures évolutions. Reste à savoir si cela va être suffisant pour espérer concurrencer les technologies installées et surtout pourquoi le développeur devrait supporter une 3e plate-forme alors que le développement web est déjà passablement compliqué et hétérogène.

La principale annonce est l'arrivée de JavaFX. Une version bêta devrait être disponible vers l'été 2011 (sans date précise). L'objectif est de recueillir les commentaires, le retour des développeurs et surtout les remontées des bugs, problèmes, etc. La version finale est attendue pour le 2e semestre 2011. Et il reste énormément de travail. Si les fonctions attendues dans le

Entity	High base deltas	Low base deltas	Medium base deltas	Low base deltas	Medium base deltas	Low base deltas
Test	16	16	80	80	160	160
admintoken.adb	4	4	2	2	10	10
admintoken.readandcheck	3	3				
admintoken.readandcheck.checkauthcert	1	1	1	1	5	5
admintoken.readandcheck.makedescription			1	1		
admintoken.readandcheck.checkidcertok				2	2	2
tcpip.adb	4	4	1	1	2	2
usertoken.adb	3	3	4	4	18	18
tokenreader.adb	1	1	2	2		
cert-id.adb	1	1	1	1	1	1
enclave.adb	1	1		2	2	2
clock-interface.adb	1	1		1	1	1
usentry.adb	1	1			1	1
configdata.adb			18	18	24	24
Message category	High	Medium	Low			
dead code	1					
test always false		1				
validity check	2	2				

programme de tests privés, celles de la bêta le sont à 25 % et aucune pour la version finale. Bref, si Oracle veut tenir le calendrier, il faudra aller vite.

Les principales nouveautés

JavaFX 2.0 mise sur des performances largement en hausse, un lot de nouvelles fonctions et API, une interface plus riche et un déploiement amélioré. Les principales modifications concernent les fonctions graphiques, le support des medias, les fonctions web. Tout d'abord, sur les API « générales », nous bénéficierons d'un nouveau lot de Java API en remplacement du JavaFX Script. Eh oui, nous n'aurons plus à apprendre le langage JavaFX.

Gain de temps ! Par contre, nous pourrions toujours utiliser Scala, des langages dynamiques. Oracle prévoit une interopérabilité avec Swing. D'autre part, la gestion multithread sera revue en profondeur. Autre bonne nouvelle, JavaFX 2 autorisera l'accès des applications par le clavier

(et 100 % accessible). La partie graphique bénéficie de sérieux perfectionnements, notamment : gestion dynamique et améliorée de la mémoire, accélération graphique 32 et 64 ciblant DirectX, OpenGL, transition plus simple, meilleure texture, refonte des conteneurs de calques, support des animations CSS.

Sur la partie media, JavaFX proposera une faible latence sur les pistes audio, l'arrivée de la vidéo plein écran, synchronisation entre media et animation. La partie web subit elle aussi une refonte totale : HTML DOM pour faciliter l'intégration HTML5, affichage de contenu HTML5, un moteur web pour afficher le contenu, les pages HTML.

Côté interface, le développeur bénéficie de nouveaux contrôles : player media, tab-view, fonction split, etc. Par contre nous n'avons pas encore de précision sur la partie outil, NetBeans devrait rester l'outil de référence. JavaFX 2 se veut l'héritier et la continuation de JavaFX 1.x. Comme nous l'avons précisé, la grosse nouveauté est l'abandon du langage interne au profit de Java. Et le compilateur JavaScript est depuis quelques mois à la disposition de la communauté.

Site : <http://javafx.com>



Retrouvez toutes les applis Nokia sur Ovi Store.



- Plus de 40 000 applis
- Des utilisateurs dans plus de 190 pays
- Des opérateurs offrant une facturation intégrée dans 34 pays
- Des applis pour plus de 135 terminaux dont 80 Symbian

NOKIA
Connecting People

www.nokia.fr/develop

ovi NOKIA



Perdu dans le Cloud Computing ?...

Retrouvez notre brochure
dans ce numéro



Erratum : «Votre domaine à partir de :
4,99 HT/an, soit 5,97 TTC/an.»

Parlons-en au : **0972 104 746** (coût d'un appel local)
ou rendez-vous sur : **ovh.com/cloudcomputing**



...OVH Global Solutions
accompagne
tous vos projets.

N°1 de l'hébergement Internet en France

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



Les nouveautés de Silverlight 5

Disponible depuis le Mix 2011, la première version bêta de Silverlight 5 apporte un grand nombre de nouveautés, que nous allons découvrir au sein de cet article.

Utilisation de la 3D

Sans doute l'une des nouveautés majeures apportées par cette version de Silverlight ! Silverlight 5 intègre la 3D accélérée, basée sur l'API de XNA (les développeurs XNA ne seront donc pas dépayés). D'un point de vue technique, ce sont les Shaders 2.0 qui sont supportés. Cependant, du fait qu'il s'agit d'une version allégée (pas de classe Effect), il est nécessaire de compiler, en mode offline, les shaders pour embarquer le code binaire [Fig.1].

XAML DataBinding Debugging

Là encore, il s'agit d'une fonctionnalité longtemps demandée par les développeurs Silverlight (et WPF !) et enfin accessible. En effet, avec Silverlight 5, vous avez la possibilité de mettre un point d'arrêt, dans le XAML, sur une expression de binding. Lors de l'exécution de l'application, Visual Studio s'arrêtera automatiquement sur le point d'arrêt et vous aurez la possibilité d'utiliser la fenêtre « Locals » pour étudier l'état du binding : [Fig.2].

Implicit DataTemplate

Fonctionnalité déjà présente dans WPF, elle permet aux développeurs de pouvoir créer des DataTemplates qui seront utilisés implicitement. Autrement dit, dès que le moteur Silverlight rencontrera un objet du type indiqué, il sait qu'il devra utiliser le DataTemplate défini, et cela sans que le développeur ait à le spécifier dans son code.

Multiples Windows

En mode "Out-Of-Browser With Elevated Privileges", il devient possible d'utiliser plusieurs fenêtres pour une même application. Attention, on ne parle pas ici de ChildWindow, le contrôle natif en Silverlight, mais bien de fenêtre (au sens Windows du terme). Des scénarios d'applications avec des fenêtres de détails, d'informations contextuelles, etc. s'ouvrent alors aux développeurs !



Fig.1

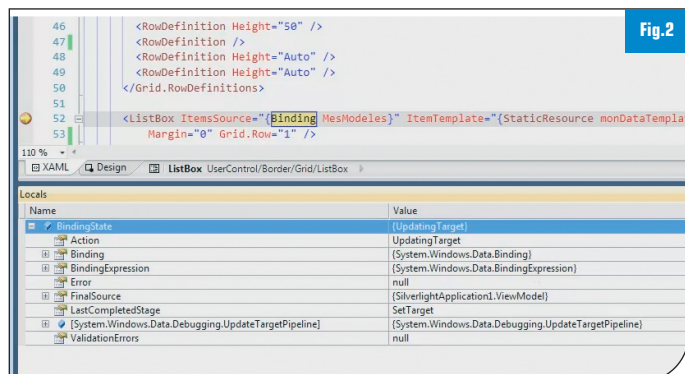


Fig.2

Custom Markup Extension

Silverlight 5, tout comme WPF (ce qui montre bien le rapprochement entre les 2 technologies), permet de créer des Markup Extensions personnalisées. Il s'agit des chaînes de caractères que l'on trouve généralement, dans le code XAML, entre les accolades, comme par exemple {StaticResource}, {Binding}, etc. La mise en place d'une Custom Extension passe par la création d'une classe héritant de MarkupExtension :

```
public class HelloExtension : MarkupExtension
{
    public HelloExtension()
    { }

    public override object ProvideValue(IServiceProvider serviceProvider)
    {
        return "Hello";
    }
}
```

Une fois la classe créée, il suffit d'utiliser son extension dans le code XAML :

```
<Grid xmlns:ext="clr-namespace:CustomMarkupExtensions">
    <TextBlock Text="{ext:Hello}" />
</Grid>
```

Cela peut s'avérer très pratique, par exemple pour simplifier le processus de localisation d'une application (comme démontré par Jeff Prorise : <http://www.wintellect.com/CS/blogs/jprorise/archive/2011/04/15/using-custom-markup-extensions-in-silverlight-5.aspx>)

Ancestor RelativeSource

Au niveau du Binding, il est maintenant possible d'indiquer que l'on souhaite se lier à un contrôle « ancêtre », via la propriété Relative-

```
Grid.Row="0"
Grid.Column="0"
CharacterSpacing="{Binding ElementName=slider, Path=Value}"
VerticalAlignment="Bottom"
```

```
<ListBox.Resources>
    <DataTemplate DataType="local:Female">
        <TextBlock Text="{Binding Name}" Foreground="Red" />
    </DataTemplate>
    <DataTemplate DataType="local:Male">
        <TextBlock Text="{Binding Name}" Foreground="Blue" />
    </DataTemplate>
</ListBox.Resources>
```

```
<Style TargetType="TextBlock"
    x:Key="tbStyle">
    <Setter Property="Foreground"
        Value="{Binding ElementName=tb, Path=Text}" />
</Style>
```

Fig.3

```
<sdk:DataGridTemplateColumn.CellEditingTemplate>
    <DataTemplate>
        <ComboBox ItemsSource="{Binding DataContext.Countries, RelativeSource={RelativeSource AncestorType=UserControl}}" />
    </DataTemplate>
</sdk:DataGridTemplateColumn.CellEditingTemplate>
```


Source. Cela permettra aux développeurs/intégrateurs de se faciliter la vie, lors de la mise en place des interfaces, en simplifiant l'écriture des Bindings ! Un grand plus donc, très appréciable !

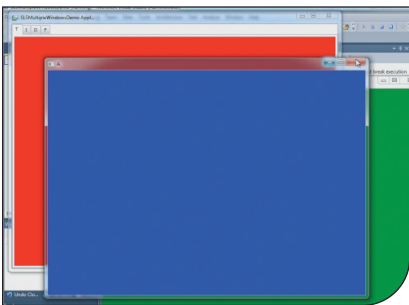
Performances / Support du 64 bits

La parseur XAML de Silverlight 5 a été optimisé pour permettre un lancement plus rapide des applications (« faster application startup »). De plus, le plugin sera, à terme, disponible en 64 bits pour les systèmes 64 bits (là où, jusqu'à maintenant, la version 32 bits était utilisée même sur un système d'exploitation 64 bits).

Support du P/Invoke

Non disponible dans la version bêta, mais annoncé pour la version finale, le support du P/Invoke (Platform Invoke) de code natif sera disponible, via l'attribut DllImport. Attention, cette fonctionnalité ne sera disponible qu'en mode « trusted » (donc avec des privilèges élevés) et rendra votre application spécifique à un système d'exploitation (Windows en l'occurrence) : il ne sera pas possible d'utiliser cette fonctionnalité sur Mac OS. On gagnera donc des possibilités techniques mais en perdant le côté

« portabilité » de Silverlight. De même, il sera possible d'accéder à tout le système de fichier de l'utilisateur, et non plus uniquement aux répertoires Mes Vidéos et Mes Images. Cependant, là encore, il faudra que votre application soit certifiée (pour éviter les problèmes de sécurité).



Utilisation du Web Browser

Depuis Silverlight 4, il est possible d'utiliser un composant Web-Browser pour afficher du contenu HTML au sens de son application Silverlight. Bien que pouvant s'avérer très pratique, cette fonctionnalité imposait à l'application d'être installée en mode « Out-Of-Browser ». Avec Silverlight 5, le WebBrowser sera utilisable également en mode « dans le navigateur » (« In Browser ») donc sans avoir besoin d'installer l'application en local.

Binding sur les Setters de Style

Cette fonctionnalité, une nouvelle fois disponible en WPF, va permettre aux développeurs d'utiliser le moteur de Binding de Silverlight dans les différents Setters des Style, afin de pouvoir référencer d'autres propriétés, issues de l'interface graphique, d'un ViewModel, etc. [Fig.3].

Fonctionnalités liées au texte

La propriété CharacterSpacing permettra de contrôler l'espace qui existe entre les différentes lettres d'un mot ou, de la même manière, entre les différents mots d'une phrase. Cela pourra s'avérer très pratique dans le cas où l'on souhaiterait créer des animations sur du texte par exemple. La fonctionnalité de « Pixel Snapping » (alignement des pixels à l'écran) permet d'avoir une meilleure lisibilité des polices, à l'écran. Silverlight 5 contient un très grand nombre de fonctionnalités, pour le plus grand bonheur des développeurs. Je ne saurais que trop vous conseiller de télécharger la version bêta et de vous faire votre propre avis !



■ Thomas Lebrun, Consultant / Formateur
MVP Client Application Development
tlebrun@access-it.fr / <http://blogs.developpeur.org/tom>



éligibles
au DIF

Formations DÉVELOPPEMENT

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

Concepts

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

Java, JEE, C++

Extrait du
calendrier

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

.Net

- Programmation C# avec Microsoft .Net Framework 4 6-10/06 • 27/06-1/07
- Développer des applications Windows avec Visual Studio 2010 12-16/09
- Introduction au développement Web avec Visual Studio 2010 4-8/07 • 3-7/10
- Développer des applications Web avec Visual Studio 2010 20-24/06 • 10-14/10
- Développer des solutions d'accès aux données avec VS 2010 23-27/05 • 8-12/08
- Développer des solutions WCF avec VS 2010 16-17/05 • 22-24/06

Open Source

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

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

www.globalknowledge.fr



Global Knowledge.

■ **Bull** s'intègre à l'offre sécurité d'Oracle, Advanced Security pour Oracle 11g, avec le module TrustWay. En utilisant Oracle Advanced Security et TrustWay box, les données sont chiffrées et déchiffrées de manière transparente dans la base de données Oracle, sans aucune altération des applications existantes, ni rupture de la continuité opérationnelle. Il est possible alors d'établir des



règles d'authentification forte pour les administrateurs et des mécanismes de contrôle pour l'établissement de responsabilités partagées afin de fournir une gestion centralisée de clés en rapport avec la politique de sécurité.

■ **Oracle** étend l'utilisation de Oracle Sun ZFS Storage Appliance, InfiniBand d'Oracle Exadata pour simplifier déploiement et sauvegarde. Oracle VDI supporte aussi ZFS pour un accès plus rapide aux ressources. L'offre ZFS Storage Appliance utilise les technologies et logiciels issus d'Oracle et de Sun.

■ **SQL Server** « Denali », la prochaine version du SGBD de Microsoft, est disponible depuis quelques mois en version technique, dite CTP, elle date de l'automne. La seconde CTP (qui devrait être disponible depuis quelques jours), corrigera les bugs de la première version et ajoutera de nouvelles fonctions comme un nouveau service d'analyses, un modèle sémantique pour la BI. D'autre part, l'équipe des outils SQL Server a annoncé « SQL Server Juneau » qui est une refonte profonde des outils Visual Studio pour les données. Ce sera le nouvel ensemble de développement dédié à SQL Server et basé sur VS Shell.

Adobe dévoile les versions 4.5 de Flash et de Flex

Adobe est-il réellement fâché avec Apple ? Pas entièrement comme l'ont démontré les outils pour les magazines numériques sur tablettes dont iPad (Digital Publishing). C'est au tour des applications Flex et Flash d'être disponibles sur les plateformes mobiles Apple, Android, Blackberry Tablet. Les versions 4.5 de Flash Builder et de Flex sont des éléments importants de la stratégie agnostique d'Adobe et de sa volonté de coller au marché. L'objectif est de disposer d'outils communs autour d'ActionScript pour ensuite déployer sur l'ensemble des plateformes, grâce à AIR.

Flash Builder 4.5 accélère le développement des projets Flex et ActionScript, notamment en autorisant les allers-retours dans Adobe Flash Catalyst CS5.5. Designers et développeurs peuvent ainsi travailler en parallèle tout au long du processus de développement d'applications. Flash Builder 4.5 inclut également des fonctionnalités rationalisant la programmation et offrant de réels gains de performances. L'ajout de Flash Builder 4.5 Premium dans Adobe Creative Suite 5.5 Web Premium constitue

une actualisation majeure de la célèbre suite logicielle destinée aux designers et développeurs web.

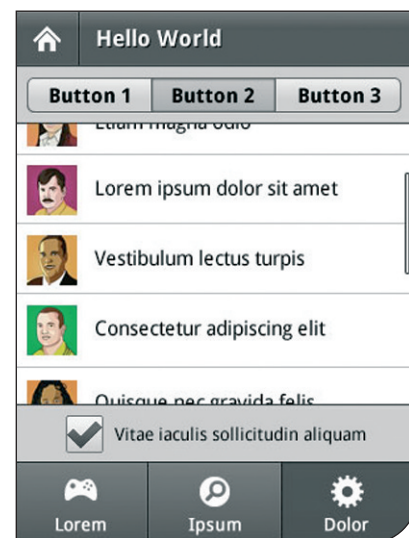
le nouveau Flex

Le framework Flex aura droit à des nouveautés courant de l'été prochain pour pouvoir s'exécuter sur les plateformes mobiles et faciliter le déploiement. Les nouveaux composants étoffent l'architecture d'habillage et de composants Spark introduite dans Flex 4, tandis que les perfectionnements de Flash Builder 4.5 – modèles de code, fonctions de génération et de finalisation de code, etc. – accélèrent la programmation des projets Flex et ActionScript.

Flash Builder 4.5 et Flex 4.5 intègrent des fonctions autorisant la création d'applications autonomes via Adobe AIR 2.6 pour les terminaux mobiles. Outre Google Marketplace, Blackberry AppWorld et iTunes AppStore d'Apple, les applications mobiles peuvent désormais être déployées sur Amazon Appstore pour Android.

D'autre part, Adobe présente une édition PHP de Flash Builder


der, en collaboration avec Zend. Basé sur Eclipse, Adobe Flash Builder 4.5 pour PHP est conçu pour accélérer le développement d'applications pour les terminaux mobiles, le web et les postes de travail à l'aide de PHP, d'ActionScript et du framework Flex Open Source. Il intègre en outre un exemplaire de Zend Studio 8 offrant aux développeurs un environnement de codage et débogage unique pour le déploiement d'applications PHP mobiles à la fois évolutives, ultraperformantes et sécurisées sur Android, Blackberry Tablet OS et iOS. Une partie des fonctions PHP le doit à l'intégration coordonnée entre Flash Builder et Zend Studio.



SkySQL renforce son offre

C'est durant la conférence utilisateur MySQL que SkySQL a dévoilé un ensemble de projets stratégiques : haute disponibilité, cloud computing, architecture, outillage. Le premier élément important est l'annonce d'une architecture de référence pour définir un modèle de déploiement dédié à la haute disponibilité, au cloud. Pour proposer son architecture de référence, SkySQL fait équipe avec les fournisseurs de technologies d'infrastructure de données les plus influents sur le marché, avec pour objectif la fourniture de solutions complètes et personnalisées pour les applications les plus répandues, notamment dans le domaine de la haute disponibilité, de l'évolutivité Web et des entrepôts de données/des solutions d'aide à la décision. Ces partenaires stratégiques apportent des technologies fon-

damentales qui renforcent considérablement l'architecture, tout en améliorant sa flexibilité et son élasticité. Les premiers partenaires sont Calpont, Canonical, Data Differential, Linbit, Monty Program, ScaleDB, Schooner Technologies, Webyog et Zimory, mais d'autres devraient rejoindre l'architecture de référence au fur et à mesure de l'évolution des besoins des clients. Avec le lancement de SkySQL Reference Architecture, le provisionnement des technologies d'infrastructure de données utilisant une base de données MySQL voit SkySQL redéfinir ses règles. Les clients optant pour les solutions basées sur SkySQL Reference Architecture ont tout à gagner de l'interopérabilité et de la facilité de gestion optimisées qu'elles apportent, et auront la possibilité de réduire les coûts de leur infrastructure




SkySQL Configurator - Alpha version

The first choice in affordable MySQL® Database solutions for the Enterprise and Cloud


Select your configuration

Choose the configuration you wish among those available in the current release of SkySQL Provisioning. You will then be guided through a process to gather all the parameters we need to build your custom installation package




Standard Server

Single server without High Availability or replication




Standard & Replication

Master server and up to 5 slaves with MySQL Replication



High Availability

Active and Standby servers in High Availability using DRBD



High Availability & Replication

Active and Standby servers in High Availability using DRBD as a master and up to 5 slaves with MySQL Replication

de données. SkySQL Provisioning, un système s'exécutant sur le Web et qui fournit l'interface graphique permettant à l'utilisateur de sélectionner les composants de son choix et de configurer une solution adaptée à ses besoins précis. Provisioning est disponible sur le site provision.skysql.com. De nouvelles fonctions seront proposées courant 2011. SkySQL

Provisioning apparaît comme un outil central dans l'offre de l'éditeur et doit permettre aux entreprises d'accélérer les déploiements, réduire le coût de possession. Surtout, le fait d'avoir un panneau de configuration en ligne est une idée intéressante pour préparer rapidement les packages à déployer sur son serveur, un peu comme SUSE Studio.



Concours de programmation parallèle inter-écoles ACCELER8

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

40 netbooks à gagner.

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



Les Rapports DevExpress



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



DevExpress[™]

WWW.DEVEXPRESS.COM

■ **Lindy** pense à vos voyages

Avec un kit pour connecter vos appareils électroniques où que vous soyez. Vous pouvez également utiliser l'alimentation USB pour charger vos lecteurs MP3, portables, iPods et PDA ! L'adaptateur dispose de 4 types de prises pour l'Europe, dont la Grande-Bretagne, les USA et l'Australie. Vous pourrez vous brancher dans les 4 coins du monde ! Prix public de 28€ TTC.

■ **Sony** veut concurrencer le

iMac d'Apple, avec un modèle tout en un : Vaio L. Doté d'un écran tactile sous Windows 7, il supporte l'affichage HD, embarque un lecteur Blu-Ray et un Intel Core 2 Duo (avec 4 Go de mémoire vive). Une machine taillée pour le desktop et la maison. Une belle alternative pour ceux qui veulent une machine quotidienne même si côté puissance pure, on sera limité. A partir de 1 200 euros.

■ Comment fixer dans une baie disque dur 3,5" un SSD 2,5" ? Il faut un chariot adapté pour connecter et fixer ses disques Flash. Début avril, **OWC** a dévoilé le Mount Pro, vendu environ 25 dollars.

■ **Samsung** a dévoilé un écran transparent de 22" qui sera, pour la première fois, fabriqué en grande série. Il supporte un contraste de 500 :1 et une transparence variable selon la couleur. Ce modèle se destine à différents marchés : publicitaire, en entreprise. Aucun prix pour le moment.

■ **Stealth** a dévoilé son nouveau mini PC haute performance à base de Intel Core i7, i5 et i3. Il intègre une connectique complète (hdmi, usb, dvi). La conception est solide et sa petite taille le rend « invisible ». Windows est proposé en option. Par contre, le prix (à partir de 1 650 \$) paraît excessif.

En attendant Google I/O

Mi-mai, Google organise sa conférence technique annuelle : I/O, occasion de revenir sur les derniers événements Google, les projets et surtout le futur immédiat, tout particulièrement smartphone et tablette, voire netbook avec Android 3 et Chrome OS.

Quelques modèles de netbooks sous Chrome circulent depuis décembre aux Etats Unis. Le concept de Chrome OS est un fonctionnement entièrement en ligne avec des applications web : rien ne s'installe sur la machine, rien n'est stocké. On dispose uniquement d'un navigateur. Et il faut une connexion au réseau permanente pour accéder aux applications et données. Quelques tests sortent. Les problèmes de pilotes, l'impossibilité de stocker localement et le besoin d'une connexion réseau sont souvent cités. L'offre en question concernera le coût des forfaits réseaux liés à la machine car si le netbook revient à 250 - 300 dollars, le forfait pourrait facilement atteindre 50 - 70 dollars par mois. Il est bien trop tôt pour tirer la moindre conclusion sur les futures machines et l'intérêt du système. Surtout, il faudra attendre



I/O pour savoir si Chrome OS et Android vont converger ou si Chrome OS finira par être purement et simplement annulé au profit d'Android.

Car jusqu'à présent, les applications Android ne sont pas compatibles Chrome OS ce qui constitue, à notre sens, une erreur. Il faut que Google trouve une solution pour unifier son offre système et éviter un double programme pour les développeurs.

Dans l'immédiat, Google doit aussi résoudre les défauts de jeunesse d'Android 3 dédié aux tablettes tactiles. Or, les premières machines mobiles connaîtraient un succès mitigé. Deux défauts seraient pointés du doigt : un système pas

tout à fait au point et le manque d'applications. Motorola avec sa tablette Xoom subirait un flop (même si aucun chiffre officiel n'a été donné). Se voulant le *iPad killer*, son prix handicaperait le produit ainsi que le système trop jeune.

On parle ici et là de 100 000 exemplaires vendus aux Etats-Unis, comparé aux millions d'unités de l'iPad, seconde génération.

Mais nous ne doutons pas qu'Android 3 se perfectionnera rapidement et la multiplication de l'offre devrait améliorer la visibilité sur le marché même si iPad 2 connaît un beau succès. ■

agenda \

MAI

Du 10 au 12 mai 2011, CNIT de Paris - La Défense
Solutions Linux/Open Source 2011
<http://www.solutionslinux.fr>

lundi 09 mai 2011, Paris, 8 rue d'Athènes, **Le Printemps du Libre**, le Think-Tank des professionnels du Logiciel Libre et Open Source - www.cnll.fr

Le 10 mai 2011, Paris 3e Espace St Martin, 3e édition des **GS Days**, Journées Francophones de la Sécurité

<http://www.globalsecuritymag.fr/Programme-previsionnel-des-GS-DAYS,20110303,22310.html>

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

jeudi 26 mai 2011, Séminaire gratuit en ligne Adobe : **Créer et gérer dynamiquement les sites Web en mode multi-canal**

Inscriptions en ligne sur <http://www.adobe-events-france.com/produit/livecycle/>

ETRANGER

Du 11 au 14 mai, Allemagne, Berlin : **Linux Tag 2011**
<http://www.linuxtag.org/2011/>

JUIN

jeudi 09 juin 2011, **WygDay 2011** - l'édition Lilloise des Microsoft Days. EuraTechnologies 165, Avenue de la Bretagne 59000 Lille
<http://wygday.wygwam.com/Inscription.aspx>

Du 14 au 17 juin 2011, Centre de Congrès de Disneyland Paris **Hack in Paris** pour découvrir la réalité concrète du hacking et ses conséquences pour le fonctionnement des entreprises.
<http://www.hackinparis.com>

**Spread for Windows Forms** à partir de € 697

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

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

**Codejock Xtreme Chart Pro ActiveX** à partir de € 160

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 empilée, 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

**Janus WinForms Controls Suite V4.0** à partir de € 637

Ajoutez des interfaces de style Outlook à vos applications .NET.

- Vues ruban, grille, calendrier, et barres chronologique/raccourcis
- Nouveau – Style visuel Office 2010 pour tous les contrôles
- Nouveau – Support des profils client Visual Studio 2010 et .NET Framework
- Janus Ribbon ajoute Backstage Menus et la fonctionnalité onglet comme dans Office 2010
- Prend désormais en charge la sélection de cellules multiples

**FusionCharts** à partir de € 139

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

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

C++

Le langage incontournable

2^e partie

Le mois dernier, nous avons débuté notre exploration du C++ et montré pourquoi, ce langage redevient incontournable. Les raisons en sont multiples : portabilité, pérennité, performance. Continuons ce mois-ci notre aventure C++.

Soyez le maître de la jungle

Dans la jungle des langages de programmation, C++ peut sembler une bête complexe. Pourquoi devriez-vous apprendre ce langage, outre le défi de l'apprendre ? C++ représente un défi, car il est puissant et pratique. Ses principes sont simples même si une partie de sa complexité vient de son évolution, très réussie, depuis un quart de siècle. Vous pouvez l'utiliser sans avoir besoin de connaître ses évolutions passées. Cet article n'est pas destiné à vous apprendre C++, mais plutôt à vous mettre en appétit avec quelques clés qui ont fait son succès.

L'applicabilité

Aucun langage ne peut tout proposer, même si C++ offre une utilisation très large. Voici quatre de ses points forts :

1 C++ s'utilise sur une très grande variété de matériels. Les environnements de développement existent sur des machines de toute taille, du plus petit système embarqué aux monstres de calculs !

2 C++ s'applique aux tâches, aux fonctions les plus diverses. Le langage permet de faire de la programmation système, des bibliothèques, aussi bien que de l'interface graphique, des analyses numériques ! Il est particulièrement adapté aux problèmes de performances ou aux grandes capacités mémoires. Lorsque ces facteurs ne sont pas critiques, par exemple dans la programmation web, des langages plus dynamiques, comme Java, C#, Python..., peuvent être plus pra-

tiques. Mais lorsque vous avez besoin d'exécuter du code sur des systèmes enfouis, de hautes performances, de manipuler d'importantes volumétries de données, ces langages ne peuvent rivaliser avec le C++ !

3 Le C++ va d'un (très) bas niveau à un niveau élevé de programmation. La clé du succès en C++ est de fournir une abstraction avec un minimum de contraintes. Sur le bas niveau, les sous-ensembles C permettent un accès efficace et fiable aux ressources matérielles. Par exemple, il fait une abstraction des adresses machines comme les pointeurs et les références. Sur le haut niveau, il fournit des mécanismes pour implémenter sa propre abstraction. Par exemple, les classes permettent de définir de nouveaux types. Ou encore, l'usage des templates. Vous pouvez réutiliser des bibliothèques hautement indépendantes tout en minimisant le

temps d'exécution. Vous pouvez apprendre le C++ en commençant par les sous-ensembles C ou apprendre le langage en commençant par les conteneurs, les algorithmes du Standard Template Library (STL). Si vous êtes familier avec le C, vous pouvez apprendre comment les classes sont des extensions (directes) de la structure C et C++ et comment les templates C++ sont propres et une alternative puissante aux macros. Surtout, vous n'avez pas besoin d'apprendre tout le langage. Beaucoup de développeurs apprennent des éléments clés, des points très précis du langage, les éléments qui leur sont utiles !

4 Une abondance de code réutilisable ! Cette riche base de données permet de vous focaliser sur les nouveaux défis au lieu de toujours réinventer la roue ! En outre, vous pouvez utiliser facilement des bibliothèques C en C++. Car C++ est omniprésent et de nombreux langages interopèrent avec lui !

Divers modèles de programmation

Certains langages forcent le modèle de développement. Par exemple, Java force l'orienté objet que vous le vouliez ou non. C++ encourage différents modèles de programmation et l'orienté objet en est un parmi d'autres. En effet, la norme ISO C++98 fait 776 pages mais ne mentionne l'orienté objet qu'une seule fois ! Ainsi, vous pouvez utiliser le C++ comme un langage procédural ou comme un « meilleur C » sur la surcharge des opérateurs, l'encapsulation via les déclarations *private* dans un *struct*. Une classe C++ n'est qu'un *struct* avec un accès au *private*. Ou vous pouvez aussi continuer à utiliser des variables globales et des *goto* en dépit de la déclaration de certains fanatiques de la programmation ! Et vous n'avez pas à abandonner vos codes C ! Vous pouvez convertir le code C en C++. Il est aussi facile d'appeler des bibliothèques C en C++. Vous pouvez utiliser C++ comme un langage orienté objet. Quand l'objet a débuté, il avait une réputation de faible performance, et de difficulté à compiler avec des langages comme Simula et Smalltalk. C++ a changé la donne. Il supporte à la fois l'héritage unique et multiple. Et l'usage d'une fonction virtuelle est à peine plus coûteuse qu'un simple appel de fonction.

Un autre modèle de développement est la programmation générique. En effet, le faible impact de l'abstraction a permis à la pro-

grammation générique de réellement décoller ! Dans ce modèle, vous pouvez implémenter des algorithmes de la manière la plus générique possible. Par exemple, écrire une routine pour le tri d'un tableau de chaînes dans l'ordre croissant. C'est assez spécifique car limité à des tableaux, et le tri se fait dans une seule direction. Vous pouvez utiliser le C++ STL header « *algorithm* ». C'est générique et efficace !

Un autre paradigme de développement C++ est la métaprogrammation par patron (template metaprogramming), le code s'exécute lors de la compilation. Des bibliothèques complexes utilisent cette approche pour optimiser les « décisions ». Par exemple, un nombre entier N connu au moment de la compilation, un métaprogramme peut calculer la factorielle de N. Autre approche de cette métaprogrammation par template, l'expression templates : une bibliothèque pour faire une optimisation de haut niveau lors de la compilation. Autre exemple, la bibliothèque « fusion » dans Boost. Elle utilise une combinaison métaprogrammation en compilation et une approche runtime pour être plus efficace en conteneurs hétérogènes. Pour en savoir plus : http://www.boost.org/doc/libs/1_46_0/libs/fusion/doc/html/fusion/

Outre les performances pures, la métaprogrammation permet aux bibliothèques de (bien faire) avancer les vérifications lors de la compilation, voir : http://www.boost.org/doc/libs/1_46_0/libs/mpl/doc/tutorial/dimensional-analysis.html. Toutes ces méthodes résument bien la philosophie intrinsèque de C++ : faites confiance aux développeurs. Certes, des fonctions, subtilités peuvent être mal comprises, mal utilisées mais si vous vous en servez judicieusement, quelle puissance sous le capot ! Et vous êtes libre de les utiliser, ou non, à votre gré.

Longévité

La longévité est un autre avantage (énorme) de C++ ! Après plus de 25 ans d'usage, le langage continue à croître et à s'adapter. Ainsi, la prochaine génération de la norme C++, alias C++0x, fournit un support pour l'écriture multithread, si cruciale aujourd'hui en contexte multicore. Elle inclut aussi les expressions lambda, simplifiant les routines génériques. Et bien que non officielle, cette évolution est présente (partiellement au moins) dans les dernières versions des compilateurs GNU, Intel et Microsoft.

Avec les différentes évolutions, il y a des éléments qui peuvent paraître étranges ou

redondants, mais ils sont vitaux pour l'exécution et la compatibilité avec du code vieux de 10, 15, 20 ans. Par exemple, C++ hérite des déclarations « *static* » du C. Un moyen pour limiter la visibilité d'un symbole dans un seul fichier source. C++ introduit d'autres solutions pour limiter cette visibilité.

Cependant, il existe des bizarreries incontournables. Dans les expressions, un tableau se désintègre quand un pointeur va vers le premier élément. La syntaxe déclarative abstraite peut être confuse pour les nouveaux développeurs mais comme pour la conjugaison, elle devient intuitive avec l'expérience. Pour mieux connaître l'histoire du C++ : **Barne Stroustrup**, *Design and evolution of C++*.

Performance

C++ est fait pour les performances sur un socle matériel commun. Nous avons évoqué plus haut, l'abstraction et son impact limité. Un élément clé de cela est le principe du « *pay as you go* ». Vous ne payez que pour ce que vous utilisez. Dans d'autres langages, vous risquez de payer plus cher l'abstraction. Ainsi, les fonctions virtuelles sont souvent utilisées pour ajouter des « *extra pointer* » à un objet pointant vers une table. Or, un objet peut avoir quelques bytes, mais plusieurs millions d'objets, bref, vous mesurez l'importance des pointeurs...

C++ permet de choisir des compromis entre la taille du code et la rapidité, spécialement dans le polymorphisme runtime et le polymorphisme compilateur. Pour résumer, si la taille du code est pour vous une priorité, écrivez les algorithmes pour fonctionner sur des objets abstraits dont le comportement des dépendances de type est encapsulé dans des fonctions virtuelles. Ainsi l'algorithme sera compilé d'un seul bloc de code appelant les fonctions virtuelles. Par contre si la performance est votre souci, la problématique sera toute autre. Vous pourrez utiliser un template pour écrire vos algorithmes+.

Remerciements

Pablo Halpern, Anton Potapov, Andrey Marochko. Traduit et adapté de l'anglais par François Tonic (Programmez !). Merci à Ralph Wargny (Intel).

■ Arch D. Robison

Utilisateur de C++ depuis 1988, architecte d'Intel Threading Building Block. Développe en C++ des jeux.

Qt : l'interface C++ par excellence

Le langage C++ est relativement portable et, bien écrit, le code fonctionne sur toutes les plateformes disposant d'un compilateur. Comme C++ standard n'a pas de GUI, Qt entre en jeu.



Lors de leur exécution, les applications Qt ne se distinguent pas des autres applications natives, elles sont même traduites par le compilateur natif. « La transcription » Qt/application native est effectuée par le framework Qt. Un fichier, désigné par .pro, décrit le contenu de l'application. Via un programme nommé qmake, il est transformé en un Makefile spécifique à une plateforme, les fichiers C++ et ui- sont convertis en C++ standard par moc. En dernier lieu, le compilateur natif compile le résultat. Les codes natifs peuvent donc être intégrés sans problème à une application Qt. Dans la mesure où ce code est bien scellé par des #defines, l'indépendance de la plateforme est maintenue.

Environnement d'exécution

Les applications Qt peuvent être développées avec divers éditeurs : pour Symbian la meilleure solution est d'utiliser Carbide, pour la plupart des autres plateformes Qt Creator (<http://qt.nokia.com/downloads/>). Les autres étapes se basent sur Qt SDK for Windows. Nous vous recommandons, dans tous les cas, l'utilisation de la version LGPL du framework. Les applications développées ne sont nullement concernées par ce choix et si les modifications au niveau des bibliothèques Qt ne sont pas transmises à Nokia, il faut procéder à nouveau à ces modifications après chaque mise à jour du framework. Il est maintenant temps de lancer Qt Creator au niveau du menu et de développer une nouvelle application de type Qt Gui en cliquant sur File -> New File or Project. Dans ce cas, le nom choisi est Prog1, la cible étant la version active de Qt. On saute les étapes Class Information et Project Management. L'exemple d'application est réalisé, le formulaire est ouvert dans le mode « Design ». Par ailleurs, dans 99,9% des exemples en rapport avec Qt, l'EDI est en anglais. Pour changer la langue de l'EDI, le meilleur moyen est de passer par « Tools-> Options->Environment ».

QMainWindow

Notre première application est une QMainWindow. C'est un formulaire à part qui peut contenir d'autres composants d'interface graphique (hérités de QWidget). Sur la gauche on

trouve la palette des composants qui peuvent être déposés dans le formulaire. Sur la droite, en haut, figure un aperçu du formulaire et, sur la droite, en bas, les propriétés des composants activés. Notre premier exemple est plutôt primitif : une zone de texte est remplie par l'utilisateur. En cliquant sur la commande le contenu de la zone apparaît dans une QMessageBox. Puis nous déposons une zone de texte (Plain Text Edit) et une commande de la palette des composants dans le formulaire et nous les rangeons. Nous adaptons ensuite leurs propriétés, nous donnons un nom à la zone de texte : TxtInput, le bouton est, lui, baptisé CmdGo. Dans un premier temps notre tâche est terminée, nous enregistrons notre formulaire et nous passons au mode d'édition en cliquant sur Edit, tout à gauche dans la ligne. Le texte suivant apparaît sur le fichier MainWindow.cpp.

```
MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
}
```

Au niveau du header, nous trouvons la déclaration suivante concernant l'objet UI :

```
Ui::MainWindow *ui;
```

Il s'agit là d'un fichier objet développé par moc qui réalise en C++ le formulaire que nous avons développé avec l'éditeur graphique dans le fichier .ui-. Lors du fonctionnement, le fichier ui- est utilisé pour accéder aux composants du formulaire.

Pour finir, nous pouvons démarrer le programme en cliquant sur Play et nous pouvons nous réjouir de notre formulaire.

Un peu plus de détails

Notre petit projet Qt contient tous les fichiers qui sont importants dans la pratique. Ci-dessous un bref résumé.

Fichier .pro

Ces fichiers règlent le développement de l'application. Tous les headers, sources, formulaires, etc. y sont référencés. Les fichiers .pro sont des fichiers texte qui peuvent être adaptés avec

n'importe quel éditeur de texte. Le paramètre Qt définit quels modules Qt sont nécessaires. TARGET définit le nom de l'application, et TEMPLATE si c'est une application ou une bibliothèque qui doit être développée. Sont ensuite listés : les fichiers code source, les fichiers header et les formulaires. Comme le montre l'exemple, les paramètres sur plusieurs lignes sont reliés par \ à la fin de la ligne ; la dernière ligne ne doit pas contenir de \.

Les fichiers .ui-

Les fichiers .ui- aident l'éditeur graphique à gérer les informations des formulaires. Chaque fichier de ce genre réalise un formulaire.

Fichier ui-*h/ Fichier moc-*cpp

Il s'agit de fichiers d'aide qui sont développés et gérés par Qt. Ils sont auto-générés et sans importance, ils changent sans cesse et ne doivent en aucun cas être chargés sur les systèmes de contrôle de version.

Signaux et slots

Malheureusement nos composants d'interface graphique ne réagissent pas jusqu'ici. L'utilisateur peut cliquer comme il veut, rien ne se produit. Et pour cause : il n'y a pas d'Event Handlers réagissant aux événements. Au niveau de Qt, les Event Handlers sont réalisés via le concept signaux/slots. Les objets envoient des signaux qui sont interceptés par d'autres objets au travers de soi-disant slots. Lors du fonctionnement, les instances réceptrices doivent s'inscrire auprès de l'émetteur, le processus d'Eventrouting est exécuté par le framework Qt.

Nous voulons réagir en cliquant sur notre commande. Nous ouvrons donc le fichier .ui- du formulaire dans Design Editor et nous activons le mode Edit Signals/Slots en cliquant sur F4 (ou sur l'icône dans la barre d'outils). Nous cliquons ensuite à gauche dans la barre. Après une procédure « glisser-déposer », l'éditeur crée une ligne partant de l'objet que nous intégrons au corps du formulaire (et NON pas à la zone de texte). Qt Creator affichera les particularités du nouveau lien. Les signaux émis par notre commande figurent sur le côté gauche de l'éditeur. Les slots du formulaire

figurent sur le côté droit. Mais comme nous voulons réaliser nous-mêmes un slot, nous cliquons sur Edit sur le côté du formulaire, nous en créons un dans la page qui s'ouvre, il est dénommé CmsGoClicked(). En cliquant sur Enter puis sur OK nous nous trouvons à nouveau dans le dialogue Configure Connections. Le signal de la commande « clicked () » est relié au nouveau slot créé. En cliquant sur OK, le lien s'affiche sous forme graphique, le formulaire est enregistré puis fermé. Au niveau du header du formulaire, le slot est défini comme suit :

```
class MainWindow : public QMainWindow
{
    ...
public slots:
    void CmdGoClicked();
};
```

BOX LAYOUT SVP

Les slots ne sont, en principe, que des fonctions membres. Tant les signaux que les slots peuvent donc transmettre des paramètres. Il convient toutefois d'observer les règles suivantes :

- Les signaux et les slots doivent avoir des types de paramètres identiques.
- Si un signal a plus de paramètres qu'un slot, ceci ne pose pas de problème. Les paramètres superflus sont ignorés. Mais si un slot a plus de paramètres qu'un signal, il n'est pas possible d'établir un lien.
- Les signaux et les slots qui remplissent les conditions précitées peuvent être reliés à volonté entre eux. C'est-à-dire qu'un slot peut répondre à plusieurs signaux et qu'un signal peut activer plusieurs slots.
- Des signaux peuvent aussi activer d'autres signaux si le second signal est relié au premier en tant que « slot ».

Dans la première ligne de la routine, une nouvelle instance QMessageBox est initiée en tant que Child. Ceci intervient par la transmission d'une référence au formulaire. Plus de détails suivront. Le texte de l'alerte est déterminé dans la seconde ligne. Il est issu de la zone de texte, l'accès aux composants du formulaire intervient toujours via l'objet ui. Dans la troisième ligne la QMessageBox est activée. Pendant l'exécution, la relation entre signal et slot est établie via la fonction Connect. Celle-ci repose sur des strings et ne peut, par conséquent, PAS vérifier la compatibilité signal/slot lors de la compilation. Après l'inclusion du fichier header QMessageBox le programme est terminé. Un clic sur l'icône verte Play et le jeu peut démarrer.

Layouts

Notre formulaire a une belle présentation. Mais comment se présenterait-il sur un Pocket PC avec une résolution d'écran de 240x320 pixels ou sur une Highend-Workstation avec 2560x1920 px ? Les développeurs sont quotidiennement confrontés à de telles questions dans le secteur du développement des multiplateformes. La réponse de Qt à ces questions est le système de la mise en page. Les composants d'interface graphique ne sont pas directement rajoutés au formulaire, mais intégrés à des soi-disant layouts. Ces layouts déterminent la position des composants au niveau de l'écran. Les composants déterminent eux-mêmes s'ils peuvent disposer de plus de place. Lorsque le formulaire est affiché ou que sa taille est modifiée, le moteur de rendu recalcule le formulaire sur la base des layouts et des informations contenues dans les composants sans que cela n'affecte le développeur. Au niveau de notre petit exemple, les tâches sont plutôt simples. La commande doit toujours se trouver au milieu du bord inférieur du formulaire, la zone texte doit occuper le reste de la place. A cet effet nous ouvrons à nouveau le fichier .ui au niveau de l'éditeur Design et nous activons le mode « Edit Widgets » avec F3. Pour le centrage, nous prenons deux spacers horizontaux et nous les plaçons à côté de la commande. Ensuite nous marquons les trois éléments et nous les intégrons à un layout horizontal en cliquant sur Lay Out Horizontally. Dès maintenant, ces trois composants sont « unis » et vont toujours se positionner les uns à côté des autres. Il ne nous reste plus que le problème de la zone de texte qui doit être positionnée par le biais de la commande. Ceci peut se faire aussi via un nouveau layout vertical, mais nous aurions alors besoin d'un autre layout de formulaire. Mais comme notre formulaire serait déjà complet via le layout vertical, nous marquons le formulaire et cliquons sur Lay Out Vertically. Dès maintenant notre formulaire s'adaptera automatiquement aux changements de taille. Il est donc temps d'aborder de plus près la fonction des composants utilisés. Pour faire fonctionner le moteur de rendu, un layout de formulaire est nécessaire. Celui-ci positionne tous les layouts et les composants qu'il contient – dans notre cas il les place les uns au dessus des autres.SizePolicy étant sur Extending en ce qui concerne la zone de texte, celle-ci prend le maximum de place. Sa hauteur est limitée par le second layout. Le layout inférieur contient deux spacers et la commande. Les spacers prennent toute la place disponible, les spacers se partagent la place à gauche et maintiennent la commande au milieu.

Objet Q/Gestion du stockage

Le header de notre classe d'exemple présente un autre argument qui n'est pas mis en œuvre en C++ standard : Q Object – Macro.

```
class MainWindow : public QMainWindow
{
    Q_OBJECT
```

Cette macro ordonne à moc de traiter la classe qu'elle contient en QObject – ceci fonctionne avec toutes les classes qui héritent directement ou indirectement de QObject. Cet héritage de QObject est (en général) requis pour les propriétés suivantes :

- Les lots et les signaux.
- La gestion du stockage.

Nous avons suffisamment abordé la question des signaux et des slots – la gestion du stockage représente un allègement pour les programmeurs en C++. Le langage en soi n'a pas de ramasse-miettes – le système Qt de gestion du stockage, dénommé Parent and Child System, tente de remédier à ceci. Il part du principe que chaque objet important relève d'un autre objet. L'objet « qui prime » et dénommé Parent, le second objet Child. Ceci apparaît clairement dans le cas de notre formulaire.

```
MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
}
```

La gestion du stockage fonctionne ainsi : si un objet Parent est supprimé, tous les objets Child le sont aussi. Si l'objet qui « prime » est un Child de QApplication (ou est sur le Stack), la boucle est bouclée – si l'application (ou le Stackframe) disparaît, tous les autres objets sans soutien du système d'opération disparaissent aussi.

Conclusion

Un développement multiplateforme avec Qt est simple, efficace et amusant. Le framework est gratuit. La documentation est disponible online et dans les magasins on la trouve en grande quantité. Il est clair qu'il n'est pas possible d'expliquer l'ensemble du Qt-Framework en deux pages. Mais avec les bases développées ici vous serez aisément en mesure de progresser – l'auteur vous souhaite beaucoup de plaisir !

■ Tam Hanna - CEO et fondateur
de Tamoggemon Limited
<http://www.tamoggemon.com>

Conception et prototypage d'un lecteur multimédia avec SketchFlow

Une des fonctionnalités innovantes présente dans Expression Blend depuis sa version 3 est la possibilité de concevoir des prototypes d'application rapidement en utilisant des composants dessinés à la main.

SketchFlow (uniquement disponible dans la version Expression Studio Ultimate) permet ainsi de démontrer la mise en page, la navigation entre les pages et offre la possibilité aux utilisateurs de donner leur feedback en ajoutant des commentaires sur les interfaces créées. Une fois le prototype terminé et satisfaisant complètement aux besoins et désirs des utilisateurs, il est possible de convertir le prototype directement en application réduisant ainsi considérablement le temps de développement. Dans cet article, nous verrons les différentes étapes de conception d'un lecteur multimédia Silverlight avec SketchFlow.

Création du projet

Commençons par ouvrir Expression Blend + SketchFlow 4 et créons un nouveau projet en sélectionnant « File > New Project... ». Dans la fenêtre qui s'affiche à l'écran [Fig.1], il est possible de choisir de créer une application SilverLight ou WPF. Dans notre cas, nous allons créer une application SilverLight, alors prenons bien soin de sélectionner SilverLight SketchFlow Application, de donner un nom à votre application, l'endroit où seront créés les projets SketchFlow, le langage utilisé (C# dans notre cas) et le .NET Framework utilisé pour le code généré (il y a le choix entre les Frameworks 3.5 et 4.0). Après avoir cliqué sur le bouton « OK »,

l'IDE va générer deux projets : MediaPlayer et MediaPlayerScreens. Le premier projet représente l'application alors que le second contient les différents écrans et composants créés pour SketchFlow.

Prenons un peu de temps pour détailler les parties les plus importantes de l'interface d'Expression Blend [Fig.2]. À gauche complètement, nous trouverons la barre d'outils permettant entre autres de sélectionner, zoomer, dessiner ou encore d'ajouter des contrôles aux écrans. À droite de la barre d'outils en haut se trouvent les onglets « Projects » et « Assets ». On peut voir dans l'onglet « Projects » l'arborescence des projets de notre solution (c'est la même arborescence que l'on retrouve dans Visual Studio 2010). L'onglet « Assets » contient l'ensemble des contrôles, styles (dont ceux propres à SketchFlow), les comportements, les formes, les effets, ... Sous ces onglets on retrouve l'onglet « Objects and Timeline » qui affiche l'arborescence des conteneurs et contrôles se trouvant dans l'écran courant au centre de l'interface d'Expression Blend. Dans la région centrale, nous trouvons le design des écrans par onglet et dans le bas, nous retrouvons le « SketchFlow Map ».

Cette partie est propre à SketchFlow et permet de définir la navigation entre les écrans. Finalement, dans la partie de droite

nous apercevons trois onglets dont l'onglet « Properties » qui permet de définir les propriétés du ou des contrôles sélectionnés (couleur, opacité, taille, position, ...).

Définition de la navigation

La première chose à mettre en place pour concevoir notre prototype est de définir la navigation entre les différents écrans que nous allons devoir créer. Par défaut il y a un seul écran appelé « Screen 1 » dans la section « SketchFlow Map ». Renommons déjà cet écran en double-cliquant sur le rectangle bleu et appelons-le « Login », ce sera le point d'entrée de notre application qui permettra à l'utilisateur de s'authentifier pour accéder à sa propre liste de lecture personnalisée. Pour créer les autres écrans, il suffit de cliquer sur le bouton droit de la souris dans la section « SketchFlow Map » et sélectionner « Create a Screen ». Nous allons ainsi créer et renommer deux autres écrans : « MediaList », qui permettra de gérer la liste de lecture, et « MediaPlayer », qui permettra de jouer les morceaux (audio ou vidéo). Pour définir l'ordre d'exécution des écrans, il suffit de passer la souris sur un des rectangles bleus (« Login » par exemple) et de cliquer sur le deuxième bouton de la barre d'outils qui s'affiche sous le rectangle puis draguer la flèche vers le rectangle voulu (dans notre cas sur « MediaList »)

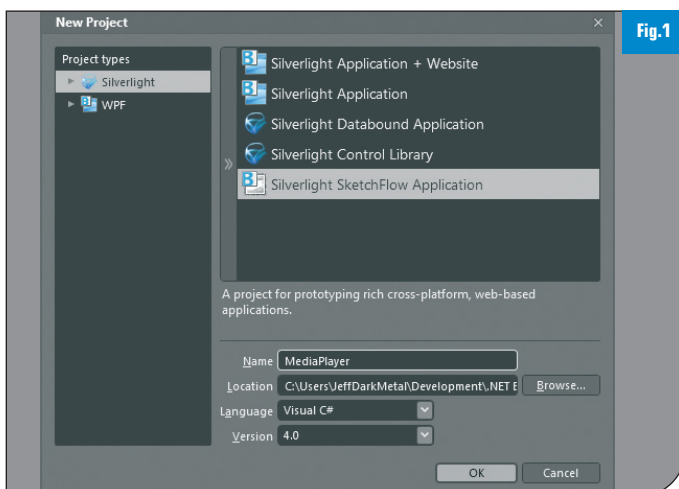


Fig.1

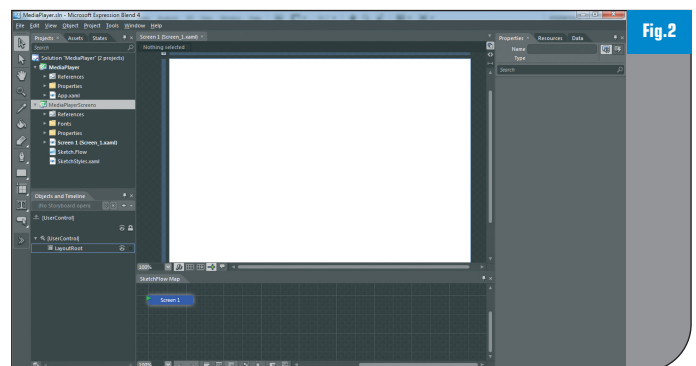


Fig.2

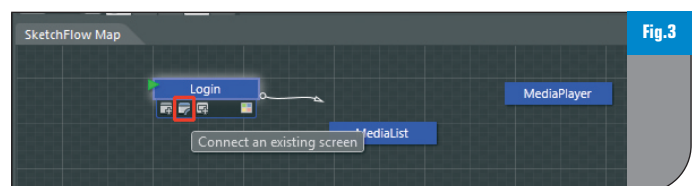


Fig.3

[Fig.3]. Ce lien signifie que de l'écran « Login » nous pourrions naviguer vers l'écran « MediaList ». Ajoutons un deuxième lien de « MediaList » à « MediaPlayer ».

Dans SketchFlow, il est également possible de créer des écrans sous forme de composants réutilisables dans plusieurs écrans. Créons un premier composant écran en sélectionnant « Create a Component Screen » dans le menu contextuel du « SketchFlow Map » que nous appellerons « PlayerBar ». Dans SketchFlow, les composants écran sont représentés par des rectangles ronds de couleur verte. Pour spécifier sur quels écrans ils doivent se retrouver, il faut cliquer et draguer les composants au-dessus des rectangles bleus des écrans et ainsi une flèche verte en pointillé se tracera entre le composant et l'écran [Fig.4]. Créons un autre composant écran qui s'appellera « Header » et traçons tous les liens comme dans la [Fig.5]. Pour résumer, nous aurons trois écrans dont deux qui contiendront deux composants écran réutilisables. Le composant « Header » sera l'en-tête qui contiendra un lien pour se déconnecter et le composant « PlayerBar » contiendra les contrôles pour jouer, arrêter, passer aux morceaux suivants ou précédents, ainsi que pour afficher la liste de lecture ou la cacher.

Design de notre première page

Les contrôles utilisés dans SketchFlow ont un style bien différent des autres contrôles classiques d'Expression Blend car ils permettent de concevoir des maquettes comme si elles avaient été dessinées sur papier. Pour accéder à ces contrôles au design un peu spécial, il faut aller dans l'onglet « Assets » et sélectionner SketchStyles dans la section Styles et une liste d'éléments s'affichera à la droite [Fig.6]. Pour réaliser le design de notre écran « Login », il faut double-cliquer sur le rectangle bleu « Login » dans le « SketchFlow Map » pour

en afficher le design au centre, puis il suffit d'ajouter les éléments voulus de la liste de contrôles Sketch. Ajoutons un rectangle pour la fenêtre de login, un titre, deux blocs de texte, une boîte de texte pour le nom d'utilisateur, une boîte de mot de passe et un bouton. En sélectionnant un ou plusieurs contrôles avec la flèche de sélection, il est possible de voir et de modifier de nombreuses propriétés telles que la couleur de fond et de la bordure, l'opacité, la taille, la position, l'alignement, etc. Étant donné que ce n'est qu'un prototype, nous allons simplement ajouter une couleur de fond au rectangle en sélectionnant le rectangle puis en exécutant les trois étapes de la [Fig.7]. Pour modifier le texte du titre, des blocs de texte et du bouton, il faut double-cliquer sur les contrôles et entrer le texte voulu. Après avoir aligné et redimensionné les contrôles, l'écran devrait ressembler à la [Fig.8].

Les composants écran

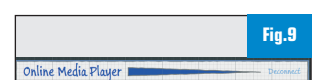
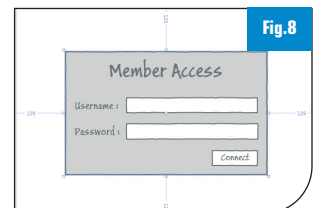
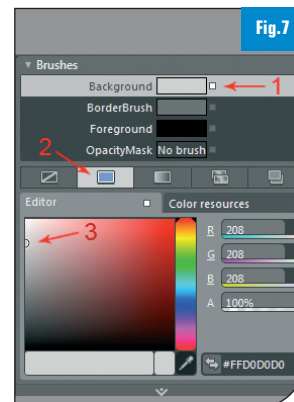
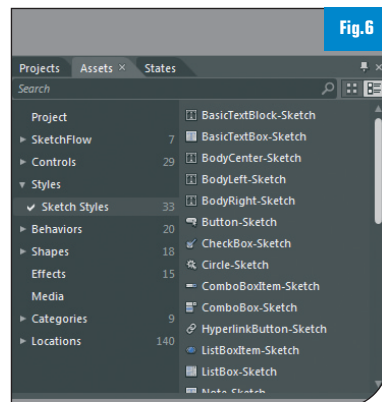
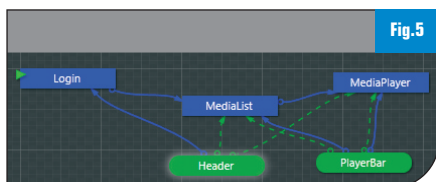
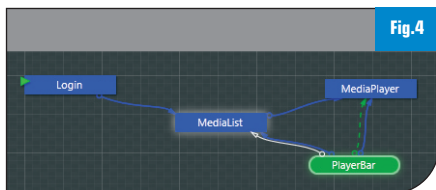
Attaquons-nous maintenant aux deux composants écran « Header » et « PlayerBar ». Afficher le premier composant « Header » à l'écran en double cliquant sur le rectangle rond vert « Header » dans la section « SketchFlow Map ». Ce composant va prendre toute la largeur des écrans « MediaList » et « MediaPlayer », alors on va sélectionner « LayoutRoot » dans la section « Objects and Timeline », et dans la section des propriétés, on va lui définir une largeur de 640 et une hauteur de 47. On va ensuite ajouter un rectangle, un titre, un triangle et un hyperlien. Le rectangle servira de fond au header (il prendra toute la taille du composant « Header »), le triangle ajoutera un peu de convivialité et l'hyperlien va permettre à l'utilisateur de se déconnecter et de retourner à l'écran de « Login ». La [Fig.9] montre un exemple de comment on pourrait agencer ces différents contrôles (le triangle a subi une rotation de 90 degrés et les couleurs

ont été modifiées). Il est important de noter que pour que les modifications faites aux composants écran soient prises en compte dans les écrans où ils se trouvent, il faut aller dans le menu et sélectionner « Project > Build Project » pour compiler le projet. On peut à présent double cliquer sur l'écran « MediaList » ou « MediaPlayer » dans la section « SketchFlow Map » pour voir l'effet le header modifié.

On va appliquer le même procédé pour faire le design du composant écran « PlayerBar », ce qui donnera la figure [Fig.10]. Ce composant a une taille de 640 par 75 pixels et les contrôles suivants ont été ajoutés pour créer ce design : un rectangle en arrière-plan avec un fond de gris, cinq cercles pour faire les boutons, un slider pour pouvoir modifier le volume et une barre de progression pour visualiser et modifier la position du media en cours. Les premier, troisième et quatrième boutons ont été créés en ajoutant une zone de texte par-dessus (« > », « |< » et « >| »). Pour ce qui est du deuxième bouton, le bouton stop, un rectangle a été créé avec un fond de la même couleur que sa bordure. Pour le dernier bouton, l'outil « Pencil » ou crayon a été utilisé pour dessiner à la main les 3 lignes horizontales [Fig.11]. Pour plus de facilité par la suite, nous pouvons sélectionner chaque bouton avec son texte, puis, dans le menu contextuel on va choisir « Group Into > Canvas » ce qui aura pour effet de regrouper les boutons et leur texte dans leur propre canevas. Il ne reste qu'à compiler le projet et repositionner la « PlayerBar » sur les deux écrans « MediaList » et « MediaPlayer ».

Les sources de données

Dans l'écran « MediaList », nous allons ajouter deux ListBox-Sketch, une pour rechercher des medias existant dans le site web et l'autre pour contenir notre liste de lecture. SketchFlow permet de définir des sources



de données de multiples provenances (XML, objets, ...) afin de fournir des exemples de données à présenter à l'utilisateur. Pour notre prototype, on va fournir une liste d'informations sur des medias qui seront affichés dans la liste des résultats de recherche de media. Dans l'onglet « Data », cliquons sur le bouton de la figure [Fig.12] afin de créer une nouvelle source de données d'exemple. Dans la fenêtre qui s'affiche, nous donnons un nom à la source de données et cliquons sur OK pour créer cette source de données générée automatiquement. Nous allons modifier cette source de données pour afficher des medias un peu plus significatifs que ceux générés. Premièrement, renommons les deux propriétés par défaut et ajoutons une autre propriété avec le bouton « + » comme sur la figure [Fig.13]. Ensuite, cliquons sur le bouton d'édition des valeurs et la fenêtre de la figure [Fig.13] s'affichera, nous permettant de modifier les valeurs des données s'y trouvant. Maintenant, rien de plus facile pour lier la ListBox de recherche de medias à la source de données que nous venons de créer. Il suffit uniquement de draguer l'élément « Collection » de la source de données dans l'onglet « Data » jusque sur la ListBox et le DataBinding se fera automatiquement comme nous pouvons le voir sur la figure [Fig.14].

Pour modifier et améliorer l'affichage des données dans la ListBox, il faut sélectionner « Edit Additional Templates > Edit Generated Items (ItemTemplate) > Edit Current » et

dans la section « Objects and Timeline » nous verrons s'afficher l'« ItemTemplate » propre à chaque élément de la ListBox. En modifiant la propriété « Orientation » du « StackPanel » et en ajoutant des zones de textes avant et après le temps d'un media, nous obtiendrons la figure [Fig.15]. Pour l'autre liste contenant notre liste de lecture, nous allons ajouter un élément manuellement en sélectionnant « Add ListBoxItem-Sketch » dans le menu contextuel de cette ListBox. Une fois terminé, l'écran devrait ressembler à la figure [Fig.16]. Terminons le design de l'écran « MediaPlayer » en ajoutant un rectangle au fond noir et avec un texte au milieu permettant d'identifier l'endroit où se trouvera la zone pour les vidéos [Fig.17].

Donnons vie au prototype

Les trois écrans sont maintenant terminés, mais nous pouvons aller plus loin en définissant des actions de navigation, de comportement, de changement d'état ou encore de lancement d'animations. Commençons par la navigation entre les écrans. Nous avons déjà fait le plan de navigation entre les pages au début de cet article, mais il reste à lier ces actions de navigation aux boutons des écrans. Rien de plus simple, il suffit de cliquer sur le bouton, l'hyperlien, ou n'importe quel autre contrôle et de sélectionner « Navigate To > Le nom de l'écran ». Il est également possible de choisir l'option « Back » ou « Forward » pour aller à l'écran précédent ou suivant. Nous pouvons donc

appliquer ce principe aux différents boutons et contrôles de nos trois écrans : le bouton « Connect » de l'écran Login, le lien « Disconnect » du « Header », ... Pour le bouton complètement à droite de la « PlayerBar », la navigation est un peu plus compliquée car le même bouton permet d'aller tantôt à l'écran « MediaList », et si on est déjà sur l'écran « MediaList », il nous permet d'aller aussi sur « MediaPlayer ». Nous allons utiliser pour cela les comportements et le DataStore de SketchFlow. Affichons tout d'abord l'écran « MediaList » et dans l'onglet « Assets », sélectionnons « Behaviors » puis nous verrons l'élément « SetDataStoreValueAction » qui permet d'assigner une valeur dans un DataStore. Dans l'onglet « Data », sélectionnons « New Data Store... », après avoir cliqué sur le bouton de la figure [Fig.18]. Nous pouvons donner un nom à notre source de données et renommer la propriété de ce nouveau DataStore (dans notre cas on va l'appeler « Name »). Maintenant, nous allons draguer l'action « SetDataStoreValueAction » sur le « LayoutRoot » de notre écran « MediaList » et lui assigner les propriétés « Property » et « Value » de la figure [Fig.19]. Veuillez noter que nous avons mis dans le DataStore le nom de l'écran à appeler depuis l'écran « MediaList ». Nous ferons le même processus pour l'écran « MediaPlayer ». Affichons ensuite le composant écran « PlayerBar » et ajoutons le behavior « NavigateToScreenAction » sous le canevas correspondant au dernier bouton

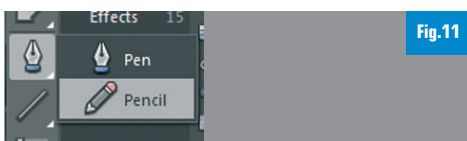


Fig.11

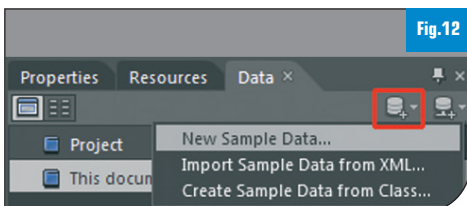


Fig.12

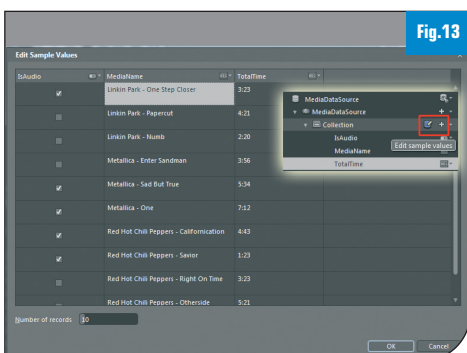


Fig.13



Fig.14

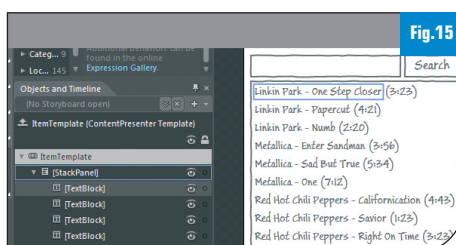


Fig.15

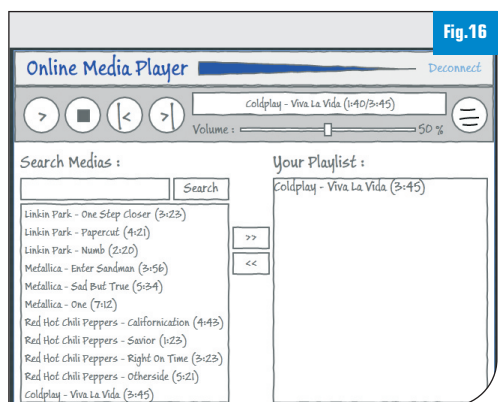


Fig.16

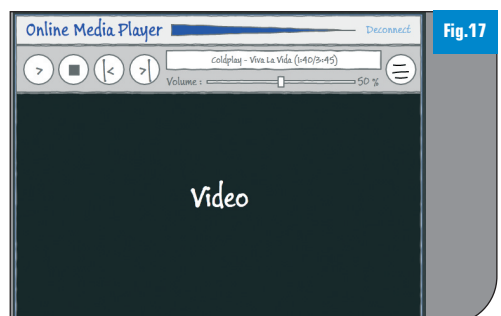
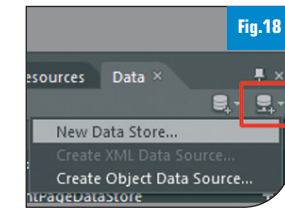
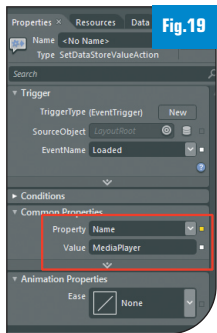


Fig.17

de la « PlayerBar » (celui qui contient les trois lignes). Dans les propriétés de ce behavior, nous pourrions faire du DataBinding sur la propriété « TargetScreen ». Il existe de nombreux behaviors dans SketchFlow, mais il y a également la possibilité de créer des animations servant à démontrer une suite d'interactions visuelles à l'utilisateur. Une autre fonctionnalité de SketchFlow est de permettre de passer d'un état à l'autre en définissant plusieurs groupes d'états et en enregistrant plusieurs états par groupes avec la possibilité d'avoir un délai de transition. Par exemple, nous pouvons voir comment créer un groupe d'état pour faire avancer la barre de progression du media joué. Nous pouvons aussi faire à différents endroits du DataBinding de contrôles (par exemple



pour le Slider du volume et le pourcentage affiché à sa droite).

Le feedback utilisateur

Une fois la première version du prototype terminé, il est temps de le tester et de le faire valider par les utilisateurs. Nous pouvons préparer un package pour l'utilisateur en sélectionnant « File > Package SketchFlow Project... », et en choisissant un répertoire de destination. Ce package sera à livrer à l'utilisateur et il pourra lancer le fichier TestPage.html pour naviguer dans le prototype. Testons notre projet en sélectionnant « Project > Run Project ». Un navigateur internet s'ouvre et affiche l'application Silverlight avec une section à gauche contenant le SketchFlow Player. Ce dernier permet de naviguer entre les pages, de lancer les animations et changements d'états et également d'écrire des feedbacks ou encore de faire des dessins sur l'interface sans en altérer la version originale. Supposons que sur la page « Login » l'utilisateur voudrait un lien pour aller vers une page d'inscription, alors il peut sélectionner le pinceau d'une certaine couleur et commencer à écrire sur l'écran « Login ». Il peut également écrire des commentaires

dans la partie Feedback du SketchFlow Player. Lorsqu'il aura terminé d'écrire ses commentaires, il pourra les envoyer sous forme de fichier *.feedback en cliquant sur le bouton dossier et en sélectionnant « Export Feedback... ». De retour dans Expression Blend, commençons par afficher la fenêtre des feedbacks en sélectionnant dans le menu « Window > SketchFlow Feedback ». Nous pouvons voir les commentaires et modifications de l'utilisateur en cliquant sur le bouton « + » et en sélectionnant le fichier *.feedback créé auparavant. Si on affiche la page Login et qu'on clique sur le commentaire dans la section Feedback, nous verrons le dessin fait par l'utilisateur. Veuillez également noter qu'il y a une petite ampoule au-dessus du rectangle bleu « Login » dans la section « SketchFlow Map » permettant de voir rapidement où se trouvent les commentaires. Il ne reste plus qu'à faire évoluer le prototype pour répondre aux besoins de l'utilisateur tout en évitant de redessiner les maquettes à la main.

■ Jean-François Brind'Amour

jfbrindamour@gmail.com

Consultant senior en développement .NET pour la société de services Groupe onePoint

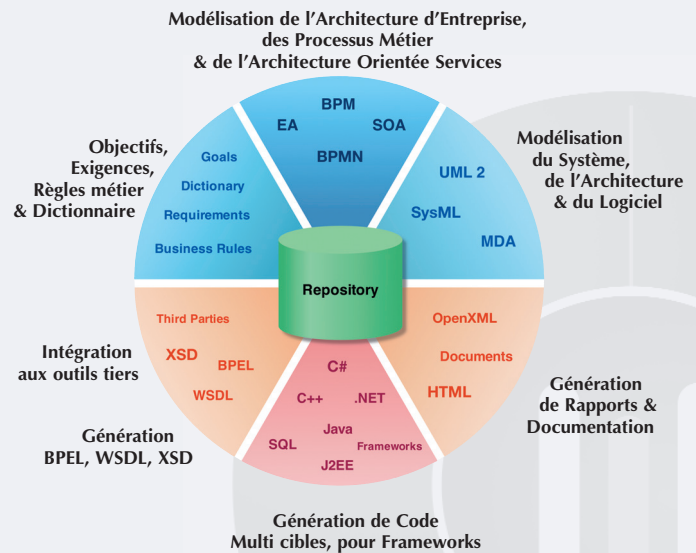
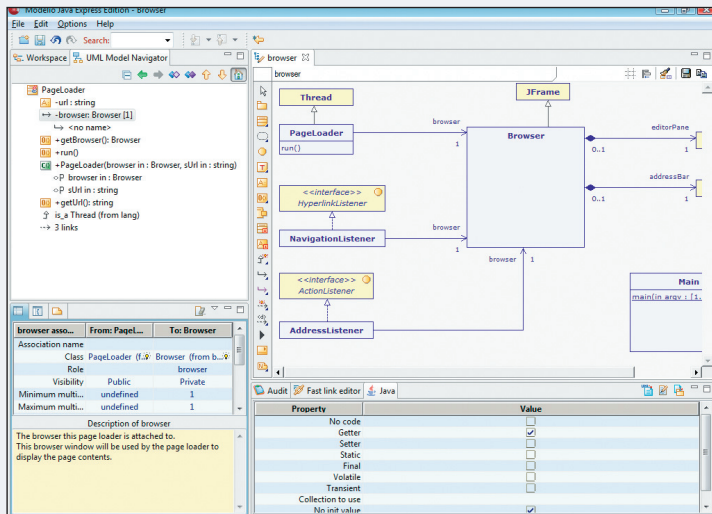


Modelio : une offre de modélisation unique

Libérez la vraie puissance de vos modèles !

UML, BPMN, Exigences ..., MDA, génération de code ...

- Modélisation intégrée de UML2, BPMN, SysML, l'Architecture d'Entreprise, les exigences, le dictionnaire, ... dans un seul référentiel
- Génération Java, C#, C++, SQL, XML, XSD, BPEL, WSDL, Hibernate...
- MDA simple et puissant - transformation, extensibilité et adaptabilité
- Travail de groupe distribué, intégré à SVN/Subversion
- Ergonomie simple, productive et familière aux développeurs (RCP/Eclipse)



Modelio est disponible en deux éditions

- **Modelio Free Edition** : Un outil de modélisation UML2, BPMN, d'Architecture d'Entreprise et de développement MDA entièrement gratuit !
- **Modelio Enterprise Edition** : La solution de modélisation et de développement MDA complète, supportant le travail en groupe et distribué, extensible avec une riche palette de modules de modélisation et de génération de code disponibles sur étagère



sales@modeliosoft.com - Tél. : 01 30 12 18 40

**Téléchargez et essayez Modelio Enterprise Edition
gratuitement pendant 10 jours !**

www.modeliosoft.com

Au cœur de SharePoint

Discret mais terriblement efficace, SharePoint est un véritable cheval de Troie pour Microsoft au cœur du système d'information des entreprises. Qu'est-ce que SharePoint ? A la base, c'est à la fois une plate-forme collaborative, de partage documentaire, un portail intranet et extranet, portail applicatif, une plate-forme de développement, plate-forme s'intégrant avec d'autres outils et serveurs Microsoft. SharePoint 2010 peut se résumer ainsi : sites web, réseau social d'entreprise, recherche, contenu, décisionnel, applications composites. Vous l'aurez compris, définir simplement SharePoint est impossible.

Cette puissance fonctionnelle ne sert à rien si vous ne pouvez l'exploiter et l'utiliser. Car SharePoint ne s'installe pas comme une suite bureautique ou un jeu. Le déploiement peut se révéler complexe, et son administration nécessite une solide expérience. Mais ensuite, « que du bonheur » ! Pour le développeur, SharePoint se révèle d'une puissance impressionnante. Désormais parfaitement intégré au modèle de développement .Net / Visual Studio, nous ne sommes pas dépayés. Vous pourrez rapidement créer des applications SharePoint avec les templates projets VS 2010. Et la plateforme supporte de nombreux frameworks et services : Linq, OData Services, .Net, COM, Web Parts, Silverlight.

Dans notre dossier, nous allons plonger au cœur de SharePoint, pour découvrir quelques-unes de ses nombreuses facettes : les outils de développement, modèle de développement avec VS 2010, comment utiliser des sources de données externes, interfacier SharePoint avec Powershell. Et ce n'est qu'un début !

François Tonic

Visual Studio 2010 et SharePoint 2010 : le couple détonnant !

Dans sa dernière mouture, Visual Studio intègre nativement les outils de développement pour SharePoint 2010. Fini les extensions et les outils tiers à ajouter pour arriver à un environnement de travail fonctionnel ! Visual Studio propose dorénavant tout l'attirail nécessaire pour réaliser des applications de manière simple et efficace. Cet article va vous présenter les principales nouveautés qu'apporte Visual Studio 2010 pour améliorer la vie du développeur SharePoint.

Comme pour n'importe quel projet .NET, le développement commence par le choix d'un modèle de projet, et SharePoint ne déroge pas à la règle ! On en retrouve ainsi dix, du projet vide à la définition de site, en passant par les workflows, les types de contenu, la Web Part « visuelle » ou encore le modèle pour les Business Connectivity Services. Vous disposerez aussi de deux modèles un peu particuliers pour importer des solutions existantes [Fig.1].

Après avoir choisi le modèle, vous passerez par une nouvelle étape intermédiaire vous demandant de spécifier l'URL du site local qui sera utilisé pour le débogage. De plus, selon le type de projet sélectionné, vous aurez la possibilité de choisir entre le déploiement en tant que solution de ferme (similaire à ce qu'on connaissait avec la version 2007) ou solution utilisateur (« sandbox »), une des nouveautés autorisant les développements au niveau des collections de site. Le projet créé est assez classique à l'exception notable de deux dossiers spéciaux « features » et « package ». Ceux-ci correspondent aux fonctionnalités et à la solution SharePoint (le fichier WSP) que nous verrons plus en détail dans la suite de

l'article. Les assemblages SharePoint sont évidemment déjà référencés et la DLL signée afin de pouvoir la déployer dans le Global Assembly Cache si besoin [Fig.2].

A ce sujet, et comme c'est un point souvent sensible dans les développements, il vous est possible de choisir la cible de l'assembly entre le répertoire « bin » de l'application Web ou le GAC, voire de ne pas l'embarquer dans le package, grâce aux propriétés du projet, ainsi que l'URL du site ou le mode « sandbox ». Ce genre de détail est significatif du soin apporté par les équipes de Redmond à la personnalisation du projet et à la volonté de nous laisser la main quand le besoin s'en fait sentir [Fig.3].

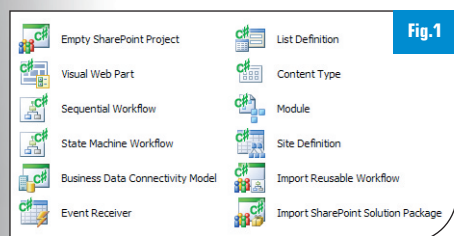
Composer avec les bons éléments

Vient ensuite la phase d'ajout des nouveaux éléments. Vous aurez le choix parmi les grands classiques que sont les WebParts, les événements ou les modèles de liste, mais aussi de nouveaux tels que les contrôles utilisateurs, les pages d'application ou la WebPart « visuelle ». Ces derniers étaient pour autant souvent utilisés dans le développement d'application, mais n'étaient

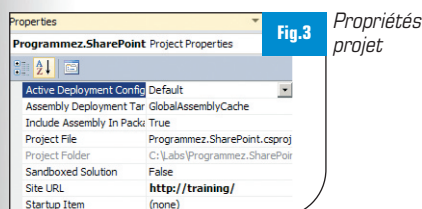
pas intégrés et nécessitaient quelques manipulations supplémentaires. Par exemple, la WebPart « visuelle », qui est une WebPart classique chargeant un contrôle utilisateur, existait dans une version très proche avec le projet CodePlex « SmartPart » (<http://www.codeplex.com/smartpart>). Vous pourrez maintenant profiter du designer Web, de la prévisualisation et de votre boîte à outils : fini les tracasseries pour réaliser des interfaces plus complexes ! [Fig.4]. Des assistants sont disponibles pour certains types d'éléments pour en simplifier l'initialisation, ainsi que certains comportements automatisés. Par exemple, le référencement automatique des fichiers dans un module (système de chargement de fichiers dans SharePoint) peut se faire par simple drag n'drop. Un gain de temps appréciable lorsque l'on charge de nombreuses images et feuilles de styles dans son site, un scénario régulièrement abordé quand on cherche à intégrer une charte graphique ! [Fig.5].

Les fonctionnalités, la base du développement SharePoint

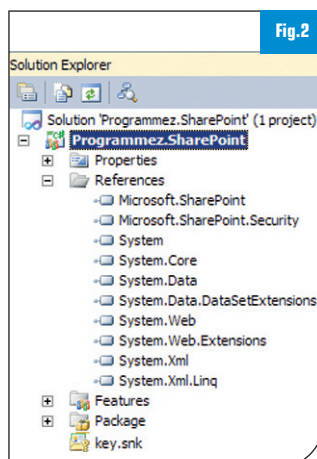
Tout ou presque dans SharePoint peut se résumer en termes de fonctionnalités (« fea-



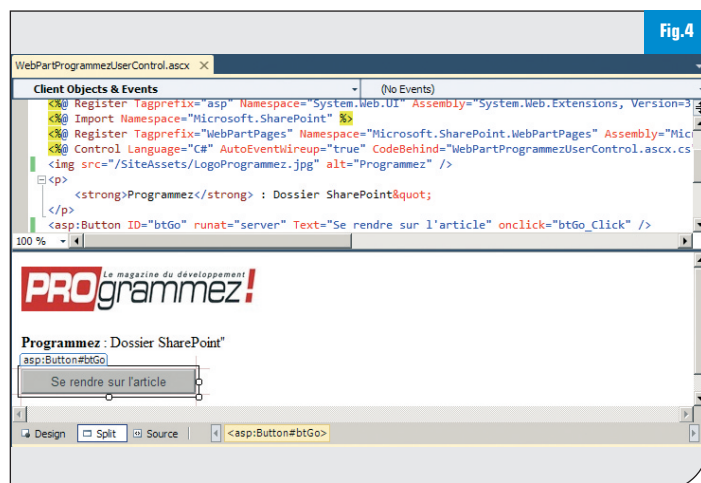
Modèles de projet



Propriétés projet



Contenu du projet





tures ») : ce sont des sortes de modules activables dans SharePoint, décrits en XML, et étendant les capacités des sites : nouvelles WebParts, modèles de workflows, type de contenu etc.

Comme expliqué précédemment, un dossier spécial « Features » existe dans tous les projets SharePoint et sera le réceptacle de toutes les fonctionnalités qui seront créées, soit automatiquement suite à l'ajout d'un nouvel élément de projet, soit manuellement par vos soins.

Un assistant intégré et la grille des propriétés vous permettent de saisir les informations telles que son titre, sa description ou la portée de son activation. Vous choisirez aussi parmi tous les éléments du projet courant lesquels seront inclus dans la fonctionnalité. Visual Studio s'occupera quant à lui de détecter les éventuelles incohérences en vérifiant que le type d'élément corresponde bien à la portée de la fonctionnalité choisie afin d'éviter des problèmes lors de l'activation de celle-ci.

Pour les besoins plus avancés, un onglet « Manifest » offre la consultation du code XML qui sera généré in fine : il correspond aux réglages effectués au travers de l'interface, mais vous pourrez aussi le personnaliser complètement en l'enrichissant de vos propres balises XML, vous laissant totale-

ment maître du résultat final [Fig.6]. L'ajout d'un événement, pour réagir sur l'activation ou la désactivation par exemple, est disponible via un menu contextuel qui générera son squelette et le déclarera dans la définition de la fonctionnalité. Vous n'aurez plus alors qu'à vous occuper de l'implémentation sans vous soucier du reste. Ce type de menu est généralisé et facilite une fois encore les opérations fréquentes en ayant tout à portée de clic.

Allo, le « 14 » ?

Au-delà des fonctionnalités, il est fréquent de vouloir déployer des fichiers dans la structure de fichiers « systèmes » de SharePoint. Le répertoire principal est quelque peu caché dans les « Program Files » et porte le nom correspondant au numéro de version de SharePoint. Petite anecdote, les versions ont sauté le nombre 13, bien connu des superstitieux !

Ce répertoire comporte de nombreux sous-dossiers dont certains servent à héberger des Web Services, des images ou des pages d'application partagées par tous les sites ou encore les ressources de localisation. Bref, il est fréquent de vouloir y placer ses propres fichiers, et pour cela Visual Studio vous propose de mapper le répertoire cible (Layouts, Images, ou celui de votre

choix) à un répertoire local de votre projet. Tous les fichiers ajoutés seront alors déployés en suivant cette correspondance : simple, net et précis ! [Fig.7].

Génération de vos livrables

Maintenant que vous avez réalisé vos développements, il vous faudra les packager afin de les installer sur les environnements SharePoint. Pour cela, SharePoint dispose d'un système de déploiement automatisé utilisant en entrée des fichiers « .wsp ». Ces derniers sont des archives compressées contenant les éléments (dll, fonctionnalités, définitions de sites...) et les instructions pour leur déploiement.

Chaque projet SharePoint vous propose un assistant vous permettant de choisir parmi les différentes fonctionnalités de la solution Visual Studio celles qui feront partie du package. Il sera également possible d'ajouter des assemblés externes dans le cas où vous intégrez certains frameworks ou composants tiers. Et l'onglet « Manifest » vous permettra comme pour les fonctionnalités de voir le fichier XML définissant la structure de l'archive et de l'éditer si besoin.

Pour en savoir plus sur ces packages, vous pouvez consulter les articles relatifs sur MSDN : <http://msdn.microsoft.com/fr-fr/library/ee231560.aspx>.

Gestion de module

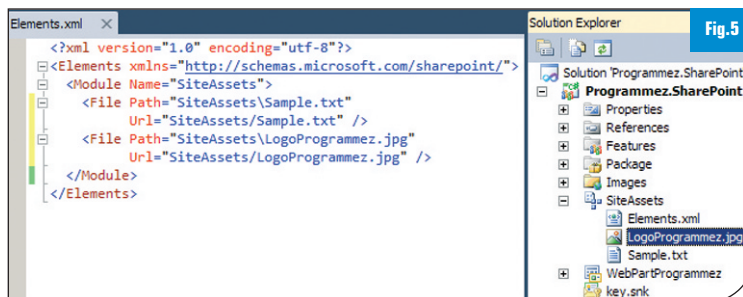


Fig.5

Assistant Fonctionnalité

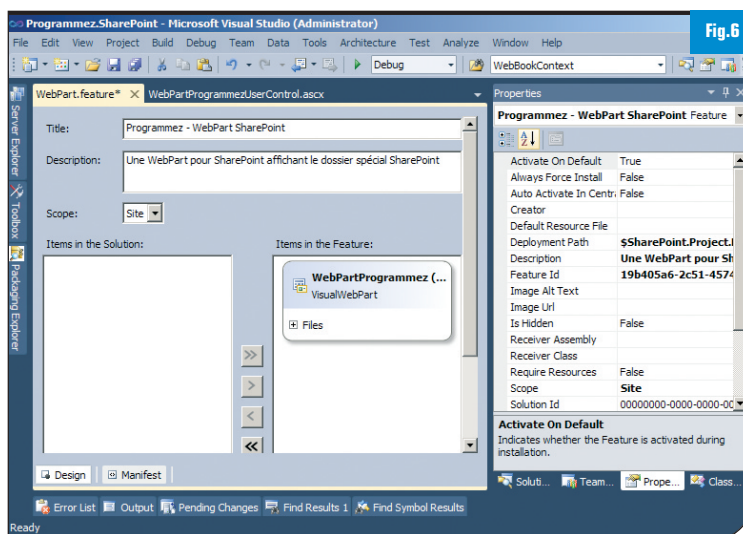


Fig.6

Et maintenant, on teste !

Vous avez écrit vos premières lignes de code et vous voulez voir ce que cela donne sur votre site SharePoint ? Facile : posez un point d'arrêt et appuyez sur « F5 » : Visual Studio se chargera de packager, déployer vos développements (en prenant soin de retirer préalablement les anciennes versions en cas de mise à jour), activer les fonctionnalités pour finir par lancer le navi-

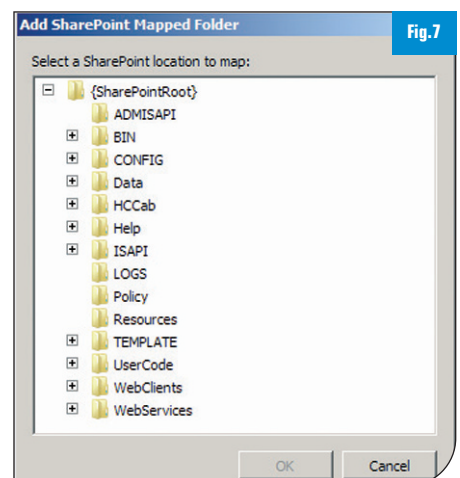


Fig.7

Répertoires mappés

gateur pour vous rendre sur la page d'accueil du site spécifié lors de la création du projet. Visual Studio s'attache automatiquement au processus, que ce soit celui de IIS (« w3wp.exe ») ou du service du code utilisateur (« SPUCWorkerProcess.exe ») selon le type de déploiement sélectionné.

Les étapes du déploiement peuvent être personnalisées via l'onglet « SharePoint » présent dans les propriétés du projet grâce à des configurations. Vous pourrez ainsi choisir quelles sont les opérations qui seront effectuées lors du déploiement et du retrait du package : exécution d'un script pré ou post déploiement (similaire au système pour commandes pré/post compilation), recyclage des pools d'application IIS, ajout / retrait de la solution, activation des fonctionnalités. Et si cela ne vous suffit toujours pas, ces étapes sont extensibles par programmation avec les projets VSIX (<http://msdn.microsoft.com/en-us/library/ee256698.aspx>) [Fig.8].

Et côté développement pur ?

SharePoint a aussi évolué au niveau de son API et de l'outillage pour interagir avec le contenu de vos sites. Nous retrouvons ainsi notre langage de requête bien connu en .NET dans sa déclinaison pour SharePoint : LINQ to SharePoint.

Vous allez pouvoir profiter de classes fortement typées pour manipuler vos éléments de liste SharePoint. Il suffit de générer ces entités à l'aide de l'utilitaire SPMetal présent dans le répertoire BIN du « 14 » en spéci-

fiant l'URL du site, le fichier de sortie et l'espace de nom comme ceci : SPMetal /web: <http://localhost/namespace:SP.Linq/code:TeamSite.cs>

Vous trouverez toutes les options pour SPMetal sur la page suivante : <http://msdn.microsoft.com/fr-fr/library/ee538255.aspx>. Ajoutez ensuite ce fichier à votre projet et la référence à l'assembly « Microsoft.SharePoint.Linq.dll ». Vous pouvez maintenant effectuer vos requêtes sans vous soucier du CAML (« Collaborative Application Markup Language »), ce langage XML décrivant les structures SharePoint et les requêtes sur le contenu. Cependant, ne faites pas encore totalement une croix dessus, certains scénarii d'usage nécessiteront encore son utilisation, LINQ to SharePoint vous servant de surcouche dans le reste des cas [Fig.9].

Dans le même ordre d'idée, SharePoint supporte ADO.NET Data Services et propose un service Web REST afin de consommer les listes SharePoint dans les applications clientes. C'est d'ailleurs un tout nouveau pan qui s'ouvre aux développeurs grâce au nouveau modèle objet client qui offre une alternative plus souple et plus normée aux services web qui n'avaient pas beaucoup évolué depuis les dernières versions. Vous avez désormais le choix parmi trois types d'API : Javascript, .NET et une dédiée pour Silverlight ! Le modèle objet ainsi exposé ressemble fort à celui que vous utilisez côté serveur pour assurer une meilleure cohérence et une montée en compétence facilitée.

Une âme d'explorateur

Pendant le développement il est fréquent de vouloir connaître la valeur d'une propriété d'un élément de SharePoint tel qu'un

identifiant de liste ou un type de contenu, le nom interne d'une colonne ou encore les fonctionnalités activées. Et bien l'explorateur de serveur s'est vu enrichi de la possibilité d'ajouter des connexions à des sites SharePoint. Vous pourrez ensuite vous balader dans la structure du site pour afficher sa structure mais aussi pour consulter dans la fenêtre des propriétés les différentes valeurs de l'élément que vous aurez sélectionné [Fig.10].

De quoi tester plus rapidement vos développements en ayant une idée précise du contenu que vous êtes en train d'utiliser.

Et depuis le Service Pack 1

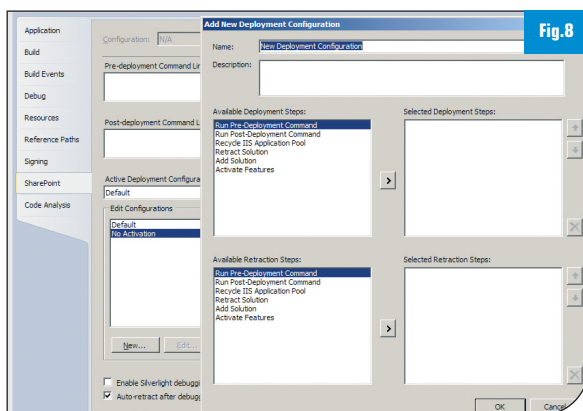
Microsoft continue de vouloir proposer des outils les plus à jour possibles, et le tout récent Service Pack 1 de Visual Studio a apporté encore quelques fonctionnalités très attendues.

SharePoint 2010 étant en 64-bit et ciblant le framework .NET 3.5, il n'avait pas eu droit au support des tests unitaires et d'intégration ni à l'IntelliTrace, le débogage historique vous permettant de sauvegarder une session de debug et de naviguer à l'intérieur après coup. C'est désormais corrigé, et vous n'avez plus de raison de ne pas mettre en place un ALM (« Application Lifecycle Management ») dans vos développements, surtout en profitant des interactions avec Team Foundation Server 2010.

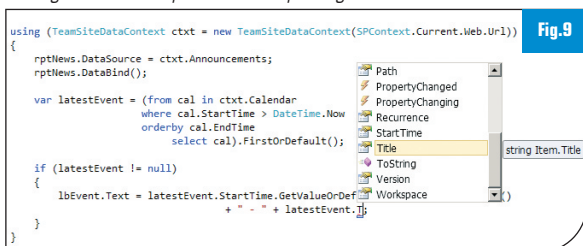
Pour conclure

Fervent utilisateur des outils tiers lors de mes développements sur SharePoint 2007, notamment de WSPBuilder (<http://wspbuilder.codeplex.com>), j'avoue que j'attendais Visual Studio au tournant. Et passé le cap de la découverte, une fois pris ses marques il est difficile de revenir en arrière. C'est un constat que je retrouve chez d'autres personnes, et j'espère vous avoir donné envie d'en savoir plus et de vous plonger dans le développement SharePoint 2010 à l'aide de Visual Studio 2010. Vous bénéficierez à la fois d'assistants et d'une simplification générale de la conception de son application, tout en gardant la possibilité de lever le capot et d'aller personnaliser la micro feature quand le cas se pose.

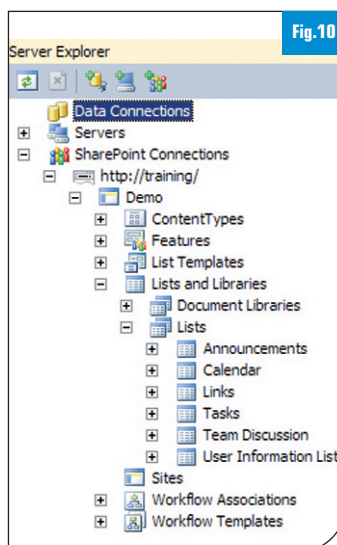
■ Gaëtan Bouveret – Consultant /
Formateur SharePoint
MVP SharePoint Server
Access It IDF / gbouveret@access-it.fr
Blog : <http://www.sharepointofview.fr/gat>



Configuration du déploiement du package



LINQ to SharePoint



Explorateur de connexions SharePoint



Les solutions **sandbox** : le développement sans empreinte serveur

Les solutions bac à sable (sandbox) sont une des principales nouveautés de la version 2010 de SharePoint. Elles permettent à des administrateurs de collection de sites de déployer eux-mêmes des développements spécifiques sur leurs sites.

En pratique, le code serveur des solutions sandbox est exécuté dans un processus séparé de celui du serveur web, qui est sécurisé, surveillé, et disposant de privilèges très limités. Le code d'une solution sandbox est donc considéré comme sans danger pour l'environnement SharePoint hôte, c'est pourquoi un administrateur d'une collection de sites peut déployer ce type de solution sans nécessiter l'intervention d'un administrateur de la ferme. D'ailleurs dans Visual Studio 2010, lors de la création d'un projet SharePoint 2010, le mode sandbox est l'option par défaut et celui qui est recommandé.

Nous allons voir aujourd'hui comment développer un composant web part sous forme d'une solution sandbox, puis, après avoir discuté des limitations liées à ce type de développement, nous verrons comment permettre l'exécution de code à privilèges élevés depuis du code sandbox.

Développement d'un composant web part

Dans Visual Studio, créer un projet de type **Projet SharePoint** vide appelé DemoSandbox [Fig.1]. L'écran suivant [Fig.4] propose de saisir l'URL du site de test. Pour les besoins de ce tutorial, il convient de saisir l'URL d'un site d'équipe. Cet écran permet aussi de sélectionner le mode de déploiement de la solution SharePoint générée par ce projet, sélectionner bac à sable.

Une fois le projet créé, ajouter un élément de type **Composant Web-Part** appelé DemoWebPart. Ecrire le code dans le corps du composant web part : [Fig.2].

En appuyant sur F5, le projet est automatiquement compilé, packagé sous forme d'une solution sandbox, et celle-ci est déployée et activée dans la galerie des solutions de la collection de sites cible. Un habitué du développement SharePoint remarquera que l'ouverture du site cible s'est faite quasi instantanément. En effet, l'architecture des solutions sandbox fait qu'il n'est pas nécessaire de redémarrer IIS après un déploiement. On peut ensuite placer le composant web

DemoWebPart
Bonjour VAD-DEV/administrator, voici les pages du site courant :
• Accueil.aspx
• Utiliser cette bibliothèque.aspx

Fig.3

Rendu de DemoWebPart

part sur la page d'accueil, le résultat devrait ressembler à ceci : [Fig.3].

Satisfait par cette web part, voyons maintenant comment déployer manuellement notre solution sandbox sur une autre collection de sites, comme on le ferait par exemple pour déployer sur un environnement de production. La première étape consiste à récupérer le fichier .wsp dans le répertoire bin\Debug ou bin\Release du projet Visual Studio. On notera que l'extension est la même que pour un développement de type ferme. Pour déployer une solution sandbox, il est nécessaire de se connecter à la collection de sites cible avec un compte qui en est administrateur. On peut alors naviguer jusqu'à la galerie des solutions : depuis une page du site racine de la collection de sites, via le menu **Actions du site**, cliquer sur **Paramètres du site**, puis dans la section Galeries sur le lien **Solutions** [Fig.4].

Cet écran permet d'installer et d'activer des solutions sandbox sur la collection de sites, ainsi que de visualiser le quota actuel de consommation de ressources. Via le ruban, cliquer sur l'onglet **Solutions** puis sur le bouton afin d'uploader le package dans la collection de

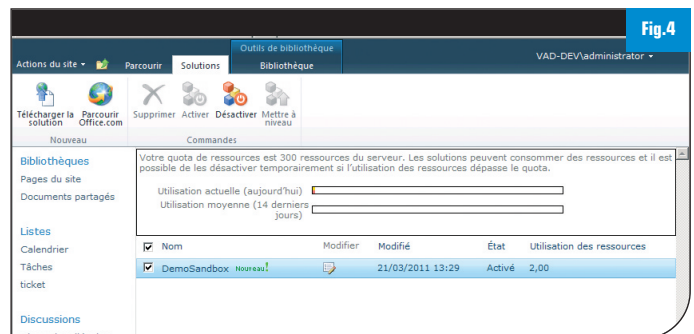


Fig.4

Ecran d'administration des solutions sandbox

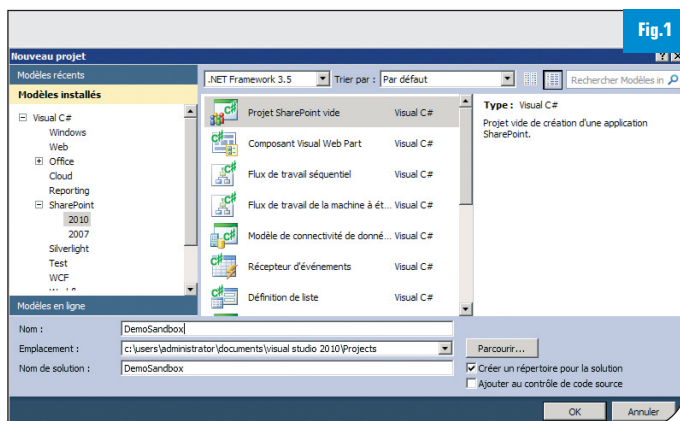


Fig.1

Création du projet

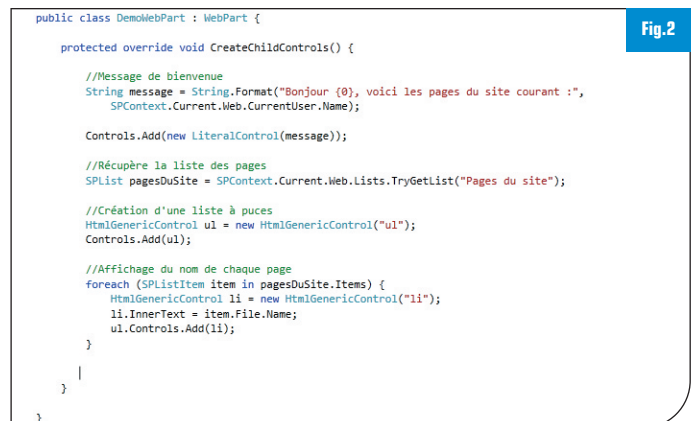


Fig.2

DemoWebPart.cs

sites. Il faut ensuite l'activer, toujours via le ruban, pour rendre disponible les fonctionnalités du package. Les fonctionnalités de scope collection de sites seront automatiquement activées lors de l'activation du package. Celle contenant notre web part, par exemple, est de scope collection de sites.

Les limitations des solutions sandbox

Les solutions sandbox permettent de réaliser un grand nombre de développements, mais pas tous. Elles ont certaines particularités conduisant à des limitations :

- aucun fichier ne peut être déposé dans le répertoire d'installation de SharePoint. Pour le développeur, cela veut dire que dans le projet Visual Studio on ne peut pas utiliser les répertoires mappés (correspondant aux répertoires `/_layouts`, `/_images`, etc.).
- le code .NET est isolé dans un processus dédié, surveillé, et dont les privilèges sont limités
- l'assembly `Microsoft.SharePoint.dll` est remplacée à l'exécution par une copie ne contenant qu'un sous-ensemble des classes et méthodes de l'API standard.

Ces caractéristiques rendent certains développements irréalisables. Un exemple intéressant est le Composant Web Part Visuel, similaire au composant Web Part classique à la différence que le composant principal est un `UserControl (.ascx)` ASP.NET. Ce type de web part nécessite que le fichier `.ascx` soit déployé dans le répertoire `14\LAYOITS\CONTROLTEMPLATES` du serveur SharePoint, chose qui est par définition impossible avec une solution sandbox.

Toutefois, dans ce cas précis, un contournement est possible. Microsoft propose une extension Visual studio 2010 SharePoint Power Tools, qui propose un nouvel élément de projet SharePoint : le composant Web Part Visuel pour sandbox. Cet élément de projet ressemble à un composant Web Part Visuel, mais le fichier `.ascx` n'est pas inclus dans le package : au moment de la compilation, ce dernier est supprimé et remplacé par un contrôle standard écrit en C# dont tous les sous-contrôles sont instanciés et initialisés par code, exactement comme défini dans le fichier `ascx`. Le `UserControl` est ainsi rendu « compatible » sandbox de manière transparente pour le développeur.

Parmi les autres types de développements habituels qui ne sont pas supportés dans les solutions sandbox, on trouvera les contrôles .NET qui ne sont pas des web parts (ceux référencés dans les master pages par exemple), les `DelegateControls`, les tâches planifiées, les workflows .NET, etc. D'une manière générale, on peut considérer que tout développement dont le périmètre dépasse la collection de sites ne peut pas être réalisé en solution sandbox.

D'autre part, le code sandbox ne dispose que d'un sous-ensemble de l'API standard. Il est par exemple impossible d'instancier un objet `SPSite` (un des objets principaux) à partir de son url comme on pourrait le faire dans une solution de type ferme. Il est toutefois possible de manipuler celui exposé au travers du contexte, via la propriété `SPContext.Current.Site` comme nous l'avons fait dans la classe `DemoWebPart`. Cela est logique car par définition, une solution sandbox n'est disponible que dans le cadre de la collection de sites dans laquelle elle est activée.

Dans les solutions sandbox, une autre différence intéressante est la manière d'enregistrer les récepteurs d'événements. Dans une solution classique, on pourrait le faire par code grâce à la méthode `list.EventReceivers.Add()`. En raison de l'architecture d'exécution plus complexe, cette méthode ne fait pas partie de l'API disponible

en mode bac à sable. Pour fonctionner, les `EventReceivers` doivent être inclus à des fonctionnalités de manière déclaratives, en XML.

La dernière grande limitation, en termes de code, concerne les ressources serveur (réseaux, disque, etc.) dont l'accès est interdit au processus exécutant le code sandbox par Code Access Security (CAS). Par défaut, dans du code sandbox, il est par exemple impossible d'accéder à un fichier sur le disque dur du serveur ou d'effectuer un appel à un web service. Le processus qui exécute le code sandbox n'en a pas les permissions.

Un mécanisme est cependant prévu pour permettre, depuis du code sandbox, d'exécuter du code avec des privilèges élevés. Ce mécanisme porte le nom de full-trust proxy.

Développement d'une opération full-trust proxy

Une opération full-trust proxy est une classe héritant de `SPPProxyOperation` packagée dans une solution de type ferme. Une fois la solution déployée sur la ferme et l'opération enregistrée, un exécutable sandbox peut appeler l'opération grâce à la méthode `SPUtility.ExecuteRegisteredProxyOperation(ass, type, args)`. Celle-ci prend en argument le nom complet de l'assembly contenant la méthode, le nom complet de sa classe, et un objet héritant de `SPPProxyOperationArgs`. Ce dernier est un type référencé à la fois par le proxy et le code sandbox, et permet de passer des arguments à l'opération.

Une architecture avec full-trust proxy comporte en principe 3 projets : le projet contenant l'opération, le projet sandbox qui consomme l'opération, et le projet qui contient la classe d'argument (une assembly commune référencée par les 2 premiers projets).

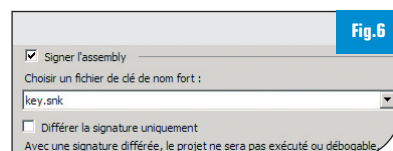
Pour l'exemple, nous allons développer un composant web part sandbox proposant un bouton permettant d'envoyer un mail à l'utilisateur courant. Un envoi de mail n'étant pas permis dans du code sandbox, nous allons devoir utiliser une opération Full-Trust Proxy.

Développement de la classe d'arguments

Créer un projet intitulé `DemoArgs`, de type librairie de classe en Framework .NET 3.5. Dans ce projet, ajouter une classe `MailProxyOperationArgs` contenant le code : [Fig.5].

Cette classe permettra au composant web part de passer les arguments nécessaires à l'appel de l'opération d'envoi d'email.

Puisque ce projet sera référencé par un projet déployé dans le GAC, il est nécessaire de le signer. Dans les propriétés du projet, dans la section signature, cocher la case « Signer l'assembly » et indiquer un nom pour la clé [Fig.6].



Signature de l'assembly



MailOperationArgs.cs



Développement de l'opération full-trust proxy

Ensuite, dans la même solution, créer un projet intitulé DemoMailOperation, de type projet SharePoint vide en mode Ferme. Dans ce projet, référencer le projet DemoOperationArgs puis créer une classe MailProxyOperation contenant le code : [Fig.7].

Ceci est la méthode full-trust qui sera appelée par le composant web part. Il est nécessaire de déclarer cette méthode par code afin qu'elle soit rendue disponible. Pour ce faire, dans le même projet, créer une feature RegisterOperation de scope Ferme, et y associer un récepteur d'événements de feature (via un clic droit sur la feature) contenant le code : [Fig.8]. L'activation de cette feature permettra ainsi l'enregistrement de notre opération.

Appel de l'opération depuis le code sandbox

Enfin, dans le projet DemoSandbox, référencer le projet DemoOperationArgs et ajouter un composant web part. Y écrire le code suivant : [Fig.9].

On prendra soin avant de déployer de récupérer le nom complet de l'assembly correspondant au projet DemoMailOperationArgs et de le copier dans la classe MailProxyOperationArgs, dans la propriété ProxyOperationAssembly. Conceptuellement, il est pratique que les noms d'assembly et de classe du proxy soient fournis dans la classe d'arguments, puisque celle-ci est obligatoirement référencée par les projets sandbox qui utilisent l'opération associée.

Une fois les 2 solutions déployées, si la fonctionnalité RegisterOpe-

ration est activée par l'administrateur au niveau de la ferme, le composant web part devrait envoyer un mail à l'utilisateur lors du clic sur le bouton [Fig.10].

Conclusion

Avec SharePoint 2010, les solutions sandbox sont le mode de développement à privilégier, et vraisemblablement, le seul qui sera supporté sur Office 365. Attention toutefois aux performances puisque l'exécution du code dans un thread séparé a un coût. Dans certains cas, par exemple pour un composant web part exposé sur la page d'un site internet, il peut être préférable d'utiliser une solution de type ferme même si le développement est réalisable en mode Sandbox. On peut voir le mode sandbox comme un moyen simple et fiable de déployer des développements spécifiques sur des environnements SharePoint mutualisés, dans lesquels le déploiement de solutions de type ferme est contraignant et conduit à une procédure de validation parfois longue.

■ Arnault Nouvel - MVP SharePoint Server

Société Winwise : <http://www.winwise.fr>

Blog : <http://blogs.developpeur.org/anouvel>

```
namespace DemoMailOperation {
    public class MailOperation : SPProxyOperation {
        public override object Execute(SPProxyOperationArgs args) {
            try {
                MailOperationArgs mailArgs = args as MailOperationArgs;

                using (SPSite site = new SPSite(mailArgs.SiteId)) {
                    using (SPWeb web = site.OpenWeb(mailArgs.WebId)) {
                        SPUtility.SendEmail(web, true, false, mailArgs.To,
                            mailArgs.Subject, mailArgs.Body);
                    }
                }

                return "Email envoyé";
            } catch (Exception ex) {
                return "Error : " + ex.Message;
            }
        }
    }
}
```

Fig.7

MailOperation.cs

```
public class MailWebPart : WebPart {
    protected override void CreateChildControls() {
        Button btnSendMail = new Button();
        btnSendMail.Text = "Envoyer moi un mail";
        btnSendMail.Click += new EventHandler(btnSendMail_Click);
        Controls.Add(btnSendMail);
    }

    void btnSendMail_Click(object sender, EventArgs e) {
        //Préparation des arguments à envoyer à l'opération
        MailOperationArgs args = new MailOperationArgs() {
            SiteId = SPContext.Current.Site.ID,
            WebId = SPContext.Current.Web.ID,
            To = SPContext.Current.Web.CurrentUser.Email,
            Subject = SPContext.Current.Web.Title,
            Body = "Bonjour depuis le bac à sable SharePoint !"
        };

        //Exécution de l'opération avec des droits élevés
        Object res = SPUtility.ExecuteRegisteredProxyOperation(
            MailOperationArgs.ProxyOperationAssembly,
            MailOperationArgs.ProxyOperationClass, args);

        //Affichage de la valeur de retour
        Controls.Add(new LiteralControl("<div>" + res.ToString() + "</div>"));
    }
}
```

Fig.9

MailWebPart.cs

```
public class RegisterOperationEventReceiver : SPFeatureReceiver {
    /// <summary> ...
    public override void FeatureActivated(SPFeatureReceiverProperties properties) {
        //service sandbox
        SPUserCodeService userCodeService = SPUserCodeService.Local;

        SPProxyOperationType proxyOperation = new SPProxyOperationType(
            MailOperationArgs.ProxyOperationAssembly,
            MailOperationArgs.ProxyOperationClass);

        //Enregistrement de l'opération
        userCodeService.ProxyOperationTypes.Add(proxyOperation);
        userCodeService.Update();
    }

    /// <summary> ...
    public override void FeatureDeactivating(SPFeatureReceiverProperties properties) {
        SPUserCodeService userCodeService = SPUserCodeService.Local;

        //Récupération de l'opération
        SPProxyOperationType proxyOperation = null;
        foreach (SPProxyOperationType operationType in userCodeService.ProxyOperationTypes)
            if (operationType.AssemblyName == MailOperationArgs.ProxyOperationAssembly
                && operationType.TypeName == MailOperationArgs.ProxyOperationClass) {
                proxyOperation = operationType;
                break;
            }

        //Désenregistrement de l'opération
        if (proxyOperation != null) {
            userCodeService.ProxyOperationTypes.Remove(proxyOperation);
            userCodeService.Update();
        }
    }
}
```

Fig.8

RegisterOperation.EventReceiver.cs

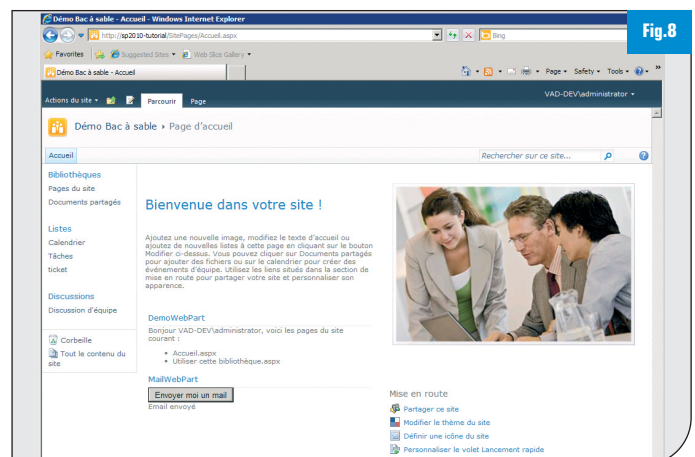


Fig.10

Connecter SharePoint 2010 à une source de données externe

Depuis la version 2007, SharePoint possède une fonctionnalité particulièrement intéressante, celle de pouvoir se connecter simplement à des sources de données externes. Dans la version 2007, elle s'appelait *Business Data Catalog*. Elle était disponible dans la version *MOSS Enterprise* (et non dans la plateforme gratuite *WSS 3.0*). Pour cette nouvelle version de SharePoint, les conditions ont changé : en effet, dès la version de base (qui ne s'appelle plus *WSS*, mais *SharePoint Foundation*) nous retrouvons la version 2.0 des *Business Data Catalog*. Cette fonctionnalité, ou service applicatif, se nomme dorénavant *Business Connectivity Services* (BCS).

Nous trouvons différentes couches :

- Le système externe : source de données externes à laquelle SharePoint se connecte,
- Type de contenu externe : entité métier représentant une donnée de ce système (un client, un produit etc.) qui contient des propriétés et des actions,
- Liste externe : liste SharePoint affichant ces données métiers sur une page.

Nous allons étudier dans cet article les différents connecteurs fournis en standard et comment les utiliser. Pour des scénarii plus complexes, nous verrons comment développer notre propre connecteur. Enfin, nous terminerons par l'utilisation du moteur de recherche SharePoint avec ce service de *BCS* afin de pouvoir rechercher dans ces données externes.

Les connecteurs en standard

SharePoint Foundation 2010 propose deux connecteurs en standard, permettant de se connecter simplement et rapidement à un système externe : le connecteur *SQL Server* et le connecteur *service web*. Grâce au connecteur *SQL Server*, nous pourrions affi-

cher dans une liste SharePoint des données provenant directement d'une table SQL, stockée dans une base de données et hébergée sur un serveur *Microsoft SQL Server* (uniquement). Partons d'un exemple concret : dans la capture nous avons une table SQL contenant une liste de livres [Fig.1]. Pour nous aider dans cette tâche, nous allons nous appuyer sur *SharePoint Designer 2010*. Cet outil, gratuit, va justement nous permettre de se connecter à cette source de données, sans fournir la moindre ligne de code. Lançons donc *SharePoint Designer*, et ouvrons notre site SharePoint : [Fig.2].

Depuis la navigation de gauche, naviguons jusqu'à la section « Type de contenu externe », et créons-en un (attention, le bouton se trouve en haut à gauche de l'interface). Nommons-le « Livres » [Fig.3].

Associations maintenant une connexion à notre source de données. Nous allons ici choisir *SQL Server*. Renseignons le serveur et la base de données à utiliser. En ce qui concerne la sécurité associée : en laissant l'option par défaut (*Se connecter avec l'utili-*

sateur courant), le compte utilisateur courant sera utilisé sur le serveur SQL. Il faudra paramétrer le service *Secure Store* de *SharePoint 2010* pour déléguer le compte utilisateur qui servira à se connecter au serveur SQL [Fig.4].

Il faut maintenant expliquer à *SharePoint* comment lire et écrire dans ce système. En faisant un clic-droit sur notre table « Livres », nous allons justement pouvoir créer toutes les méthodes *CRUD* (*Create, Read, Update et Delete*). Dans le modèle XML associé, *SharePoint* va en fait générer toutes les requêtes SQL pour interagir avec cette table : [Fig.5].

Nous pourrions ainsi créer une nouvelle liste basée sur le type de contenu externe sauvegardé (depuis le ruban de *SharePoint Designer*). Naviguons maintenant jusqu'à notre site *SharePoint* et nous obtenons bien les données provenant de notre table SQL (il faudra tout de même avoir au préalable donné les droits suffisants au niveau du service applicatif *BCS*, depuis la console centrale d'administration) [Fig.6].

Il est possible de modifier l'affichage de la



Fig.1

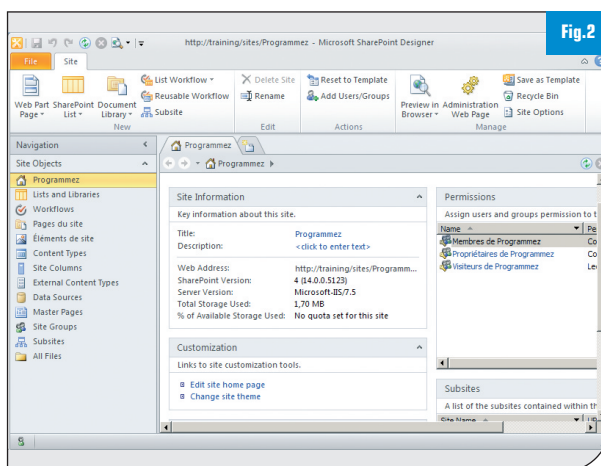


Fig.2

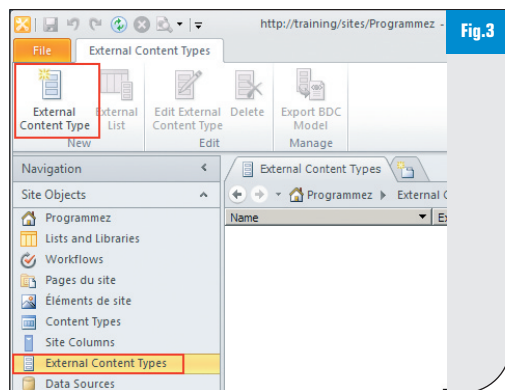


Fig.3

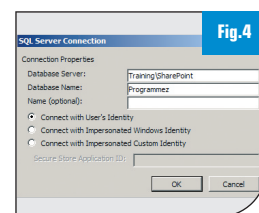


Fig.4

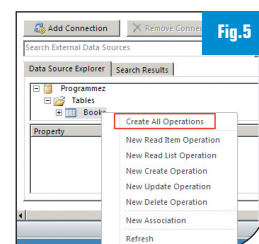


Fig.5



liste, ajouter, modifier et supprimer des éléments, bref, d'utiliser cette liste comme n'importe quelle liste *SharePoint*. La seule différence est que les données ne sont pas stockées dans *SharePoint*, mais dans notre serveur SQL. Aussi simplement que nous venons de nous connecter à une table SQL, nous pourrions nous connecter à un service web (asmx) ou WCF. Il y a cependant une contrainte : pour que *SharePoint* puisse se connecter à une source externe via les BCS, il est nécessaire que le service externe expose deux méthodes : une pour récupérer toutes les instances de l'entité, et une pour récupérer une instance d'une entité pour un identifiant donné. Ces deux méthodes, ou descripteurs, sont communément appelées *Finder* et *SpecificFinder*. D'ailleurs, en reprenant l'exemple précédent avec le connecteur SQL, si nous épluchons le modèle XML généré, nous retrouverons bien une référence à une méthode *Finder* et *SpecificFinder*. Mais comme nous avons vu, *SharePoint* s'est débrouillé tout seul pour générer les requêtes SQL. Or, dans le cas d'un connecteur vers un service web/WCF, nous allons devoir indiquer à *SharePoint* quelle est la méthode exposée par le service qui correspond au *Finder* et quelle est celle correspondant au *SpecificFinder*. Ces deux méthodes sont les deux seules nécessaires pour *SharePoint*. Si toutefois vous voulez pouvoir mettre à jour, ajouter ou supprimer une donnée externe via ce service, il faudra implémenter les méthodes associées au niveau de votre service et les lier aux descripteurs *SharePoint* correspondants. Reprenons notre jeu de données précédent, mais exposons maintenant les « Livres » via un service WCF. Ci-dessous la signature de

notre service, ainsi que le type retourné : [Fig.7].

Le type renvoyé par notre service se nomme *Book*, il expose les mêmes propriétés que notre table SQL, à savoir son numéro ISBN (notre clé ici), son titre et son auteur. Une fois de plus, *SharePoint Designer 2010* est l'outil qui va nous permettre de créer simplement notre connexion à ce service. Ouvrons donc notre site, et créons un nouveau type de contenu externe. Cette fois-ci, nous n'allons pas choisir SQL Server comme connecteur mais WCF Service (cela peut tout à fait être un service asmx). Donnons l'URL de notre service, et éventuellement les paramètres d'authentification via le *Secure Store Service*. Une fois la découverte du service faite par *SharePoint Designer*, naviguons jusqu'à notre service, et listons ses méthodes. Nous retrouvons bien ici nos deux méthodes *GetAllBooksFromCatalog* et *GetBookFromISBN*. Spécifions *GetAllBooksFromCatalog* comme étant notre méthode *ReadList*, celle qui nous renverra donc tous nos livres (dont le descripteur interne est *Finder*). Un assistant de paramétrage s'ouvre, dans lequel nous allons devoir préciser quel est l'identifiant de l'entité métier livre, ici notre propriété ISBN. Si notre service externe renvoie beaucoup d'éléments, nous pouvons via cet assistant limiter le nombre d'entités remontées. Il est également possible d'avoir plusieurs méthodes de type *ReadList*, permettant ainsi d'afficher dans des listes externes des données basées sur des critères de filtres différents. Spécifions enfin *GetBookFromISBN* comme méthode *ReadItem* (descripteur interne *SpecificFinder*), en s'appuyant sur l'assistant présenté par *SharePoint Designer* [Fig.8].

Sauvegardons le tout, et comme nous l'avons fait précédemment, créons une liste externe. Cette liste, une fois affichée dans le navigateur, nous montrera tous les livres récupérés via ce service.

Si les données externes ne sont ni stockées dans une base de données hébergée par un serveur SQL, ni exposées via un service web

(ou exposées via un service qui ne propose pas les bonnes méthodes), il est possible de développer notre propre connecteur selon deux façons :

- Le connecteur .NET
- Le connecteur personnalisé

Le connecteur .NET sera utilisé quand l'interface exposée reste statique, ou ne change pas fréquemment. En revanche, un connecteur personnalisé sera développé lorsqu'il sera nécessaire de surcharger le type ou de changer fréquemment l'interface exposée. Adaptons notre exemple : toutes les références à nos livres sont dorénavant stockées dans un système propriétaire et, fort heureusement, ce système propose une API accessible en .NET pour y accéder. Nous allons donc développer un connecteur .NET utilisant cette API. Nous n'allons plus utiliser *SharePoint Designer*, mais *Visual Studio 2010* qui possède pour le coup un modèle de projet approprié pour ce type de développement. Un package wpf sera généré et ce sera ce package qu'il faudra déployer sur les serveurs *SharePoint*. Via le designer de *Visual Studio*, nous allons pouvoir définir le type à exposer (ici, encore *Book*) et les différentes méthodes pour récupérer tous nos livres, ou un seul à partir d'un identifiant, ou pour ajouter, mettre à jour ou supprimer un livre [Fig.9].

Dans la classe *BookService.cs* nous allons trouver l'implémentation de notre logique permettant de récupérer tous les livres (*ReadList*), ou un seul d'après un identifiant (*ReadItem*) ou d'en ajouter un (*AddBook*) [Fig.10].

Nous trouverons également une classe contenant notre entité, avec les propriétés que nous attendons (ISBN, Titre, etc.). Si nous ouvrons le modèle en XML (et non dans le designer), nous pouvons constater que chacune de nos méthodes de la classe *BookService* est associée à un descripteur

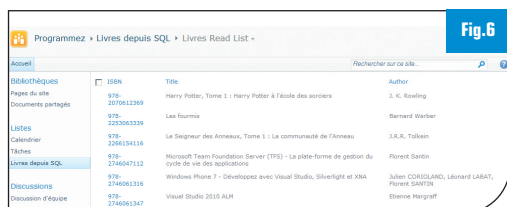


Fig.6

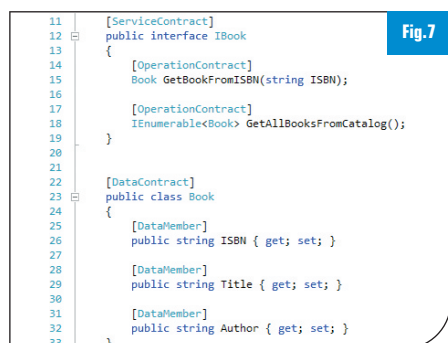


Fig.7

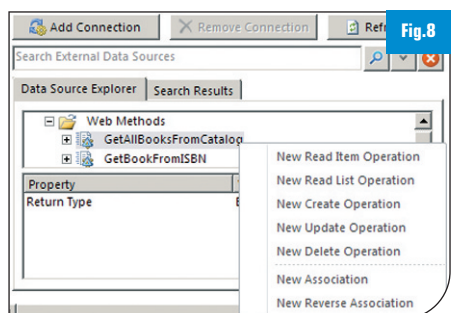


Fig.8

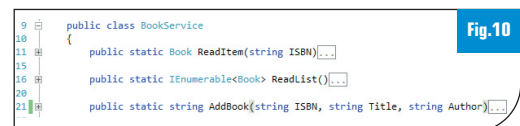


Fig.10

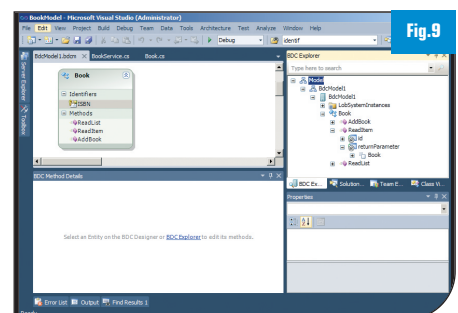


Fig.9

indiquant à SharePoint quelle est son utilité (*Finder* pour *ReadList*, *SpecificFinder* pour *ReadItem* et *Creator* pour *AddBook*). C'est ainsi que nous allons définir toutes les autres actions possibles : *Delete*, *Updater*, *StreamAccessor*, *AccessChecker*... Attention, chaque méthode que vous allez pouvoir associer devra avoir une signature particulière. En effet, la méthode associée au descripteur *SpecificFinder* doit, par exemple, prendre en paramètre un Identifiant [Fig.11].

Rechercher dans nos données métiers

Nos données métiers sont bien connectées à notre portail SharePoint, nous avons des listes externes nous permettant d'afficher ces données dans nos pages. Il ne nous manque plus qu'une chose : pouvoir indexer ce contenu dans le moteur de recherche (*SharePoint Server* uniquement) pour pouvoir effectuer une recherche sur ces données depuis notre portail. Si nous avons utilisé le connecteur Serveur SQL ou Service Web, alors aucune configuration supplémentaire du modèle n'est requise. Si toutefois un connecteur a été développé, il faudra apporter quelques modifications au modèle XML :

1 En indiquant qu'il est utilisable dans l'interface de recherche : il faudra donc ajouter la propriété « *ShowInSearchUI* » avec une valeur [peu importe d'ailleurs la valeur] au modèle : [Fig.12].

2 En spécifiant quelle méthode de type *Finder* sera utilisée par le moteur (en effet, nous avons vu que nous pouvions en avoir plusieurs). Nous devons aussi spécifier quelle propriété de l'entité contient la date de dernière modification de l'élément (nous verrons juste après l'utilité de cette propriété). Dans cet exemple, une propriété supplémentaire a été ajoutée à notre entité *Book*, *LastUpdated* : [Fig.13]

La dernière étape consiste à paramétrer le moteur de recherche pour qu'il puisse parcourir ces données externes pour les indexer. Naviguons jusqu'à notre console centrale d'administration et ajoutons une

nouvelle source de contenu à notre service de recherche. Nous allons bien entendu choisir ici « Lignes de données métiers » et sélectionner notre modèle *Livres* précédemment créé [Fig.14].

Une fois la source de données ajoutée, il ne nous reste plus qu'à indexer son contenu. *SharePoint Server 2010* propose deux modes d'indexation :

- 1 Complète : tous les contenus sont parcourus pour créer l'index,
 - 2 Incrémentale : il ne prend que le contenu modifié depuis la dernière indexation.
- Nous voyons donc ici l'utilité de la propriété *LastModifiedTimeStampField* que nous avons ajoutée dans notre modèle. Elle va nous permettre d'indiquer à l'indexeur de recherche quel est le nouveau contenu, ou le contenu fraîchement modifié, à indexer.

Que se passe-t-il si vous voulez appliquer la sécurité sur les résultats de la requête. En effet, le comportement par défaut de *SharePoint* lors d'une recherche est de ne montrer, à l'utilisateur émettant la requête, que les résultats auxquels il a accès. Pour réaliser cette tâche, *SharePoint* effectue une action qui s'appelle le « *Trimming* » ou hachage. Cette étape consiste à parcourir chacun des éléments remontés par la recherche et à vérifier si l'utilisateur a la permission suffisante pour le voir.

Lorsque l'on modélise notre propre *Trimmer*, la question à se poser est : « *Est-ce que je peux retrouver la liste des permissions au moment de l'indexation, ou est-ce que je dois attendre la requête de l'utilisateur pour hacher les résultats un à un ?* ». Dans un souci d'optimisation, s'il est possible, lors de l'indexation, de stocker la liste des permissions, la requête émise par l'utilisateur n'en sera que plus rapide. Si effectivement les permissions ne sont connues qu'au moment de la requête, il faudra

hacher chacun des résultats avant de les afficher à l'écran, alourdissant de surcroît l'action de l'utilisateur.

Pour mettre un mécanisme de hachage de sécurité au moment de la requête, il va falloir créer une nouvelle classe implémentant l'interface *ISecurityTrimmer2*, se trouvant dans l'assembly *Microsoft.Office.Server.Search.dll*. Cette interface propose notamment d'implémenter la méthode *CheckAccess*. Cette méthode prend en paramètres l'identité de l'utilisateur et un ensemble d'URL, qui sont les éléments externes hachés (le hachage se passe par lots, et dès qu'un lot est prêt, il est envoyé à la méthode). Il ne nous reste plus qu'à implémenter la logique permettant de vérifier auprès du système distant si l'utilisateur peut lire les éléments. Cette méthode doit renvoyer un tableau de « bits » indiquant pour chacun des éléments métiers si l'utilisateur a accès ou pas (0 ou 1). Dans l'exemple suivant, le trimmer indique que l'utilisateur peut tout lire : [Fig.15].

Une fois l'assembly déployée sur les serveurs *SharePoint*, il faut enregistrer ce « *trimmer* » au niveau de la ferme en utilisant la commande *PowerShell new-spenterprisearchsecuritytrimmer*.

Conclusion

A travers cet article nous avons vu qu'il était possible de se connecter à une source de données externe depuis *SharePoint* et cela de façon très aisée via les connecteurs fournis. Si toutefois le système externe pose plus de contraintes, ou si le scénario de connexion est plus complexe, il est possible de développer son propre connecteur. L'avantage d'utiliser les *Business Connectivity Services* est que toute la plateforme *SharePoint* prend connaissance de ces données, que cela soit la recherche, les web parts, les profils utilisateurs etc. L'interopérabilité n'a donc plus de limite...

Julien Chomarat – Consultant / Formateur *SharePoint*

Access It IDF / jchomarat@access-it.fr

blog : <http://www.sharepointofview.fr/Julien>

Fig.11

```
<Method Name="ReadList">
  <Parameters>...</Parameters>
  <MethodInstances>
    <MethodInstance Type="Finder" ReturnParameterName="returnParameter">
    </MethodInstance>
  </MethodInstances>
</Method>
<!-- end finder method -->
<!-- start specific finder method -->
<Method Name="ReadItem">
  <Parameters>...</Parameters>
  <MethodInstances>
    <MethodInstance Type="SpecificFinder" ReturnParameterName="returnParameter">
    </MethodInstance>
  </MethodInstances>
</Method>
<Method Name="AddBook">
  <Parameters>...</Parameters>
  <MethodInstances>
    <MethodInstance Name="AddBookInstance" Type="Creator" />
  </MethodInstances>
</Method>
```

Fig.12

```
<LobSystemInstance Name="BdcModel1">
  <Properties>
    <Property Name="ShowInSearchUI" Type="System.String">x</Property>
  </Properties>
</LobSystemInstance>
```

Fig.13

```
<MethodInstance Type="Finder" ReturnParameterName="returnParameter" Default="true" Name="ReadList">
  <Properties>
    <Property Name="RootFinder" Type="System.String">x</Property>
    <Property Name="LastModifiedTimeStampField" Type="System.String">LastUpdated</Property>
  </Properties>
</MethodInstance>
```

Fig.14

Nom : Livres

Sélectionnez le type de contenu à analyser :

- ☐ Sites SharePoint
- ☐ Sites Web
- ☐ Partages de fichiers
- ☐ Dossiers publics Exchange
- ☒ Ligne de données métiers
- ☐ Référentiel personnalisé

Sélectionnez l'application de service Business Data Connectivity :

☐ Analyser toutes les sources de données externes de cette application de service Business Data Connectivity

☒ Analyser les sources de données externes sélectionnées

- ☐ AdventureWorksLT2008
- ☒ BdcModel1

Fig.15

```
12 public class CustomTrimmer : ISecurityTrimmer2
13 {
14     public void Initialize(System.Collections.Specialized.NameValueCollection staticProperties,
15                           SearchServiceApplication searchApplication)
16     {
17     }
18     public BitArray CheckAccess(IEnumerable documentCrawlUrls,
19                               IDictionary<string, object> sessionProperties, Identity userIdentity)
20     {
21         BitArray retArray = new BitArray(documentCrawlUrls.Count);
22         for (int x = 0; x < documentCrawlUrls.Count; x++)
23         {
24             retArray[x] = true;
25         }
26         return retArray;
27     }
28 }
29
30 }
```




Administration PowerShell pour SharePoint

PowerShell a été introduit en 2006 afin de proposer des fonctionnalités de scripting avancé pour l'environnement Microsoft et par extension aujourd'hui pour SharePoint 2010. Ce langage orienté-objet a été développé en s'appuyant sur .NET et peut être enrichi par du développement .NET. C'est ce qui a été réalisé par Microsoft pour SharePoint 2010, offrant ainsi des commandes spécifiques à la plate-forme.

SharePoint 2010 utilise PowerShell 2.0, inclus dans Windows 7. La fonctionnalité Windows PowerShell est à activer sur Windows Server avec une mise à jour vers PowerShell 2.0 à appliquer si vous n'êtes pas sur 2008 R2. Microsoft a développé ses propres commandes PowerShell (cmdlet) pour SharePoint que vous pouvez utiliser sur un serveur de la ferme en exécutant la commande :

```
Add-PSSnapin Microsoft.SharePoint.Powershell
```

Il est possible d'exécuter des cmdlets sur un serveur SharePoint distant en configurant le Remote PowerShell (utilise Windows Remote Management) avec notamment la cmdlet *Enable-PSRemoting* que nous ne détaillerons pas dans cet article.

SYNTAXE

Les cmdlets sont sous la forme *Action-Objet* même si cette syntaxe n'est pas obligatoire. Pour afficher les processus Windows, nous exécuterons *Get-Process*. Nous pouvons spécifier des paramètres en ajoutant *-NomDeParametre ValeurDuParametre* à notre cmdlet. Nous exécutons la « *Get-Process -Name w3wp* » pour récupérer les processus SharePoint.

Nous préfixons nos variables d'un signe \$. Bien que typées, elles peuvent être implicitement déclarées lors de leur première utilisation. Vous pouvez ainsi affecter le résultat de la commande *Get-Process* dans une variable *\$mesProc* afin de la parcourir par la suite. De plus, PowerShell est un langage orienté objet, les objets ayant alors des propriétés comme *Id* pour le PID d'un processus.

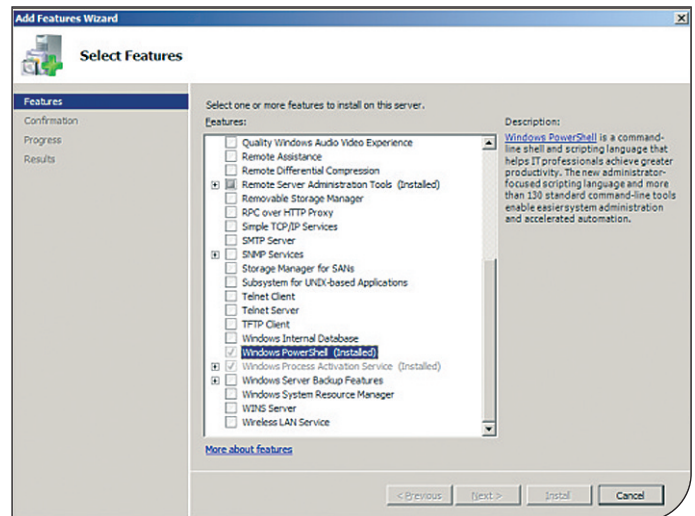
Nous pouvons mettre des scripts en place faisant appel à des tests et des boucles pour des traitements avancés. La syntaxe utilisée est assez simple et ressemble sensiblement à du C#.

```
if($maVar -eq $null){} //Test
do {} while($maVar -ne $null) //Do-while
for($i=0 ; $i -lt 5; $i++) {} //For
foreach($proc in $mesProc) { Write-Host $proc.Id ; } //For-each
affiche les PID des processus
```

Ainsi en ayant suivi les prérequis, nous pouvons utiliser les cmdlets spécifiques à SharePoint. La plupart d'entre elles ont la syntaxe *Action-SPObjet* comme *Add-SPSolution* qui permet d'ajouter une solution WSP à notre ferme SharePoint :

```
Add-SPSolution -LiteralPath "C:\PS\MaSolution.wsp"
```

Ainsi, le modèle objet, les tests et les boucles permettent de réaliser des scripts pour SharePoint qui remplaceront le duo batch-*stsadm* moins flexible.



CAS D'UTILISATION

Nous allons expliquer comment PowerShell peut être utilisé avec SharePoint 2010 dans vos projets. Après une partie configuration d'administrateur, nous déploierons nos solutions WSP et finirons par les possibilités de sauvegarde et de restauration.

Droits pour utiliser PowerShell

SharePoint nécessite des droits de Shell Admin pour exécuter certaines cmdlets. Si vous ne les avez pas, une personne qui a ces droits devra vous les accorder. Une manière de connaître les administrateurs Shell est de taper *Get-SPShellAdmin* dans une console PowerShell SharePoint exécutée en administrateur (clic droit, exécuter en tant qu'administrateur). Cette cmdlet vous retournera la liste de tous les administrateurs Shell SharePoint. L'un des administrateurs listés va pouvoir vous ajouter au groupe. Pour ce faire il lui faudra utiliser la commande *Add-SPShellAdmin* en spécifiant votre login.

```
Add-SPShellAdmin -UserName DOM\csu #domaine DOM, login csu
```

Cette cmdlet donne notamment des droits *db_owner* aux bases de données. Accordez donc ce droit avec précaution si vous êtes administrateur Shell.

Création d'une Web Application

Les fonctions *stsadm* pour la configuration SharePoint ont également été adaptées en PowerShell. Pour créer une application web, nous allons utiliser la cmdlet *New-SPWebApplication* en spécifiant les mêmes paramètres que ceux demandés sur l'interface web. Nous avons, entre autres, besoin du compte de service qui sera utilisé (*-ApplicationPoolAccount*), du nom de la base de contenus (*-Database*).

```
Windows PowerShell
PS C:\> Select-FeatureDefinition "00bfe71-4e45-48d4-b4d0-7e45c011bae5"

Name                : FeatureDefinition/00bfe71-4e45-48d4-b4d0-7e45c011bae5
Id                  : 00bfe71-4e45-48d4-b4d0-7e45c011bae5
DisplayName          : TeamCollab
SolutionId           : 00000000-0000-0000-0000-000000000000
ReceiverAssembly     : 
ReceiverClass        : 
Properties            : 
Version              : 0.0.0.0
Scope                : Web
AutoActivateInCentralAdmin : False
ActivateOnDefault    : True
RootDirectory        : C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\Template\Features\TeamCollab
Hidden               : False
ActivationDependencies : 00bfe71-d1ce-424e-9c62-a44894e0104, 00bfe71-7e5d-4186-9ba8-c847ac750105, 00bfe71-d22-43b2-a848-c957897908100, 00bfe71-f301-423d-b9d1-da7a54c50110...
GroupForInstall       : False
RequireResourceFile  : False
DefaultResourceFile  : core
Type                 : Microsoft.SharePoint.Administration.SPFeatureDefinition
Status                : Online
Parent               : SPFarm Name=SharePoint_Config
Farm                  : SPFarm Name=SharePoint_Config
UpgradedPersistedProperties : {}

PS C:\>
```

seName) et de l'instance SQL Server utilisée (-DatabaseServer). PowerShell attendra un objet de type utilisateur pour le compte de service plutôt qu'une chaîne de caractères spécifiant le login. Nous allons alors utiliser la fonction *Get-SPManagedAccount*. Il faut bien faire attention à spécifier un compte déjà déclaré comme compte de service ou il vous faudra utiliser *Set-SPManagedAccount* avant.

DatabaseServer est un paramètre qui vous permet de définir l'instance de la base SQL Server qui sera utilisée pour créer la base de contenu dont le nom est spécifié via le paramètre *DatabaseName*. Pour ce faire, il faut spécifier la valeur sous la forme « *serveur\instance* ». Nous obtenons alors la commande suivante :

```
New-SPWebApplication -Name MaWebapp -Path "C:\inetpub\wwwroot\VirtualDirectories\80" -Port 80 -URL "http://spvm" -ApplicationPool MaWebapp_Pool -ApplicationPoolAccount (Get-SPManagedAccount "DOM\WAadmin") -DatabaseName WSS_Content -DatabaseServer "sql2008\sp2010"
```

Déploiement de solutions développées

Imaginons que nous avons développé et packagé des modèles de sites dans un fichier WSP grâce à Visual Studio 2010. Nous allons ainsi voir comment déployer nos fichiers WSP de manière automatique via des scripts de déploiement de nos solutions.

Après avoir obtenu notre solution WSP, nous allons ajouter un script PowerShell (*.ps1) qui s'occupera d'installer notre solution. Ce duo de livrables facilitera ainsi le travail des administrateurs et évitera les erreurs humaines.

Nous commençons par l'ajout de notre fichier WSP sur SharePoint en utilisant *Add-SPSolution* en spécifiant le chemin absolu de notre fichier WSP en paramètre.

```
$sol = Add-SPSolution -LiteralPath "C:\PS\MaSolution.wsp"
```

Votre solution est maintenant ajoutée sur SharePoint mais n'est pas déployée. Nous avons récupéré le résultat de notre commande (la solution ajoutée) dans la variable *\$sol*, ce qui nous permettra ensuite de déployer la solution avec *Install-SPSolution*. Les paramètres que nous allons fournir à cette cmdlet vont dépendre de la solution. Nous ajoutons le paramètre *-GacDeployment* avec la valeur *\$true* si la solution contient des DLL à déployer dans le GAC (*\$sol.ContainsGlobalAssembly* à *True*). De plus, si la solution contient des ressources d'étendue application web (propriété *ContainsWebApplicationResource* à *True*) alors nous ajoutons le paramètre *-AllWebApplications*. Autrement nous spécifions l'URL de l'application web concernée (paramètre *-WebApplication* avec URL). Pour ce dernier paramètre nous effectuons donc un test dans notre script afin de ne déployer la solution que sur certaines applications web s'il le faut.

```
if ($sol.ContainsWebApplicationResource)
{ Install-SPSolution -Identity $sol.Id -GACDeployment:$sol.ContainsGlobalAssembly -AllWebApplications }
else
{ Install-SPSolution -Identity $sol.Id -GACDeployment:$sol.ContainsGlobalAssembly -WebApplication "http://spvm:8888" }
```

Tout comme avec *stsadm*, un job est créé et pour le démarrer immédiatement. En revanche avec PowerShell il faut temporairement arrêter le service « SharePoint 2010 Administration » (SPAdminV4) pour démarrer le job.

```
Stop-Service -Name "SPAdminV4"
Start-SPAdminJob
Start-Service -Name "SPAdminV4"
```

Il nous reste alors à créer notre collection de sites avec les mêmes paramètres que sur l'interface web. Notons que la valeur du paramètre *Template* prend la forme *NomTemplate#NumeroConfiguration*.

```
New-SPSite -Url http://spvm -OwnerAlias DOM\Acsu -Name "Mon site" -Template TMP#0
```

Backup / Restore

SharePoint nous offre certaines fonctionnalités de sauvegarde et de restauration. Pour une collection de sites, nous utiliserons *Backup-SPSite* qui nous générera des fichiers BAK, utilisés par la cmdlet *Restore-SPSite* pour la restauration.

```
Backup-SPSite "http://spvm" -Path "C:\spvm.bak"
Restore-SPSite "http://spvm2" -Path "C:\spvm.bak"
```

Il est également possible d'effectuer une sauvegarde de la base de configuration grâce à la commande *Backup-SPConfigurationDatabase*.

```
Backup-SPConfigurationDatabase -DatabaseName "SPConfigDB" -DatabaseServer sql2008 -Directory \\share\Backup -ShowTree
```

Enfin, nous pouvons sauvegarder toute la ferme avec la commande *Backup-SPFarm*.

```
Backup-SPFarm -Directory \\share\Backup -BackupMethod full
```

Nous pouvons ainsi réaliser des scripts PowerShell qui seront exécutés en tâches planifiées afin d'effectuer des sauvegardes régulières de notre plate-forme SharePoint.

CONCLUSION

PowerShell pour SharePoint 2010 nous propose de nombreuses possibilités afin d'automatiser des tâches pour toutes les phases d'un projet. Nous pouvons ainsi créer des procédures assez robustes permettant d'éviter des erreurs, de gagner du temps et même de planifier des sauvegardes. Nous pouvons maintenant délaissier *stsadm* pour PowerShell enfin abouti pour SharePoint 2010.

■ Christian SU
Consultant SharePoint - MCNEXT



SharePoint : un marché qui explose

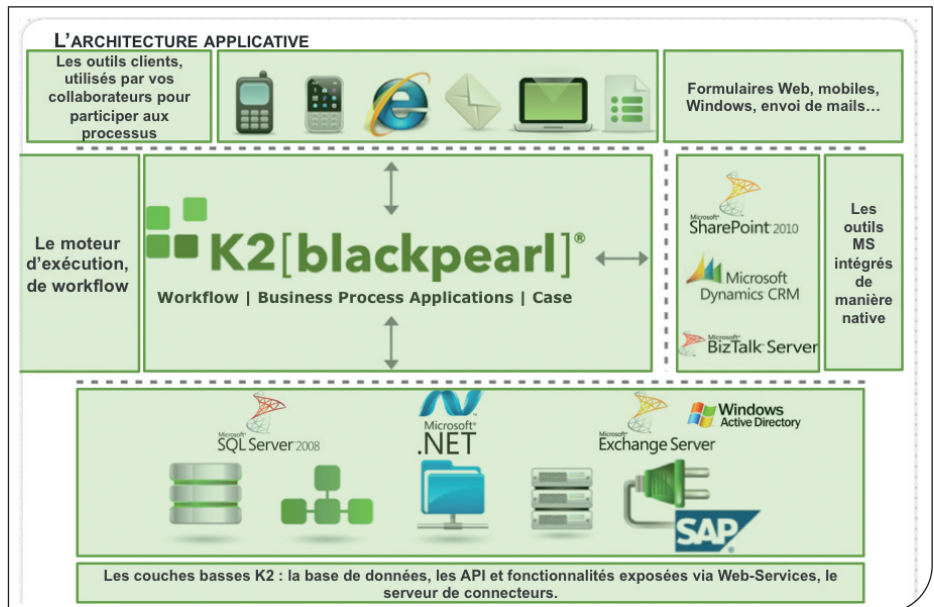
SharePoint resta longtemps dans l'ombre mais peu à peu, cette plateforme complète et intégrée, de partage et de collaboration a su s'imposer, version après version. Aujourd'hui, SharePoint est un des piliers incontournables de Microsoft en entreprise et pèse lourd dans la croissance de l'éditeur.

SharePoint, c'est jusqu'à 20 000 nouveaux utilisateurs par jour disait Jared Spataro (directeur de la branche SharePoint chez Microsoft) à nos confrères de ZDnet. En 2008, la solution dépassait pour la première fois le milliard de dollars de chiffre d'affaires.

En 2009, il était aux alentours de 1,3 milliard. Et ce n'est pas fini car avec la version 2010, SharePoint a monté encore d'un cran et a simplifié son installation et configuration, deux points noirs des anciennes versions. SharePoint 2010 est la continuité de la version 2007 qui fut une édition majeure avec de très nombreuses nouveautés. La prochaine version majeure est attendue en 2013 [?].

Un riche écosystème

SharePoint est une plateforme applicative ressemblant à un diamant brut qu'il faut tailler, sertir. Seul, SharePoint remplit de nombreuses fonctions mais souvent, il faut des applications / outils complémentaires, des développements. « SharePoint est une fondation sur laquelle on construit des étapes » précise Nicolas Martin (K2). Et si l'utilisateur veut aller plus loin, il lui faudra des développements spécifiques ou des



outils complémentaires. SharePoint est une plateforme très riche, en standard. « Il faut prendre les briques, réfléchir à ce que l'on veut faire. » confirme Gaetan Bouveret (Access-it). Il y a deux types d'utilisateurs SharePoint : ceux qui restent en standard et ceux qui migrent des applications, développent spécifiquement.

Si durant plusieurs années, SharePoint fut

regardé comme un simple intranet, outil de collaboration, ce n'est plus du tout le cas aujourd'hui.

SharePoint est une plateforme applicative et d'intégration très puissante. « Des entreprises migrent l'existant, développent de nouvelles applications et des DSI s'en servent pour réduire le nombre d'applications. » complète Gaetan. Le fait que la plateforme s'intègre avec .Net et Visual Studio facilite cette approche.

Trouver des compétences SharePoint : mission impossible ?

L'un des problèmes actuels du monde SharePoint concerne les compétences et les développeurs spécialisés. Un constat : la rareté des profils experts avec une solide expérience SharePoint, avec éventuellement la reconnaissance MVP.

Renaud Comte, de Wygwam, ne dit pas autre chose : « il y a peu de ressources vraiment qualifiées. Des projets peuvent être en échec du fait du manque de compétences ou de connaissances suffisantes de la plateforme ». Par exemple, à Wygwam, il y a trois experts MVP connus et reconnus. Quel est le profil du développeur SharePoint ?

Société	Type	Commentaires
K2	Editeur	Spécialisé dans les solutions et outils workflow et BPM intégrés à SharePoint. Plusieurs solutions disponibles : K2 blackpearl, Case Management Framework.
Access-IT	SSII	Très orienté conseil. Cible avant tout les éditeurs. Possède une forte offre de formations.
Wygwam	SSII	SSII spécialisée sur les technologies Microsoft. Possède une très forte compétence SharePoint. Wygwam forme les équipes, suit les projets, aide aux développements.
Winwise	SSII	SSII orientée technologies Microsoft. Propose du conseil, développement, aide aux projets, ainsi que des formations.
Calinda	Editeur	Editeur français proposant des solutions complémentaires pour aider les utilisateurs à déployer, compléter l'usage de SharePoint : MailSpace, SocialFactor, etc.
Microsoft Service	Consulting	Section consulting / développement de Microsoft. Aide les éditeurs, les clients et SSII sur les projets SharePoint.
MCNext	SSII	Possède 4 pôles de compétences Microsoft dont SharePoint. L'équipe regroupe 25 personnes certifiées SharePoint.
MEG-JIC	Indépendant	Pierre Erol Giraudy est un MVP SharePoint maîtrisant depuis plusieurs années la plateforme. Son offre : conseil, organisation, implantation SharePoint en entreprise.
Circum	Agence web	Agence web multi-technologie possède un pôle technique et projet SharePoint.
Eglia	Formation	Formateur SharePoint, avec certification mcts.
AvePoint	Editeur	Outils de migration, d'optimisation et d'administration autour de SharePoint.

Ce tableau présente une sélection non exhaustive de sociétés.

Ce sera plutôt un développeur .Net et/ou ASP.Net ayant une bonne connaissance de Visual Studio, de l'environnement .Net en général et des outils Office.

Wygwam embauche des profils juniors ayant un potentiel. Ils sont formés durant plusieurs semaines avant d'entamer leur premier projet.

Un projet, une mission SharePoint ne veut pas dire forcément du développement pur. Car l'expert SharePoint, surtout quand il travaille dans une SSII ou en indépendant, doit avoir un rôle de consulting, de formation et d'accompagnement des utilisateurs, des entreprises.

Il va aider aux développements, à configurer, à architecturer le projet et à bien identifier chaque rôle (intégrateur, développeur front side et back office, publishing). Le développeur SharePoint a un rôle important car il est à la fois administrateur, développeur et formateur. Et surtout les compétences varient beaucoup : design, développement .Net, workflow, etc.

Cette rareté tend le marché et chaque SSII cherche à trouver le bon profil. Ce qui peut hausser le niveau des salaires de départ.

Un projet type des projets SharePoint ?

Non, pas réellement. SharePoint étant extrêmement souple, vous pouvez en faire ce que vous voulez, ou presque. Cependant,



comme nous l'avons vu plus haut, un projet SharePoint est complexe et demande une grande rigueur, vu la richesse de la plateforme. Il faut pouvoir aligner SharePoint avec les objectifs, les projets. Un projet pourra ressembler à cela :

- compréhension des attentes, des besoins
- définir de l'architecture, du déploiement et

des services SharePoint à déployer

- installation et configuration
- développement du ou des projets et déploiement
- formation des utilisateurs

Mais aujourd'hui, l'entreprise comprend que SharePoint n'est plus uniquement de l'intranet, du portail, de la collaboration. Ces fonctions de base ne sont plus l'apanage de la plateforme et si c'est juste pour faire cela, l'outil n'est pas utile !

Quelques partenaires SharePoint en France

Il existe plusieurs catégories de partenaires autour de SharePoint : les éditeurs proposant des solutions complémentaires, des SSII / intégrateurs, des experts indépendants.

Vous pouvez contacter Microsoft pour connaître les principaux partenaires en France ou encore des clubs utilisateurs SharePoint.

Ex. : UGSF : <http://clubmoss2007.org/default.aspx>.

■ François Tonic

winwise

Société d'expertise spécialisée sur les technologies Microsoft



Conseil & audit

Mission d'expertise

Réalisation de projets à engagement de résultat

Centre de formation & de certification

www.winwise.com



Formation des développeurs : un pari sur l'avenir

Pour les SSII ou les DSI d'entreprise, former et certifier ses développeurs, c'est les motiver et les fidéliser tout en investissant dans l'avenir.

« **F**ormer ses développeurs c'est investir pour l'avenir, estime François Salaun, Pdg de Softeam. Au delà du développement, ils doivent rapidement apprendre les techniques de conception, les design patterns, les frameworks open source, les techniques de tests et d'intégration continue, car aujourd'hui, avec les méthodes agiles, conception, développement et tests sont étroitement liés. »

Michel Belli, directeur général d'Orsys,



ajoute : « Les développeurs dans une entreprise ont sans cesse besoin d'adapter les programmes aux besoins des utilisateurs. En les formant aux nouvelles techniques, ils apportent un gain certain de productivité à toute l'entreprise. De plus, cela contribue à l'amélioration de la performance informatique à la base de toute bonne organisation, que l'on se trouve dans un contexte d'entreprise privée ou publique. En cas de sous-traitance des nouvelles technologies, cas très répandu, il est important que les développeurs internes se forment pour comprendre et contrôler le travail des SSII. »

Les technologies et les outils évoluent tellement vite, tout comme les nouveaux marchés, que les développeurs doivent se maintenir à niveau pour garantir leur employabilité ou leur niveau de compétences. Philippe Ménard, directeur commercial et marketing du département

informatique de Demos, conseille les SSII : « Il est recommandé d'avoir une vision à trois ans, de faire le bilan des profils et des compétences-cibles, et d'évaluer et de positionner les salariés. Nous aidons également à réaliser des parcours de professionnalisation. »

En outre, sous la pression de la demande, le marché de l'emploi est tendu aujourd'hui, les développeurs se font plus rares. « Il existe à ce jour de nouveau un déficit de compétences en développement d'applications en général ; les SSII qui constituent 60% de nos clients récurrents recommencent en effet à nous « démarcher » et lancer des plans de recrutement importants », constate Corinne Gomes, direction marketing et communication, Global Knowledge. C'est un marché essentiel pour les organismes de formation. De plus, soumises à un turn-over important, les SSII sont d'autant plus promptes à former leurs développeurs, afin de les motiver et les fidéliser, un objectif souvent partagé par les DSI d'entreprise.

Quatre types d'organismes

Quatre types d'organisme sont en concurrence sur le marché de la formation des développeurs. Il existe des organismes de formation spécialisés en informatique comme Global Knowledge, ou généralistes, comme Demos ou Cegos, avec sa filiale dédiée à l'informatique, IB Cegos. Il y a des SSII, sociétés de conseil ou d'édition de logiciels, qui ont un pôle formation ouvert, telle la SSII spécialisée dans l'open source, Alter

Way, et son pôle Alter Way Formation (issu de l'acquisition d'Anaska en 2008).

En dernier lieu, il y a les SSII qui proposent des formations dans le cadre de projets clients. C'est le cas de Nelite, qui forme ses clients aux technologies Microsoft via le centre de formation Formaone.

Nombre d'organismes, outre les cours standard sur étagère en interentreprise et en intraentreprise, proposent des formations adaptées aux besoins spécifiques du client, facturant le temps de préparation. Ainsi, Alter Way Formation est en train de former 250 développeurs de La Poste section courrier, en fonction des projets des différentes entités, à HTML/Ajax, PHP, Linux, Apache, Zend Framework.

Certains organismes sont basés uniquement en région parisienne, d'autres ont plusieurs centres de formation en France, tel IB Cegos avec des centres dans six villes. Selon les organismes et les stages, les formateurs ne font que cela, ou parfois

Des formations en perpétuel renouvellement

Pour garantir la qualité de leurs stages et en proposer de nouveaux, les organismes réalisent une veille technologique et marché régulière. Les partenariats avec les éditeurs et les constructeurs placent certains organismes en amont des sorties de nouveaux produits. Les supports de cours sont remis à jour très régulièrement. Les formateurs remontent les suggestions des stagiaires, les commerciaux les demandes des entreprises.



DR Une formation chez Egilia Learning



DR Un cours à Alter Way Formation.

Panorama de formations (sélection et listes non exhaustives)

Organisme de formation	Type d'organisme	Site Internet	Formations proposées*	Formations les plus demandées	Exemples de formation		Quelques clients formés
Alter Way Formation (anciennement Anaska)	Pôle formation de la SSII	http://www.anaska.com	Développement en open source	PHP niveaux 1 et 2, Zend Framework Développeur, My SQL Développeur	Zend Framework Développeur, 5 jours, 1990 € ht : 23 mai, Paris, Lyon.	MySQL Développeur, 5 jours, 1900 € ht : 6 juin, Paris, Lyon.	La Poste, Orange, SNCF, Skyrock
Bull formation	Pôle formation de la SSII	www.bull-formation.com	Développement Internet, Unix, Business Intelligence et base de données	NC	Google App Engine, 3 jours, 1545€ht : 22 au 24 juin, région parisienne.	Programmation en Shell et outils Unix, 4 jours, 1380 € ht : 14 au 17 juin, région parisienne.	NC
Demos	Organisme de formation généraliste	www.demos.fr/fr/formations-informatique.htm	Langages et développement Java, Java EE, Microsoft, Web, mobile	NC	Android: développement d'applications pour mobile, 4 jours, 1940 € ht : 14 au 17 juin, Paris; du 26 au 29 septembre, Paris.	EJB 3X: mise à niveau pour développeurs EJB 2, 3 jours, 2211 € ht : 6 au 8 juin, Paris; 16 au 18 août, Paris.	NC
Edugroupe	Organisme de formation spécialisé en informatique	www.edugroupe.com	Développement C, C++, internet, mobile, IBM i, Java, Linux, Lotus Domino Notes, Sharepoint, Visual Studio, Websphere	NC	IBM i : les nouveaux outils de développement (RD Power / Rdi), 2 jours, 980€ht: 16 juin, Paris.	Linux: développement Python, 3 jours, 1500 € ht : 9 mai, Paris.	NC
Egilia	Organisme de formation spécialisé en informatique	www.egilia.com	Développement C++, Java, Java EE, Microsoft .net, Web	Java certifié, MCPD .Net	Développeur MCPD .NET, 20 jours, 3495 € ht : 23 mai au 17 juin, Paris.	Développeur Java certifié Sun, 10 jours, 3495 € ht : 30 mai au 10 juin, Lille, Paris.	Free, Philips
Global Knowledge	Organisme de formation spécialisé en informatique	www.globalknowledge.fr	Concepts de développement et modélisation (UML, XML), Java (Java, JEE, C++), les applications Microsoft, développement en environnement Open source.	SharePoint	Développer et personnaliser les Applications pour MS SharePoint 2010, 5 jours, 2490 € ht : 23 au 27 mai, Rueil ; 6 au 10 juin, Lyon.	Programmation C# avec Microsoft .NET Framework 4, 5 jours, 2290 € ht : 6 au 10 juin, Lille; 27 juin au 1er juillet, Rueil.	Cap Gemini, Expectra, Renault, St-Gobain
Hervé Schauer Consultants	Pôle formation du cabinet de conseil	www.hsc-formation.fr	Programmation sécurisée en PHP, sécurité informatique	Programmation sécurisée en PHP, tests d'intrusion, analyse forensique	Programmation sécurisée en PHP: e-learning 1 mois, 350 € ht	Programmation sécurisée en PHP: e-learning 3 mois, 500 € ht	Boursorama
IB Cegos	Organisme de formation spécialisé en informatique	www.ib-formation.fr	Technologies Java, Microsoft, Unix, IBM Lotus Domino, Internet et mobilité	Microsoft Visual Studio, Microsoft .Net, Silverlight, C#, Java	Développement d'applications avec Silverlight 3.0, 3 jours, 1590 € ht : 16 mai, Lyon; 10 octobre, Paris.	Visual Studio 2010 - Programmation C# avec Microsoft .Net Framework 4, 5 jours, 2475 € ht : 16 mai, Strasbourg; 23 mai, Rennes.	NC
Orsys	Organisme de formation spécialisé en informatique	www.orsys.fr	Objet, UML, Java, Java EE, .Net, C, C++, Visual Basic, Perl	C#, développer en .Net, PHP 5, développer un site Web dynamique, l'essentiel de Java et de l'objet, programmation objet en C++	L'essentiel de Java et de l'objet, 5 jours, 2270 € ht : 20 au 24 juin Strasbourg, Toulouse et Lille ; 25 au 29 juillet, Aix-en-Provence.	PHP 5, développer un site Web dynamique, 4 jours, 2110 € ht : 24 au 27 mai, 26 au 29 juillet Nantes et Rennes.	Alcatel Lucent France, Arcelor, Banque de France, BNP-Paribas
Softeam	Pôle formation de la SSII	www.softeam.fr	Méthodes et modèles, Développement JEE et Frameworks, .Net, Web 2.0, XML et autres langages	langages de développement (Java, JEE, et C#), frameworks JEE Struts, JSF, Spring,	Maîtrise du framework JSF - Java Server Faces, 3 jours, 1350 € ht : 9 au 11 mai, Paris; 30 mai au 1er juin, Aix en Provence.	Maîtrise du développement RIA avec GWT, 3 jours, 1350 € ht : 8 au 10 juin, Sophia Antipolis ; 29 juin au 1er juillet, Paris,	Amadeus, Orange, Pages Jaunes
Softfluent	Pôle formation de l'éditeur	www.softfluent.com	CodeFluent Entities, technologies de développement Microsoft		Développez avec Silverlight 4.0., 4 jours, 1960 € ht : 24 au 27 mai, Antony.	Développez avec ASP.NET 4.0, 5 jours, 2200 € ht : 20 au 24 juin, Antony.	pôle nouveau
Valtech Training	Pôle formation du cabinet de conseil	www.valtech-training.fr	Architecture logicielle, Java, Java EE, Microsoft .Net, développement Web, XML/Web services, et mobile, UML, C++, Oracle	Java et la conception objet, développement d'une application sur iOS	Java et la conception objet, 5 jours, 2038 € ht : 6 juin Paris la Défense et Toulouse.	Développement d'une application sur iPhone et iPad, 5 jours, 1941 € ht : 23 mai, Paris la Défense; 12 septembre, Toulouse.	Commission Européenne, France Telecom, Thales, CapGemini, Gemalto, Generali Vie
Winwise	Pôle formation de la SSII	www.winwise.com	Conception et développement avec les technologies Microsoft	Visual Studio 2010, Sharepoint 2010	Développer des solutions d'accès aux données avec Microsoft Visual Studio 2010, 5 jours, 2450 € ht : 2 au 6 mai, Boulogne-Billancourt.	Implémenter et gérer Microsoft SQL Server Integration Services, 3 jours, 1450 € ht : du 23 au 25 mai, Boulogne-Billancourt.	NC

NC: non communiqué

*liste non exhaustive

sont des consultants qui font aussi de la formation.

Les sessions de formation durent en général de trois à cinq jours sur un sujet précis. Des cursus plus longs peuvent aussi être proposés par certains organismes. Ainsi, IB Cegos propose « Développeur d'applications Windows sous Microsoft .Net », sur quinze jours qui comprend en fait trois formations. Des formations sont proposées durant l'été, ce qui permet, pendant le creux d'activité où les développeurs sont moins débordés, de les envoyer suivre une session.

Sur un marché concurrentiel, les organismes tentent de se différencier. Un certain nombre mettent en avant l'expérience



terrain de leurs formateurs. **Nicolas Barry**, directeur des opérations d'Alter Way Formation, souligne : « *Alors que l'open source devient une alternative crédible pour les grands comptes, nos*

formateurs sont des « geeks » très actifs dans la communauté open source, à la fois techniciens et pédagogues ». Egilia met en avant le suivi post formation. L'éditeur Softfluent a créé son pôle de formation Softfluent Training, en partenariat exclusif avec la société Pluralsight, acteur important des formations Microsoft aux Etats-Unis ; chaque participation à un cours physique donne accès à un an d'abonnement offert à l'offre en ligne Pluralsight On-Demand.

L'e-learning divise les organismes de formation. Pour certains, rien ne remplace le face-à-face. D'autres proposent des formations en ligne, plutôt les plus simples.

La certification, gage de compétences

Un développeur certifié, c'est un argument de vente pour les SSII, tout comme pour les développeurs freelance. « *D'après plusieurs*



études, note **Florence Sheng**, responsable du centre de formation de Winwise, la majorité des responsables embauchant des professionnels de l'informatique indi-

quent que les employés certifiés sont généralement plus productifs. La certification est un gage de qualité qui devient indispensable pour se distinguer de ses concurrents. » Elle est donc un point de repère pour les recruteurs, qui savent que le développeur peut être très rapidement opérationnel. La certification fait bien dans le CV pour trouver un nouvel emploi ; en termes d'évolution de carrière, elle permet de gravir les échelons.

Nombre d'organismes proposent des certifications. Orsys propose cinq cycles certifiants pour les développeurs. Certains font passer des Certificats de Qualification Professionnelle (CQP), certification officielle enregistrée au Répertoire National des Certifications Professionnelles (J.O. du 22/11/2010), délivrée par la Commission Paritaire Nationale de l'Emploi (CPNE). La branche Syntec CICF a créé le CQP « Développeur nouvelles technologies ». Il peut être obtenu en alternance dans le cadre du contrat de professionnalisation, de la période de professionnalisation ou de la validation des acquis de l'expérience (VAE). Demos propose par exemple deux CQP « Développeur nouvelles technologies », l'un orienté .Net, l'autre orienté Java, sur six mois, en alternance en entreprise.

Laurent Pénisson, chargé de la gestion des partenaires, Microsoft Learning : « *Parmi nos huit certifications, nous avons Microsoft Certified Professional Developer (MCPD), qui nécessite de passer plusieurs examens sur les technologies et le métier.* » Il valide la maîtrise de Visual Studio et du .Net Framework. L'autre certification pour

Essor du développement mobile parmi les formations les plus demandées

Avec l'explosion des smartphones et autres tablettes, la formation au développement mobile (iPhone, Android, Windows 7) a logiquement le vent en poupe. Mais les formations aux plateformes et langages Java, Java EE, C++, PHP, ou Microsoft .Net et autres technologies Microsoft, restent des valeurs sûres. Xavier Paradon, Directeur Technique, Valtech Training, observe « *un peu surpris, qu'il reste encore une grande population à migrer vers le développement orienté objet* », sa formation « Java et la conception objet » pour découvrir Java ayant été n°1 de ses ventes en 2010, en hausse de 35%. Parmi les autres formations porteuses, on trouve celles répondant aux besoins autour des outils de développement de Rich Internet Applications (RIA), comme Silverlight ; SharePoint (collaboratif) est aussi bien placé.

les développeurs est Microsoft Certified Solution Developer (MCSD) pour .Net.

Le cabinet Hervé Schauer Consultants, spécialiste de la sécurité, propose des certifications Global Information Assurance Certification (GIAC) de ses formations Sans, issues du partenariat avec Sans, institut américain de formation à la sécurité des

systèmes d'information.



« *Une certification sur une technologie donnée indique que la personne dispose de certaines connaissances techniques*, tempère **Omid Bayani**, directeur général

en charge de la formation de Softfluent, mais ne garantit pas qu'elle saura les mettre en application dans un réel projet client. »

■ Christine Calais



Salle de formation d'Orsys



Une formation Valtech Training.

Marketing digital CSS3 Ajax Sécurité Web Développement iPhone
GWT Web Services HTML5 Méthodes et pratiques agiles
SOA Silverlight Développement Android Hibernate
Java Scrum UML
XML Java EE
EJB 3
Développement Web
Gestion de projet
C++

Maître Chang, 4 jours de formation et déjà 30 000 disciples *

www.valtech-training.fr - info@valtech-training.fr
+33 (0)1 41 88 23 00



* Comme déjà 30 453 personnes
Téléchargez SMS Speaker, son application Android®



Open Source

Le monde s'ouvre à vous !



Des données aux licences ouvertes

L'open source fait partie du quotidien des entreprises, des utilisateurs même sans le savoir. Les box ADSL intègrent souvent des logiciels ouverts, tout comme les systèmes d'exploitation, les navigateurs. Cette présence se fait donc à différents niveaux. Dans la bureautique, les suites ouvertes comme Open Office, LibreOffice... sont régulièrement utilisées sur les postes de travail. N'oublions pas les succès des navigateurs Chrome et Firefox et même du moteur de rendu HTML, Webkit, intégré dans de nombreux navigateurs. La base de données est elle aussi très ouverte à l'open source : MySQL, PostgreSQL, Ingres. Mais il est clair que le monde informatique ne pourrait pas être 100 % open source. L'informatique sera un mélange d'éditeurs propriétaires et commerciaux et

de logiciels ouverts. C'est la réalité du terrain. Nous en faisons l'expérience à Programmez !. Pour certaines fonctions, nous pouvons utiliser du logiciel libre, pour d'autres, nous devons garder des outils propriétaires. Et nous ne sommes qu'un petit exemple. Il y a l'habitude mais aussi le pragmatisme de l'utilisateur et de l'IT à considérer.

Dans ce dossier, nous reviendrons sur le projet Mageia qui tente de faire oublier son origine et des polémiques nées lors de la création de ce fork de Mandriva. Nous verrons comment créer un package applicatif 100 % open source et nous nous poserons la question de l'intérêt des données ouvertes (open data), un marché en pleine explosion.

■ François Tonic

Défis et nouveaux marchés

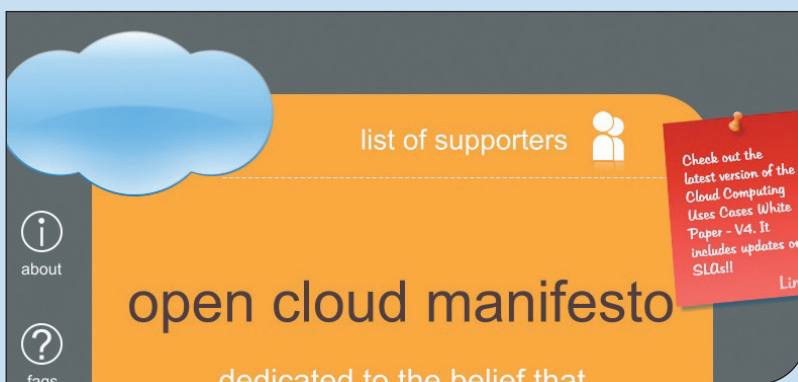
L'Open Source n'a pas une, mais une multitude de réalités dans les standards, les logiciels, en entreprise, à la maison, envers le développeur. Si Linux sur le poste de travail est un échec pour de nombreuses raisons, le logiciel ouvert est une réalité et le marché utilise indifféremment le propriétaire et l'ouvert, selon le contexte, l'usage.

En mars 2011, le Cigref (réseau de grandes entreprises) a publié un rapport sur l'open source dans les grandes structures : maturité et gouvernance. En 2011, l'intérêt de l'open source est bien compris et offre une autre vision, une autre approche de l'informatique, même si tous les problèmes ne sont pas résolus et que de nouvelles problématiques apparaissent, comme nous allons le voir dans ce dossier.

L'étude « maturité et gouvernance de l'open source » du Cigref

Les entreprises ayant choisi l'informatique ouverte se « justifient » par la pérennisation des composants, le levier budgétaire et technique avec les fournisseurs, le fait d'être plus standard avec des formats dits ouverts. Mais les contraintes ne sont pas à négliger. Ainsi, les auteurs, sur les licences, préconisent que : « *L'entreprise doit aussi analyser le type de licence des solutions étudiées, se poser la question de la compatibilité des licences avec les modules Open Source existants déjà au catalogue de l'entreprise et in fine de l'opportunité de mettre les solutions étudiées au catalogue des modules Open Source de l'entreprise.* ». Comprendre les enjeux, les obligations et les éventuels conflits entre les licences est impératif.

Et ce n'est pas aussi simple que cela. Il y a quelques semaines, OpenLogic, société spécialisée dans les applications mobiles et le support aux entreprises, a dévoilé une statistique étonnante : sur les 635 applications inspectées, 10 % contenaient du code open source. Ces dernières respectaient peu les clauses de licences (- 30 %). Et certaines clauses des contrats des boutiques applicatives mobiles et/ou des



licences ouvertes peuvent être incompatibles. Un casse-tête que l'on oublie bien trop souvent d'évoquer. Mais l'entreprise, les utilisateurs doivent aussi suivre l'actualité. Un projet open source, un logiciel libre est pérenne car la communauté assure le développement. Théoriquement. La force d'un logiciel ouvert se mesure au dynamisme de sa communauté, le support assuré par des sociétés, etc. Or de nombreux projets sont mort-nés ou s'arrêtent car limités à quelques personnes. Et la fragmentation d'une communauté, à cause de forks n'est pas à négliger : OpenOffice, MySQL, Mandriva, etc. Cela peut brouiller la perception du projet. L'étude prend un exemple très concret : OpenOffice (ou LibreOffice). « *Aujourd'hui, pour beaucoup d'entreprises, la principale alternative à la suite MS Office est Open Office de la communauté Open Source. Cet avis est renforcé par le nombre d'acteurs qui contribuent au développement d'Open Office (Oracle, Red Flag, IBM, Novell) et qui peut continuer à le maintenir si l'un d'eux fait défaut, ou développer de leur côté des éléments complémentaires ou alternatifs alors que Microsoft est seul.* », dit le rapport. Un des soucis de l'entreprise concerne la pérennité documentaire, comment assurer la bonne lecture d'un document vieux de 15 ans ou d'un document actuel dans 15 ans. Bref utiliser un format standard (voire normalisé) s'avère indispensable. On

a souvent opposé ODF, OpenXML, PDF. « *Les formats cibles d'échange et de stockage cités sont ODF (Open Document Format) et PDF (Portable Document Format). Le format ODF ne couvre néanmoins pas tout et pour certains le format*

OpenXML de Microsoft est plus puissant. Les entreprises savent donc qu'elles doivent, pour des usages spécifiques, conserver la suite historique MS Office. La cohabitation des deux solutions est facilitée par l'amélioration continue de la compatibilité entre les deux formats, l'ouvert et le propriétaire, et par l'ouverture native des formats de Microsoft (dans Office 2007). Les documents bâtis dans les règles de l'art se lisent assez bien dans MS Office 2003 comme dans Open Office mais la compatibilité n'est pas atteinte à 100%. ». Le développeur web aura le même souci de tests, de cohabitation. Il faudra bien à un moment donné tester sur l'ensemble des navigateurs du marché.

Un support de plus en plus banalisé

Aujourd'hui, les principales SSII, et la plupart des éditeurs open source / libre proposent des services, du support technique autour de l'open source. Le support fut parfois la faiblesse de certains projets ouverts mais désormais, l'offre couvre assez bien l'outillage ouvert. Dernière annonce importante : la collaboration HP – Linagora pour fournir un support open source couvrant jusqu'à une quarantaine d'outils. Un marché en pleine explosion même si les compétences sont encore difficiles à trouver. Nous verrons cela dans les prochains numéros. ■

MySQL : l'écosystème, le produit, la véritable histoire

Créé à l'origine pour répondre aux seuls besoins de ses créateurs, MySQL est progressivement devenu un produit autour duquel gravite désormais une véritable communauté et à partir duquel une société a été créée.

Devenue en 2001 « jeune entreprise innovante financée par capital risque », la société connaît plusieurs phases de développement avant d'être finalement rachetée par Sun Microsystems en 2008, laquelle passe à son tour sous le contrôle d'Oracle en 2010. Cette année et pour la première fois, MySQL sera ainsi présenté en profondeur à la plupart des participants de COLLABORATE 11, l'événement annuel de l'Independent Oracle User Group ou IOUG (Groupe indépendant d'utilisateurs Oracle). Cette présentation vise à décrire le contexte historique de MySQL, notamment pour les utilisateurs n'ayant pas suivi de près l'évolution de MySQL mais qui souhaitent bénéficier d'une technologie complémentaire à celle déjà utilisée.

Les origines de MySQL remontent au début des années 80, époque où un Finlandais, *Michael Widenius*, en écrit les premières lignes de code. Widenius (surnommé *Monty* par son entourage) décide alors de délaisser la technologie de base de données existante pour rédiger ses propres routines ISAM, destinées aux applications personnalisées pour ses clients finaux. Son entourage et moi-même mettons en doute l'utilité de ce qui s'apparente à une réinvention de la roue. Parallèlement à ses projets clients, il consacre un peu de temps au développement de ses routines ISAM jusqu'en 1995, lorsqu'un important client demande la migration d'une application vers SQL ainsi qu'une interface Web. Monty constate alors que sous Sybase, le délai d'exécution d'un lot quotidien passe de deux à 26

heures ! Monty informe le client qu'il est possible de créer une interface SQL au-dessus de ses routines ISAM et, par conséquent, lui demande de lui préciser un délai pour cette migration. Celle-ci devant être exécutée dans un délai d'un an, Monty développe, en neuf mois, un moteur SQL et un optimiseur axés sur ses routines ISAM.

Autour d'un produit, une base d'utilisateurs formant une communauté

Sous l'impulsion de *David Axmark*, cofondateur de MySQL, le code est rapidement diffusé auprès de la communauté sous une licence libre (non encore GPL). À l'aide d'une interface *Perl* similaire à *mSQL*, système très répandu à l'époque, MySQL conquiert rapidement de nombreux utilisateurs qui en apprécient la *stabilité*, les *performances* et la *facilité d'utilisation*. Sous la devise « zéro bogue », MySQL ne présente

qu'un très faible nombre d'anomalies, et ce, malgré l'absence d'un responsable d'assurance qualité chez MySQL AB jusqu'en 2005. L'excellente stabilité des serveurs MySQL se traduit par une longévité de fonctionnement exceptionnelle et un service ininterrompu de base de données destiné aux serveurs Web. Monty est obsédé par l'optimisation du code, ce qui entraîne irrémédiablement une amélioration des performances. La grande facilité d'utilisation est rendue possible par la « règle des quinze minutes » souhaitée par David, laquelle stipule qu'entre l'instant où un utilisateur final ayant simplement entendu parler de MySQL décide de tester cette technologie et le moment où le premier ensemble de résultats SELECT est créé, pas plus de 15 minutes doivent s'écouler. Ce quart d'heure doit inclure : la navigation sur le site Web et les pages de téléchargement, le téléchargement, l'installation et la configuration du progiciel, l'insertion d'échantillons de données et l'émission de la première requête SELECT.

Élevé au rang de référence de base de données au sein de la communauté *Perl*, MySQL forme par la suite un véritable couple avec PHP, notamment grâce à la mise en œuvre en moins de 24 heures, par Monty, de la clause LIMIT, une extension de syntaxe MySQL permettant aux boutiques Web de mettre en œuvre des recherches de produits tout en réduisant considérablement le temps système de la base de données. La pile Web est mise en place sous l'appellation LAMP (Linux - Apache - MySQL - Perl/PHP/Python) à la fin des années 1990.



CONSENSUS
Consensus may not always be possible. Or desirable.

“ Le consensus n'est pas toujours possible, ni souhaitable. ”

Marten Mickos, directeur général de MySQL AB et Michael « Monty » Widenius, cofondateur de MySQL

À partir d'un mouvement, une jeune entreprise innovante à croissance élevée, financée par CR

En 2000, les fondateurs de MySQL décident, dans un véritable « acte de foi », d'abandonner la licence libre au profit de la norme industrielle *General Public Licence* ou GPL (Licence publique générale). Les ventes de la société chutent de 75 % mais se redressent en six mois seulement, atteignant un taux de croissance deux fois supérieur. Début 2001, *Marten Mickos* est nommé directeur général et attire les premiers investisseurs en capital risque. Maintes fois racontée, la croissance de MySQL AB comme jeune entreprise innovante, à croissance élevée et financée par CR est un mélange de hautes technologies, d'un grand sens des affaires et d'un timing parfait. À l'époque, le serveur de bases de données MySQL a déjà conquis une communauté de plusieurs millions d'utilisateurs. Sous la direction de Marten, MySQL AB devient une société financièrement performante, et compte quelque 450 employés (contre une petite dizaine de personnes avant l'arrivée de Marten) lors du rachat de la société par *Sun Microsystems* pour un milliard de dollars en février 2008. Les innovations techniques et communautaires, associées à un grand potentiel d'inventivité, permettent d'identifier des catégories de clients, d'établir des partenariats ou encore de racheter à *Ericsson* la technologie *MySQL Cluster*; le tout dans un vrai mouvement d'entreprise. Innovation majeure, le matériel de génération de pistes permet d'analyser les visiteurs Web et les 60 000 téléchargements journaliers de MySQL afin de rechercher des clients éventuels et de les rediriger vers l'équipe de ventes intérieures ou, pour les marchés plus importants, vers l'équipe des ventes sur le terrain.

MySQL définit sa relation avec Oracle

La relation avec *Oracle* est dans le viseur de MySQL, et ce, depuis les prémises de la société. Au début de l'aventure MySQL, les domaines d'application des produits sont peu nombreux et MySQL est alors considéré comme un *gadget*. Sa réputation « *zéro transaction* » est toujours vivace, même longtemps après l'adoption de ces mêmes transactions. Créé par un autre Finlandais, *Heikki Tuuri*, le moteur

transactionnel primé *InnoDB* est le troisième moteur du genre proposé par MySQL. Appuyé par la plus petite équipe d'assistance, ce moteur bénéficie néanmoins du meilleur de la technologie et de l'expertise de la communauté. Au début des années 2000, deux ou trois cas d'usage et témoignages de clients sont publiés, parmi lesquels celui de la *NASA* qui déclare avoir réalisé d'importantes économies suite au remplacement d'*Oracle* par MySQL. Il semble évident que MySQL et *Oracle* répondent à des besoins distincts : un produit léger, quasiment libre et particulièrement stable, performant et facile d'utilisation - sans fonctionnalités de pointe - est clairement adapté à des utilisations pour lesquelles un produit complet lourd, exclusivement commercial et fermé s'avère moins agile en termes de technologies et de licence. MySQL devient alors la base de données du Web, desservant 17 des 20 plus grands sites Web du monde entier. La relation MySQL AB-*Oracle* prend un tournant lorsqu'en 2005, *Oracle* rachète *InnoDB* à *Heikki Tuuri*. Conséquence de ce rachat, le développement par MySQL AB de MySQL comme base de données pour *SAP J/3* est interrompu. Cependant, *Oracle* honore et renouvelle les accords existants entre *InnoDB* et MySQL, ce qui permet à MySQL de continuer à vendre des produits et des services *InnoDB* sur le marché MySQL.

L'acquisition du propriétaire de MySQL, *Sun Microsystems*, par *Oracle* suscite de nombreuses inquiétudes dans la mesure où MySQL et *Oracle* sont souvent considérées comme deux sociétés diamétralement opposées. Lors du rachat de *Sun* par *Oracle*, plusieurs acteurs clés ont déjà quitté *Sun*, notamment l'ancien directeur général de MySQL AB, *Marten Mickos* et les fondateurs de MySQL AB, *Monty* et *David*. À la tête d'une campagne anti-rachat, *Monty* déclare que cette acquisition menacerait l'avenir de MySQL. Les craintes de *Monty* sur une possible réduction de la concurrence sur le marché des bases de données trouvent un écho auprès de la *Commission Européenne* qui lance alors une enquête approfondie, repoussant par là même la clôture de l'acquisition de quatre à cinq mois. Suite à cette enquête, *Oracle* lance, par le biais d'un communiqué de presse (voir ci-dessous), dix promesses adressées à la base d'utilisateurs MySQL.

Ce qui n'a pas changé chez MySQL lors du rachat de Sun par Oracle

Le 21 avril 2009, à l'annonce par *Oracle* d'un « accord définitif portant sur l'acquisition prochaine de *Sun* par *Oracle* », j'étais le vice-président de *Sun* chargé de la Communauté MySQL. Quelques heures à peine après avoir entendu parler de cette acquisition, j'ai écrit une note de blog intitulée : « Ce qui n'a pas changé chez MySQL » afin de rassurer la communauté des utilisateurs et de relativiser les changements à venir. Cet événement ayant eu lieu lors de la conférence annuelle des utilisateurs MySQL, plusieurs déclarations immédiates ont été faites, les quatre plus importantes étant les suivantes :

- 1. La large base d'utilisateurs MySQL est préservée.** Les intérêts économiques de ces utilisateurs ne sont pas fonction du propriétaire de MySQL. Hétéroclite, la communauté d'utilisateurs MySQL compte aussi bien des utilisateurs occasionnels que des spécialistes des travaux internes de MySQL ayant contribué à la base du code.
- 2. Un grand vivier d'experts MySQL est toujours disponible** au sein de *Sun Microsystems*, dans les départements d'assistance, de conseil, de formation, d'ingénierie, etc. de la société. Ces experts sont très fidèles aux utilisateurs MySQL qu'ils servent depuis de nombreuses années.
- 3. MySQL est toujours sous licence GPL.** La licence GPL constituait un filet de sécurité pour les utilisateurs craignant que l'esprit « open source » ne soit pas respecté par MySQL AB. Cet esprit a été conservé sous l'égide de *Sun Microsystems*. De plus, la licence « open source » continue à protéger sa base d'utilisateurs, quel que soit le propriétaire de MySQL.
- 4. Des fondateurs de MySQL, surtout un en particulier, n'ont pas donné signe de vie depuis longtemps.** Toutefois, leur passion pour MySQL et ses utilisateurs reste intacte.

Deux ans plus tard, ces quatre promesses ont été globalement tenues.

- 1. Promesse entièrement tenue :** la large base d'utilisateurs MySQL existe toujours et ses intérêts sont indépendants du gérant du produit.
- 2. Promesse partiellement tenue :** le vivier

de talents existe toujours, fidèle à la base d'utilisateurs MySQL. Notons toutefois que certains talents ne travaillent plus pour Sun Microsystems (aujourd'hui Oracle). Un déploiement de l'écosystème a été observé : si, dans le passé, MySQL AB et Sun se trouvaient au centre de toutes les décisions concernant MySQL, un nombre croissant de sociétés participe, de nos jours, à cet effort. Ces sociétés ont été soit fondées par d'anciens employés MySQL (Petrona étant la première du genre), soit créées indépendamment de MySQL AB et de Sun, soit créées par d'anciens employés ayant récemment quitté MySQL (ex : Monty Program Ab et SkySQL Ab).

3. **Promesse tenue :** MySQL est toujours sous licence GPL. Certains produits liés à MySQL incluent des composants propriétaires, comme MySQL Enterprise Manager, mais cela était déjà le cas à l'époque de MySQL AB. Des extensions propriétaires sont certainement envisageables. Les bogues signalés par les clients prévalent sur ceux signalés par la communauté. De même, les contributions de code se retrouvent rarement dans la base de code centrale MySQL. Lancés à l'époque de MySQL AB, ces développements ont progressé au fil du temps ainsi que sous le contrôle de Sun, et pourront accélérer selon les intérêts du gérant MySQL actuel.
4. **Promesse entièrement tenue :** les fondateurs de MySQL AB sont toujours existants et actifs. Si la passion de Monty pour MySQL est parfois discutée dans sa forme, sa fidélité vis-à-vis de la base d'utilisateurs est incontestable.

Les dix commandements d'Oracle vis-à-vis des clients, développeurs et utilisateurs MySQL

Bien avant l'approbation par la Commission Européenne du rachat de Sun par Oracle, la société a proposé une série d'engagements vis-à-vis des clients, des développeurs et des utilisateurs MySQL. De cette promesse sur cinq ans, quatre années restent à venir. C'est pourquoi je recommande à toutes les personnes intéressées par MySQL de lire le communiqué de presse (en anglais), disponible à l'adresse <http://www.marketwire.com/press-release/Oracle-Makes-Commitments-to-Customers-Deve->

lopers-and-Users-of-MySQL-NASDAQ-ORCL-1090000.htm (URL réduite : <http://tinyurl.com/yzdg9tb>). S'il porte sur des sujets peu évidents pour les néophytes [« disponibilité continue du *moteur de stockage* APT, déclaration de « non-assertion vis-à-vis d'une tierce partie ou d'un fournisseur tiers pour l'implémentation de moteurs de stockage sous licence GPL » et « *MySQL Storage Engine Advisory Board* » (Comité consultatif de moteurs de recherche MySQL)], ce communiqué inclut également des engagements plus accessibles portant sur des améliorations sous licence GPL, un effort accru en R&D ou la poursuite de la disponibilité en ligne du *MySQL Reference Manual* (Manuel de référence MySQL), d'une assistance non obligatoire et, à la discrétion annuelle du client, d'un engagement de licence similaire destiné aux clients commerciaux, lesquels continueront de bénéficier d'un *MySQL Customer Advisory Board* (Comité consultatif de clients MySQL). Bien qu'un communiqué de presse n'ait aucune valeur juridique, Oracle devrait tenir ses dix engagements afin de ne pas faire l'objet d'une surveillance gouvernementale accrue dans le cadre d'acquisitions futures.

Le résultat final : conséquences pour les membres actuels de l'IOUG

Le contexte décrit plus haut ne présenterait qu'un intérêt purement académique si l'histoire de MySQL n'avait aucune conséquence pour l'utilisateur actuel. Il est difficile d'appréhender l'utilisation actuelle de MySQL sans connaître la nature de cette technologie. Cet article m'a permis, je l'espère, de raconter l'histoire de MySQL de manière suffisamment pittoresque pour que les membres de l'IOUG aient toutes les cartes en main afin d'évaluer les différentes options disponibles.

Je suggère aux membres de l'IOUG d'étudier les six conséquences suivantes :

1. **Même Oracle l'affirme : MySQL est un complément d'Oracle.** Dans de nombreuses situations, il est conseillé aux utilisateurs Oracle d'utiliser MySQL comme base de données Web en raison de sa légèreté.
2. **La présentation de MySQL comme une alternative ou non à Oracle** dépendra de votre interlocuteur. Il revient à chaque utilisateur MySQL de définir les limites exactes et d'évaluer le champ

d'application dans lequel MySQL peut se substituer à Oracle afin de réaliser des économies de coût total de possession. Les sociétés se sentent alors désarmées, car les intérêts des différents fournisseurs altèrent l'objectivité de leurs recommandations.

3. **L'écosystème MySQL inclut un ensemble « open source » d'attentes.** Les utilisateurs, clients et partenaires MySQL savent aujourd'hui ce qu'ils peuvent attendre du produit mais également de la communauté et des fournisseurs de produits et de services. Cette différence culturelle entre MySQL et Oracle justifie l'existence d'autres fournisseurs de services, indépendants d'Oracle.
4. **MySQL est une marque commerciale d'Oracle Corporation et/ou de ses filiales.** Par conséquent, MariaDB et les autres variétés de MySQL, basées sur et tout à fait autorisées par le modèle de licence GPL, ne peuvent pas porter l'appellation « MySQL ». Des variantes sont cependant disponibles et le client final dispose d'un large choix.
5. **Outre la marque commerciale, le site www.mjsql.com appartient également à Oracle,** de même que les droits d'auteur sur le logiciel MySQL. Cependant, une majorité croissante de développeurs, d'ingénieurs de support, de conseils, de formateurs, d'ingénieurs et responsables commerciaux MySQL AB connus par les utilisateurs MySQL à l'époque de MySQL AB sont liés à des sociétés autres qu'Oracle.
6. **MySQL est toujours sous licence GPL,** ce qui signifie une grande liberté de choix pour l'utilisateur MySQL. En raison de l'attractivité de l'open source en général et de la licence GPL pour logiciels libres en particulier, les utilisateurs MySQL bénéficient d'une utilisation, d'une assistance et d'un déploiement quasi illimités de MySQL.

Remerciements :
images Consensus,
Succès et Jeune
entreprise innovante
créées par Zack
Urlocker, ex-VPE
Produits chez MySQL
AB.

■ Kaj Arno,
VPE Produits,
SkySQL Ab



Mageia : du fork à la réalité du projet

Mageia est le fork de Mandriva Linux, annoncé le 18 septembre 2010. Cette annonce a été très vite accueillie avec enthousiasme. Sept mois après, le projet se développe et s'apprête à livrer une première release stable le 1er Juin prochain.

1 S'assurer du bien-fondé du fork

Même si, comme on le verra plus tard, Mageia est né dans l'urgence, un certain nombre de discussions ont eu lieu au sein de la communauté, avec ses acteurs majeurs. L'objectif principal : sonder le bien-fondé de l'idée d'un tel fork.

Quel est le niveau d'enthousiasme quant à cette idée ? Pouvons-nous nous appuyer sur un noyau dur d'enthousiastes et de contributeurs potentiels ?

Car après tout, on ne forke pas pour forker mais pour faire vivre une idée, un projet, une communauté. Dans le cas de Mageia, la réponse fut très positive dans les quelques semaines qui ont précédé le lancement du projet.

2 Poursuivre des objectifs communs et réalistes

La création d'une identité est liée intimement à la définition d'un ou plusieurs objectifs communs. Dans les semaines qui ont suivi la création de Mageia, ces objectifs ont été annoncés et expliqués :

- rendre le contrôle de la distribution à la communauté (nouvelle gouvernance),
- développer et livrer la distribution de manière indépendante (mise en place de l'infrastructure hard et soft, solide, maintenable et auditable),
- constituer des équipes et les faire collaborer,
- fournir une première version de la distribution sans innovation technique particulière mais stable (livrée ce 1er juin), utilisable au quotidien,
- utiliser ce premier cycle de moins d'un an pour se projeter sur les années à venir, avec de nouveaux projets dans et autour de la distribution (à partir de cet été).

Ne pas annoncer, ne pas expliquer ces objectifs, c'est générer des frustrations parmi les contributeurs mais aussi les futurs utilisateurs. Les espérances sont toujours fortes et généralement tout le monde attend du fork qu'il soit complètement opérationnel... hier ! 1 mois après

l'annonce de Mageia, des utilisateurs enthousiastes réclamaient déjà une version stable. Avancer donc, mais sûrement. Ces objectifs doivent intégrer et assumer une grande part d'incertitude quant aux objectifs à plus long terme (même si des tendances étaient annoncées dès le premier jour). Mieux vaut en effet y réfléchir, les mûrir en accord avec une communauté en développement et éviter ainsi des revirements ultérieurs, coûteux en termes de motivation et de ressources.

3 Établir un plan et un calendrier clairs

Il faut également étaler ces objectifs dans le temps, avec des étapes précises et datées à l'avance pour à la fois susciter l'attente et la satisfaire - celle des uns qui suivent et celle des autres qui travaillent à ce projet.

La préparation et la mise en œuvre de ce calendrier, initialement par les fondateurs, puis par le bureau et le conseil, est le métronome du projet. C'est cela qui permet d'inscrire les souhaits, les tendances de la communauté et les contraintes techniques dans une perspective productive.

4 Créer une identité propre

Mageia construit progressivement sa propre identité, bien à elle. Pendant les premiers temps du fork, Mageia a été confondu largement avec son projet d'origine, Mandriva Linux, que ce soit en termes d'objectifs, de communauté, de raison d'être. Une des questions les plus fréquemment posée reste « Pourquoi ce fork ? Pourquoi ne pas mutualiser les efforts ? ». Pas de distribution concrète à tester, sans doute aussi par manque de communication vers l'extérieur.

Un des éléments constitutif de cette identité c'est son identité visuelle. Le logo de Mageia a été choisi par l'équipe des fondateurs à l'issue d'un concours lancé dans la communauté et après consultation de cette dernière. Ce choix est intervenu 3

mois après le lancement du projet et a sans doute contribué à la difficulté d'identification du projet.

Enfin pour que cette identité s'affirme, Mageia a dû également assumer clairement les raisons du fork, les expliquer à la fois aux enthousiastes, aux curieux mais également aux détracteurs. Ceci en restant dans un registre clair et objectif malgré un contexte passionné et tendu, et ce, sans entrer dans des polémiques. C'est ce qui a été fait dans un premier temps avec la déclaration de création de Mageia.

5 Des règles de fonctionnement

La confiance et le respect (des personnes, des engagements, des objectifs) sont deux points fondamentaux sans lesquels ce projet ne peut pas tenir, sans lesquels il ne sert à rien de s'y investir [Fig.1].

Dès le départ, les fondateurs ont proposé comme constituants du projet :

- une gouvernance simple, accueillante (des équipes, un conseil communautaire et un bureau) pour structurer le suivi et les décisions et permettre de gérer les conflits, inévitables ; un projet a beau être communautaire, la collaboration ne suppose pas l'anarchie ;
- repris et adapté le code de conduite formulé un an plus tôt au sein de l'Assemblée Mandriva ;
- proposé un ensemble de valeurs définissant le projet (reformulées et organisées en un seul document par l'équipe marketing du projet).

Malgré le temps passé à discuter, expliciter ces règles de fonctionnement, celles-ci ne semblent pas toujours comprises par certains jeunes contributeurs ; là encore, un effort dans la présentation de la constitution du projet est nécessaire.

6 Délégation et expérience

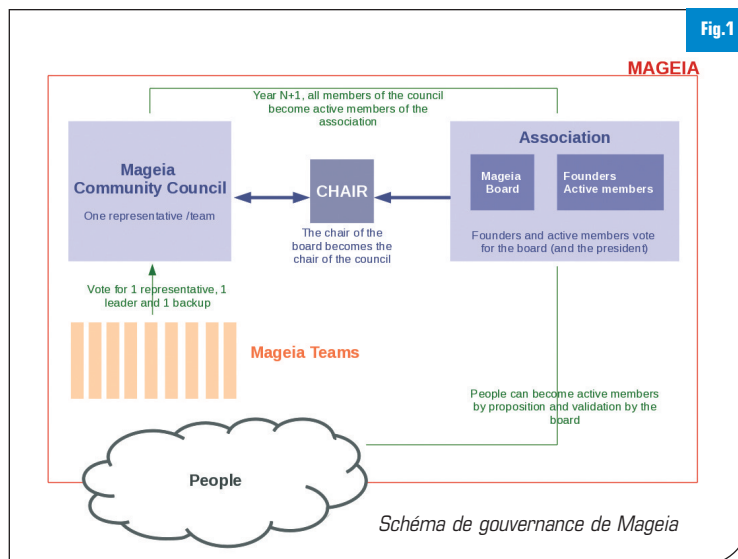
Un des points positifs de l'expérience Mageia a été la possibilité de démarrer l'aventure avec des gens expérimentés :

anciens packagers Mandriva, traducteurs, animateurs de forums, testeurs... C'est là un facteur essentiel de réussite. Cette équipe avait déjà connaissance des contraintes techniques, spécifications pour l'infrastructure et a pu rapidement à la fois fournir un système de base pour bâtir la suite de la distribution et encadrer des contributeurs moins expérimentés.

Cette équipe a pu se répartir des rôles d'animation et management des différents aspects du projet. Une équipe soudée et expérimentée a évité l'épuisement face à la somme des attentes et du travail des premiers mois.

7 Proposer des outils au service du projet et non l'inverse

Si la décision de forker pour Mageia a été prise en urgence, la mise en place de certains éléments d'infrastructure, même sous une forme temporaire, sont apparus rapidement essentiels pour maîtriser son démarrage et maintenir l'enthousiasme et la motivation du début. Il s'agit essentiellement des nécessités pour communiquer et coordonner les contributions : listes de diffusion, IRC, forum, wiki, blog. Pourquoi multiplier ces outils ? Parce que contributeurs, utilisateurs ou curieux ne privilégient pas les mêmes canaux de communication en ligne. Un développeur utilisera plus volontiers une liste de diffusion ou un canal IRC. Un utilisateur s'orientera plutôt vers un forum. Une fois ces outils de base mis en place, le parcours du combattant n'est pas terminé. Il a fallu mettre en place une infrastructure permettant de gérer les contributions : packages, traductions, tests, releases... Ce fut un des points les plus difficiles à résoudre. Devons-nous choisir un modèle existant quitte à perdre beaucoup de temps pour l'adapter à Mageia ou partons-nous de l'infrastructure existante de Mandriva Linux ? Un choix cornélien qui de toute façon devait prendre du temps pour la mise en place et potentiellement décourager les attentes des plus impatientes. Au final, décision fut prise de partir sur un existant avec de nombreuses améliorations.



8 Communication et pédagogie

Deux autres maîtres mots pour réussir le lancement d'un fork. La naissance de Mageia a déclenché une vague d'enthousiasme qui s'est traduite par une liste de contributeurs potentiels qui se sont inscrits spontanément pour participer. Il faut dans un premier temps distinguer les véritables contributeurs des supporters et moduler ce premier élan. Toutefois, supporter ou contributeur, l'objectif-clé est de maintenir le contact et la motivation ainsi exprimés. Il faut communiquer. Ça n'est jamais naturel. L'équipe s'astreint à proposer un post de blog hebdomadaire pour expliquer la progression de la mise en place du projet, ses besoins, même si certaines phases ont eu moins d'impact direct et de réalisations concrètes pour les contributeurs. Chaque réunion du conseil est publique et accessible à tout un chacun. La communication s'accompagne également d'une bonne dose de pédagogie pour faire comprendre les choix, expliquer les décisions, expliquer les lenteurs parfois... Un soin particulier par exemple a été apporté à l'explication du processus de bootstrap (régénération de la base de la distribution). Ce processus a nécessité 1 mois et demi de travail et impliqué de restreindre l'équipe aux seuls contributeurs avertis. Expliquer pourquoi cette restriction et pourquoi tout ce temps passé.

Si la communication est un maître mot et si Mageia s'est efforcé d'en faire un leitmotiv, il n'en reste pas moins qu'elle demeure une problématique à résoudre au

quotidien. Mageia est un projet international, l'amplitude des fuseaux horaires et les disponibilités de chacun rendent la tenue de réunions communes difficiles et faire un choix d'horaire génère toujours des frustrations. La barrière de la langue (même si l'anglais est la langue privilégiée) mène parfois à des incompréhensions voire des conflits.

9 Gérer la diversité des rôles

Une nouveauté pour les membres fondateurs de

Mageia a été de gérer des rôles qui ne sont pas cantonnés au seul domaine technique. Marketing, communication, graphisme, design, autant de domaines moins familiers mais indispensables à la vie du projet. Difficulté pour recruter des compétences en la matière et difficulté pour les intégrer aux côtés des autres équipes et favoriser une communication transversale. Mageia se retrouve à gérer ce contexte : les équipes techniques contre le reste du monde ! Si le démarrage du projet a mis l'accent sur le côté technique de la mise place des infrastructures, la suite devra relever le défi de voir le marketing et la communication monter en charge et éclairer la stratégie de plus long terme.

10 Il n'existe pas de petit conflit

Point sensible de toute société, à fortiori des communautés en ligne : les conflits. Qu'ils éclatent entre deux membres de la communauté ou entre un contributeur et un membre des organes décisionnels, ils génèrent des frustrations, des incompréhensions, et deviennent même parfois contagieux. Les responsables d'équipes sont donc particulièrement sensibilisés pour gérer ces conflits en amont et les désamorcer avant qu'ils ne prennent de l'importance. Les maîtres mots : transparence, respect du code de conduite et des valeurs de Mageia. En dernier recours, il est prévu que le Council tranche les désaccords.

■ Anne Nicolas et Romain D'Alverny
Membres de l'équipe des fondateurs de Mageia - <http://mageia.org>

Les licences logicielles

Avant utilisation, le logiciel vous invite à accepter une licence, c'est à dire une liste de règles et de restrictions. La plupart du temps, on se contente de cliquer sur le bouton Suivant pour échapper à ces étapes ennuyeuses et incompréhensibles, pour pouvoir enfin utiliser le logiciel.

Bien qu'en apparence anodine, cette étape conditionne en réalité l'usage que vous pouvez faire du logiciel. Il est donc intéressant d'avoir quelques connaissances sur le sujet. Nous vous proposons, dans cet article, de faire un rapide survol des licences les plus répandues.

Une licence logicielle est un texte établissant vos droits et devoirs lors de l'utilisation d'un produit logiciel, qu'il soit fourni sous forme binaire (comme Microsoft Office) ou sous forme de code source (comme le noyau GNU/Linux). Il convient de noter qu'une licence logicielle ne constitue en aucun cas un droit de propriété sur le logiciel, contrairement au contrat que vous acceptez implicitement si vous achetez un bien matériel (un grille-pain, ou une télévision 70 pouces à effet 3d et à écran Amoled)[1]. Si la licence ne vous donne pas de droits de propriété, à qui appartient alors le logiciel ? Il y a des variations subtiles selon les pays, en particulier entre l'Europe et les USA, mais la règle générale est qu'en l'absence de licence, c'est le droit d'auteur qui s'applique.

Le droit d'auteur est régi dans le détail par le code de la propriété intellectuelle (en particulier l'article L111-1), sur lequel nous n'allons pas nous appesantir. Notons simplement qu'il donne à l'auteur (qu'il s'agisse d'un individu ou d'une société) un droit de divulgation – c'est-à-dire, le droit de reproduire l'œuvre et d'en tirer profit. Cependant, le droit d'auteur ne permet pas à l'auteur de contrôler totalement son œuvre, qui peut être modifiée par des tiers, sauf si « cette modification

est préjudiciable à son honneur et à sa réputation ».

Pour renforcer les prérogatives de l'auteur, on ajoute donc une licence logicielle qui permet de combler les trous juridiques laissés ouverts par le droit d'auteur.

LES PRINCIPALES LICENCES

On peut classer les licences en deux grandes catégories : les licences dites « libres » et les licences dites « propriétaires ». Parmi les licences libres, nous allons étudier la GPL, la LGPL, et la BSD. Pour les licences propriétaires, nous nous contenterons de présenter un EULA générique, représentatif de la diversité existante.

Licences propriétaires

Les licences propriétaires se présentent en règle générale sous la forme d'un EULA – ou « End User Licence Agreement ». Cet EULA doit être accepté avant de pouvoir utiliser le logiciel.

Voici les principales caractéristiques d'un EULA propriétaire :



- Restriction de diffusion
- Restriction de modification et d'étude
- Restriction d'usage
- Limitations des responsabilités légales

La restriction de diffusion

Elle consiste à interdire à l'utilisateur de diffuser, par exemple, des copies du logiciel, ou encore à l'installer sur plus d'un ordinateur. Cette clause se retrouve par exemple dans les EULA de Microsoft.

Elle interdit également souvent de revendre le logiciel (ce qui est en contradiction, aux USA, avec la « first sale doctrine »...). Mais qui ne se transpose pas directement en droit français).

La restriction de modification et d'étude

Elle consiste à interdire à l'utilisateur d'étudier la manière dont le produit logiciel fonctionne, par exemple par décompilation. Il faut noter que cette mention, très souvent présente dans les EULA de sociétés américaines, peut se trouver en contradiction avec le droit français qui autorise ce genre d'opérations « à des fins d'interopérabilité ». [2]

La restriction d'usage

Elle peut prendre différentes formes, allant du nombre d'exemplaires installables du logiciel, jusqu'à des interdictions d'utilisation du logiciel dans un contexte non approuvé (par exemple, interdiction d'installer le logiciel dans N machines virtuelles en lui faisant croire qu'il est sur une unique machine simple cœur...)

(1) C'est d'ailleurs pour cette raison qu'il est illégal, au moins en France, de vendre conjointement et sans possibilité de vente séparée un lot constitué de matériel et de logiciels distincts, puisqu'ils sont de nature différentes, contrat de cession d'un côté et offre de service optionnelle soumise à conditions de l'autre. La vente de PC avec Windows pré-installé et non remboursable est un exemple type de violation de ce principe.

(2) C'est également pour cette raison que d'un côté, il devrait être légal en France de décrypter un DVD par des techniques de reverse-engineering. Mais de l'autre côté, la loi DAVSI va dans le sens contraire. Voir l'excellent blog de maître Eolas pour plus d'informations.

Limitation des responsabilités légales

Elle consiste à refuser d'endosser une quelconque responsabilité pour un usage particulier du logiciel. Par exemple, si le logiciel est utilisé dans un avion et si l'avion s'écrase, l'éditeur se dégage de toute responsabilité !

La licence BSD

La licence BSD (Berkeley Software Distribution license), issue de l'université de Berkeley aux États Unis, est une des plus permissives. En effet, elle ne comporte que deux restrictions :

- L'obligation de mentionner la licence BSD dans les sources du logiciel ou d'un logiciel dérivé
- La limitation des responsabilités légales

En conséquence, du code source sous licence BSD peut parfaitement être incorporé à du code propriétaire. Apple ne s'en est pas privé dans Mac OS X, fortement inspiré de BSD Unix, et Microsoft qui a incorporé du code dans la pile TCP/IP de Windows au milieu des années 90.

La licence BSD autorise donc la « privatisation » d'un logiciel libre, sans que les modifications soient reversées dans le produit original.

La licence GPL

La licence GPL (ou GNU Public Licence) est une des plus utilisées et emblématique dans le monde. Le noyau GNU/Linux en particulier est soumis à la version 2 de cette licence. La licence GPL a ceci de particulier qu'elle utilise astucieusement le droit de la propriété intellectuelle pour imposer des restrictions qui, au final, préservent la liberté de l'utilisateur !

La GPL garantit 4 libertés fondamentales :

- La liberté d'exécuter le programme — pour tous les usages ;
- La liberté d'étudier le fonctionnement du programme ;
- La liberté de redistribuer des copies — ce qui comprend la liberté de donner ou de vendre des copies ;
- La liberté d'améliorer le programme et de publier ses améliorations.

L'ensemble de ces libertés impliquent l'accès au code source du logiciel. La GPL est

ainsi conçue qu'un dérivé d'un logiciel GPL doit lui aussi être sous licence GPL, ce qui garantit récursivement la continuité des droits accordés à l'origine par l'auteur — ce dont la licence BSD, par exemple, ne se soucie pas. A ce titre, la GPL est souvent dite « virale » ou « copyleft » : du code source « mélangé » à un logiciel GPL doit être GPL ou devient GPL, et il est interdit de réutiliser du code GPL dans un logiciel soumis à une licence plus restrictive. (3)

Somme toute, il est interdit d'interdire !

Le choix de la licence GPL est sans doute un des facteurs qui a permis au noyau GNU/Linux d'être aussi répandu aujourd'hui. Autrement, il aurait été « contaminé » par du code propriétaire et probablement absorbé par ses concurrents.

En dehors des 4 droits fondamentaux, la GPL comporte également, comme toutes les licences logicielles, une clause de non responsabilité.

La licence LGPL

La licence GPL ne permet, en substance, d'utiliser du code GPL que dans des logiciels GPL. Si le logiciel est précisément une librairie, destinée à servir de brique de base dans la construction d'autres logiciels, la licence peut devenir un frein à la diffusion et l'utilisation. C'est pourquoi la LGPL (ou Lesser GNU Public Licence) a été conçue. Elle permet d'utiliser des bibliothèques LGPL avec du logiciel propriétaire, pourvu que l'édition de lien se fasse de manière dynamique. Ce concept de « liaison dynamique » a surtout du sens pour les langages compilés comme le C, le C++, ADA, etc. Avec des langages plus dynamiques comme Java, C#, Python, Ruby, ou PHP, la liaison statique au moment de la compilation n'existe pas.

En PHP par exemple, utiliser une librairie revient à copier des fichiers sources PHP au bon endroit.

Ils se retrouvent donc intimement mêlés avec le code propriétaire.

Quoi qu'il en soit, toute modification apportée à un logiciel sous LGPL doit aussi être soumise à la LGPL (s'il est diffusé). On retrouve donc bien le caractère « viral » de la GPL, destiné à maintenir les 4 libertés de base dans tous les dérivés du logiciel.

Compatibilité des licences

A ce stade, le lecteur aura deviné que ces différentes licences ne sont pas vraiment compatibles entre elles... S'il est possible d'incorporer du code BSD dans un logiciel propriétaire ou dans un logiciel GPL, il n'est pas possible d'inclure du code GPL dans du code BSD ou du code propriétaire, car on ne respecterait pas les 4 libertés de base. Par contre, on peut très bien construire un logiciel propriétaire en utilisant des bibliothèques LGPL, ou concevoir une solution logicielle comportant des briques GPL, du moment qu'elles ne sont pas « mélangées » avec les briques propriétaires. Il est ainsi très fréquent de rencontrer de l'électronique basée sur un noyau GNU/Linux sous GPL mais avec une interface graphique propriétaire, conçue avec des bibliothèques LGPL ! (4)

Et les brevets logiciels ?

Les licences logicielles n'abordent pas la problématique des brevets logiciels. Rappelons qu'aujourd'hui les brevets logiciels n'ont aucune valeur légale en Europe, même si l'OEB (Office Européen des Brevets) en accorde à la pelle !

Que se passe-t-il si du code source « libre » (GPL, BSD) implémente un algorithme protégé par un brevet logiciel ? La situation n'est pas tranchée, surtout aux États Unis, où certaines sociétés se sont réunies pour racheter les brevets afin de permettre à tous un usage sans restriction. Certaines sociétés, comme Novell ou Microsoft, passent entre elles des accords exclusifs sur des portefeuilles de brevets, excluant d'une éventuelle couverture juridique les autres utilisateurs du même logiciel Open Source...

CONCLUSION

Les licences logicielles, même si elles ne se présentent le plus souvent que sous la forme d'une inoffensive boîte de dialogue, ont des implications profondes sur les droits des utilisateurs et des concepteurs de logiciel. Nous espérons que cet article vous aura permis d'en maîtriser les rudiments.

■ Pascal Ognibene

Directeur technique Valtech

(3) A noter que les modifications d'un logiciel GPL, si elles ne sont pas diffusées, n'ont pas à être incorporées non plus dans le produit original. Si vous décidez d'utiliser en interne dans votre entreprise une version légèrement modifiée d'un logiciel GPL, personne ne vous ennuiera. Mais si vous diffusez une version binaire de ce produit modifié (en la vendant ou non), vous devrez diffuser les sources modifiées.

(4) Par exemple, presque tous les téléviseurs LCD récents de Sony, LG, Samsung, les GPS TomTom, ou encore l'intégralité des Box internet.

Un socle applicatif Java sur des briques open source



Le développement, c'est à la base une activité diversifiée, créative, voire ludique et le foisonnement du monde open source l'illustre fort bien. Mais travailler sur des applications de gestion d'entreprise, c'est être confronté à des contraintes qui obligent à faire passer ces attraits après des enjeux de productivité, de robustesse, de stabilité et de maintenabilité.

Faire cohabiter ces deux visions, celle de la créativité exploratoire et celle de la rationalité capitaliste n'a rien d'évident. Si l'open source apparaît gratuit au premier abord, son usage ne l'est pas et peut même se révéler particulièrement coûteux. L'utilisation d'un socle applicatif apporte le cadre qui répond aux enjeux d'entreprise. Que doit-on attendre d'un socle applicatif ? Comment le constituer ? Illustration avec l'exemple du socle applicatif open source Improve Foundations dans le monde Java. Improve Foundations est un socle technique destiné à cadrer le développement d'applications de gestion en Java. Édité par la société Open Wide Technologies, il est utilisé depuis plusieurs années par des grands acteurs de l'économie française. Il est constitué d'un cœur sous licence open source suffisant pour de petits projets. Le code source de ce socle applicatif est téléchargeable gratuitement sur le site Internet du projet : www.improve-foundations.org/openif.

Sélectionner les briques open source

Adopter une stratégie open source, c'est pour toute entreprise se confronter à l'embarras du choix : il y a tant de solutions concurrentes pour un problème donné et tant de problèmes spécifiques que la sélection ressemble soit à un casse-tête, soit à une loterie. D'un côté, on se lance dans des études déjà coûteuses, rien que pour évaluer ce qui existe. De l'autre, on choisit un peu au hasard, en fonction de la mode et des choix du voisin, et en croisant les doigts pour l'avenir... Un socle applicatif se définit avant tout comme une distribution de composants open source. Il vise à éviter que ce travail de sélection ne soit fait sur chaque projet, ce qui limite les coûts, les risques et favorise l'efficacité des développeurs qui peuvent passer d'une application à l'autre sans se former à différents contextes techniques.

Constituer un socle applicatif commence donc par ce travail de sélection qui repose essentiellement sur :

- une activité régulière de veille technologique qui assure une connaissance pointue de l'étendue des solutions et de leur adéquation aux problèmes posés par les applications d'entreprise ;
- un examen détaillé de la qualité des projets open source, que ce soit sur le code lui-même, la justesse de la documentation, la pertinence des acteurs mais aussi la cohérence de l'orientation des projets sur le long terme ;
- une évaluation de la popularité des projets, leur taux d'adoption ; même si popularité ne rime pas forcément avec qualité, choisir une solution méconnue, aussi innovante soit-elle, constitue inévi-

tablement un risque important ;

- une estimation de la pérennité de la solution enfin, toujours difficile à garantir dans l'open source, mais souvent liée à sa popularité d'une part et au niveau d'investissement des acteurs d'autre part. On n'insistera jamais assez sur l'attitude de recul nécessaire à cette activité de sélection. Elle consiste à se méfier des modes pour s'assurer qu'au delà du simple attrait inhérent à la nouveauté, les composants ont une réelle valeur ajoutée et qu'ils ont fait leurs preuves dans un contexte professionnel. Combien de projets ont-ils ainsi été lancés sur des solutions immatures, obligeant les développeurs à des acrobaties inattendues ou à des refontes techniques prématurées ? Si l'open source comporte des risques, c'est sur celui-ci qu'il faut avant tout se montrer vigilant.

Assembler les pièces du puzzle

Le travail ne se limite toutefois pas à la constitution d'une liste de composants basée sur les critères précédents : puisque les applications seront basées sur cette brochette d'éléments potentiellement disparates, il est indispensable de vérifier qu'ils peuvent cohabiter. La question est plus complexe qu'il n'y paraît : choisir quelques projets open source, c'est tirer à soi dix fois plus de bibliothèques puisque chaque projet dépend souvent lui-même de plusieurs autres projets. Par transitivité, cela augmente donc très vite la liste des éléments à assembler. Et comme si cela ne suffisait pas, il y a des risques de conflits. En effet, si vous avez choisi les composants A et B, il peut arriver que chacun d'eux ait besoin du composant C pour fonctionner mais dans des versions différentes ! Comme une webapp JEE classique ne peut a priori reposer que sur une seule de ces versions, on sent bien qu'un sérieux problème se profile...[Fig.1].

Par chance, si l'ensemble de la sélection a été faite à un instant t, on constate que les projets suffisamment actifs diffèrent relativement peu en termes de dépendances. A et B utilisent C dans des versions différentes mais l'écart est suffisamment faible pour que A accepte de fonctionner avec celle demandée par B ou inversement. Il convient donc de faire des choix pour résoudre ce genre de conflits. A ce sujet, l'usage de Maven s'avère bien utile. *Improve Foundations* par exemple est organisé en projet Maven multimodule avec une configuration parente (POM) qui centralise les dépendances et les versions utilisées. En cas de conflit, Maven réalise des choix par défaut, selon un algorithme souvent acceptable mais pas toujours. Pour maîtriser le résultat, il n'y a pas de solution miracle : on impose une version dans le POM parent et on



Vision des dépendances d'Improve Foundations Open Edition qui met en évidence la complexité de l'intégration

s'assure que tout va bien par une batterie de tests. Dans le cas d'Improve Foundations, cette stratégie s'est jusqu'à présent révélée suffisante. Reste qu'en théorie, rien ne garantit que ce genre de conflit puisse être résolu. Si A et B réclament effectivement des versions différentes de C, il faut élaborer des tactiques de contournement délicates ou envisager des architectures à la OSGi qui garantissent l'isolation des modules et de leurs dépendances respectives. Bref, on l'aura compris, au delà même de la problématique de gestion des dépendances, bâtir un socle applicatif peut difficilement se faire sans passer par la case intégration continue. *Improve Foundations* est ainsi construit quotidiennement de manière automatique à partir du code source extrait du référentiel. Le processus comprend une série de tests unitaires, des tests d'intégration automatisés sur une webapp utilisatrice et un ensemble de mesures de qualité (règles de codage et métriques). Ceci nécessite donc la mise en place d'une plateforme avec des outils tels que Maven, JUnit, HttpUnit, Hudson, Sonar, Nexus... sans oublier quelques serveurs d'applications et de bases de données pour vérifier que le socle est bien opérationnel sur des cas concrets.

Cadrer l'architecture et les usages

Packager une distribution de solutions open source, c'est déjà bien, mais ce n'est pas suffisant. Choisir Struts, Spring et Hibernate, ou GWT, Guice et EclipseLink (par exemple), cela donne une indication sur le contenu technique des applications mais ne cadre pas l'architecture. Beaucoup de solutions open source répondent à un besoin bien identifié mais vous laissent aller voir ailleurs dès que vous sortez de leur périmètre. Ainsi, vous avez peut-être de bonnes pièces détachées mais il vous manque le plan pour assembler la machine...

Cette remarque est sans doute moins pertinente pour des frameworks comme Spring qui bénéficient d'une situation plus centrale. Toutefois, il y a mille et une manières d'utiliser Spring. Ainsi, la phase d'architecture reste incontournable pour préciser l'assemblage des éléments et la manière dont ils seront utilisés. Penser socle applicatif, c'est dépasser cette frontière : n'y a-t-il pas moyen d'alléger le travail d'architecture récurrent en partant du constat que les applications de gestion ont des besoins similaires ? Ne peut-on pas proposer une architecture « clef en main », quitte à laisser une certaine ouverture pour traiter les cas particuliers ? C'est en effet le deuxième objectif d'un socle applicatif après l'assemblage des briques. Dans notre exemple, Improve Foundations y

répond d'abord en apportant une architecture en couches orientée services. Rien de nouveau sous le soleil ? Effectivement, quitte à faire un socle applicatif, autant qu'il repose sur des choix qui ont fait leur preuve... Ici, le socle ne se contente pas de proposer un conteneur de composants génériques mais de vraies notions de couches et de services, « codées en dur ». Cela signifie que ces concepts sont imposés et structureront l'application. Ainsi, Improve Foundations propose d'organiser les applications selon 3 couches :

- une couche « Entreprise » constituée de services d'accès aux données élémentaires (DAO) et partageables entre plusieurs applications (d'où son nom indiquant son côté transversal),
- couche « Application » constituée de services métiers élémentant le comportement fonctionnel (transactions, algorithmes fonctionnels, contrôles métier),
- couche « Client » généralement dédiée à l'IHM (présentation utilisateur).

Bien que cette architecture de référence puisse être adaptée à des contextes particuliers, celle-ci s'avère valable pour la grande majorité des situations [Fig.2].

La configuration est basée sur des fichiers XML mais reste limitée à l'essentiel : quelques paramètres techniques transverses et pour chaque couche, la déclaration des services identifiés par leur interface et l'implémentation Java à activer. Un mécanisme permet d'externaliser des paramètres, ce qui permet de changer d'environnement facilement, du développement à la recette, de la recette à la production, et à un service d'exploitation d'être autonome sur le paramétrage qui les concerne sans avoir connaissance des spécificités des frameworks choisis.

Autre exemple, l'accès aux bases de données. On a beau faire du Java, la manière de configurer et exploiter les accès à une base de données peut varier sensiblement selon que l'on utilise JDBC, JPA, Hibernate, ou MyBatis par exemple. Le socle applicatif, sans avoir la prétention de rajouter une surcouche complète à ces solutions bien connues, facilite leur intégration en proposant l'homogénéisation du paramétrage et de l'API d'accès aux connexions. Cette intégration va plus loin dans le cas d'*Improve Foundations* puisque son noyau apporte un moteur de gestion des transactions indépendant de l'implémentation sous-jacente. On peut ainsi mélanger différentes solutions techniques vers plusieurs bases de données au sein d'une transaction JTA sans que cela ait d'impact sur la complexité du code. Même esprit pour le mécanisme de



Une architecture qui a fait ses preuves

« remoting » qui permet d'indiquer par simple paramétrage que l'accès à un service se fait de manière distante et non plus locale. Le code Java reste identique et l'implémentation du protocole de communication, qu'il s'agisse de SOAP, de REST ou de RMI, est déporté dans les extensions techniques proposées par le socle.

Ce ne sont ici que quelques exemples. Dans le cas d'*Improve Foundations*, on pourrait encore présenter les mécanismes de valves, de contextes de thread, d'injection, de sécurité, évoquer aussi l'IHM, le reporting ou les fonctions de supervision. A chaque fois, l'objectif est d'exploiter les possibilités des solutions open source tout en proposant des services prêts à l'emploi et suffisamment cadrés pour qu'un développeur non expert puisse s'en servir en se concentrant sans risque sur les besoins de l'utilisateur final.

On le voit bien, le socle applicatif ne se contente pas de rassembler des briques open source, il apporte ici une surcouche qui vise à cadrer, homogénéiser et simplifier le développement.

Illustration par l'exemple

Pour vous donner une vision plus concrète, voici un exemple d'utilisation d'*Improve Foundations*. Commençons par un petit service d'accès aux données utilisant Hibernate. Voici l'interface du service décrivant le contrat :

```
public interface ItemService extends Service {
    Item findById(Integer id) throws ServiceException;
}
```

Son implémentation :

```
public class ItemServiceImpl extends HibernateConnectedService
implements ItemService {
    public Item findById(Integer id) throws ServiceException {
        return (Item) getDefaultConnection().load(Item.class, id);
    }
}
```

```
}
}
```

Sa déclaration en XML au niveau de la couche Entreprise :

```
<layer abstraction=»fr.improve.foundations.architecture.EnterpriseLayer»
    class=»fr.improve.foundations.architecture.EnterpriseLayer»>
    <services>
        <service abstraction=»fr.improve.demo.enterprise.services.ItemService»
            class=»fr.improve.demo.enterprise.services.impl.ItemServiceImpl» />
    </services>
</layer>
```

Ce premier service est utilisé par un autre service dans la couche supérieure dont voici l'interface :

```
public interface AppItemService extends Service {
    Item getItem(Integer code) throws ServiceException;
}
```

Et l'implémentation :

```
public class AppItemServiceImpl implements AppItemService {
    public Item getItem(Integer code) throws ServiceException {
        ItemService service = (ItemService)
            EnterpriseLayer.getInstance().getService(ItemService.class);
        return service.findById(code);
    }
}
```

Dans cette implémentation, on récupère un représentant du service de la couche Entreprise par une ligne de code, il serait également possible de procéder par injection de dépendance. Par



WHAT'S NEXT ?

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

Un événement exceptionnel à Paris !

Les meilleurs experts rassemblés pour vous

Découvrez les technologies & acteurs de demain...

www.whatsnextparis.com



ailleurs, un test sur le code reçu en paramètre a été volontairement omis ici pour alléger ce code. La déclaration en XML est similaire à la précédente. Veuillez noter la présence d'une TransactionValve dont l'utilité ne saute pas aux yeux sur cet exemple mais qui signifie que tout traitement sur les services de la couche Application seront implicitement transactionnels :

```
<layer abstraction=»fr.improve.foundations.architecture.Application
Layer» class=»fr.improve.foundations.architecture.Application
Layer»>
  <pipeline>
    <valve class=»fr.improve.foundations.core.transaction.Transaction
Valve»/>
  </pipeline>
  <services>
    <service abstraction=»fr.improve.demo.application.services.
AppItemService»
      class=»fr.improve.demo.application.services.impl.AppItem
ServiceImpl» />
    </services>
  </layer>
```

Pour terminer, jetons un coup d'œil à la configuration générale, celle de la couche technique transverse :

```
<foundation>
  <connectionManager>
    <connectionFactory name=»db1» class=»fr.improve.foundations.
extension.hibernate.HibernateConnectionFactory»>
      <dialect><xsl:value-of select=»connection/db1/dialect»
/></dialect>
      <dataSourceName><xsl:value-of select=»connection/db1/
datasource»>
      </dataSourceName>
      <mappingDirectoryLocation>/mappings</mappingDirectory
Location>
    </connectionFactory>
  </connectionManager>
  <services />
  <layers>
    <xsl:call-template name=»clientLayer» />
    <xsl:call-template name=»applicationLayer» />
    <xsl:call-template name=»enterpriseLayer» />
  </layers>
</foundation>
```

On y entrevoit le paramétrage d'une *ConfigurationFactory* pour l'accès aux données. Certaines valeurs sont exprimées à travers des variables XSL qui sont précisées dans un fichier d'environnement externalisé. Comme pour les autres couches, le tag <services> permet de déclarer des services mais qui auront ici un rôle technique et transversal.

Vous remarquerez également la déclaration des trois couches de l'architecture applicative. Il nous faudrait également regarder la couche Client mais cela nous ferait sortir du cadre de cet article. Le site internet du socle vous permettra d'aller plus loin dans la découverte. Bien sûr cet exemple est particulièrement simple et ne donne qu'une vision superficielle mais il illustre déjà que le socle Improve Foundations n'apporte pas de solutions extravagantes mais vise au contraire à cadrer des pratiques connues en favorisant la simplicité.

Industrialiser et accompagner

Travailler sur les étapes précédentes, aussi importantes soient-elles, a peu d'intérêt si on néglige celle qui consiste à rendre tout cela utilisable et utilisé. Et il y a encore du pain sur la planche...

La réalisation de projets Java nécessite souvent la mise en place de toute une chaîne d'industrialisation qui va de l'outil de modélisation à la plate-forme d'intégration continue en passant par l'environnement de développement et le référentiel de code. Cette plate-forme peut-être enrichie d'outils spécifiques, sous forme de plugins Eclipse et Maven, qui exploiteront le cadrage apporté par le socle pour apporter une meilleure productivité.

Evidemment, un bon socle applicatif se doit d'être accompagné d'une panoplie de documents et exemples pour faciliter la prise en main des développeurs.

Et en complément, des sessions de formation doivent pouvoir être données pour que les équipes, internes ou externes, soient opérationnelles rapidement [Fig.3].

Enfin, c'est le but d'un socle applicatif d'assurer des fondations solides pour des applications qui vont devoir tourner pendant une dizaine d'années si ce n'est plus alors que pendant ce temps le monde open source aura évolué quotidiennement. Travailler sur un socle applicatif cela signifie aussi accompagner les évolutions de l'open source dans la durée pour en limiter les impacts sur les applications et c'est sans doute là son plus grand défi.

Conclusion

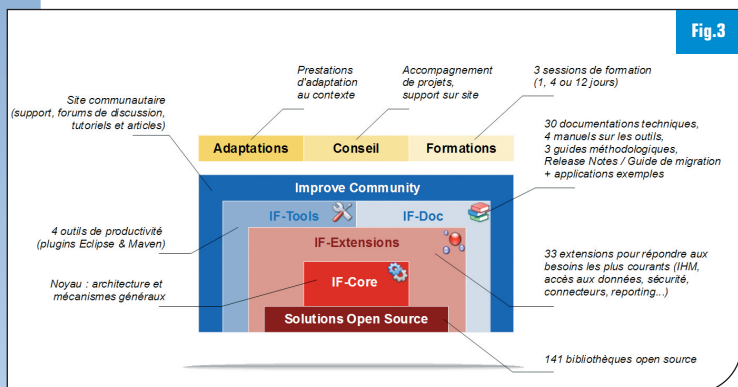
Cet article a rappelé l'intérêt de mettre en place un socle applicatif pour les développements d'entreprise au dessus des briques open source disponibles. Il a voulu montrer en quoi consistait la construction d'un tel socle et révéler, si vous n'en aviez pas déjà conscience, qu'il s'agit d'une activité à part entière s'inscrivant sur le long terme.

Sans être forcément innovant, le socle applicatif joue un rôle pragmatique de pivot pour faire cohabiter l'effervescence du monde open source et les enjeux de stabilité et d'efficacité de l'entreprise. Si en tant que développeur, le travail sur des applications de gestion ne vous amuse plus, vous pouvez toujours participer à la construction d'un socle applicatif !

■ Frédéric Esnault

Directeur Technique d'Open Wide Technologies

<http://technologies.openwide.fr>



Panorama du socle applicatif Improve Foundations

Open (Source|Data)

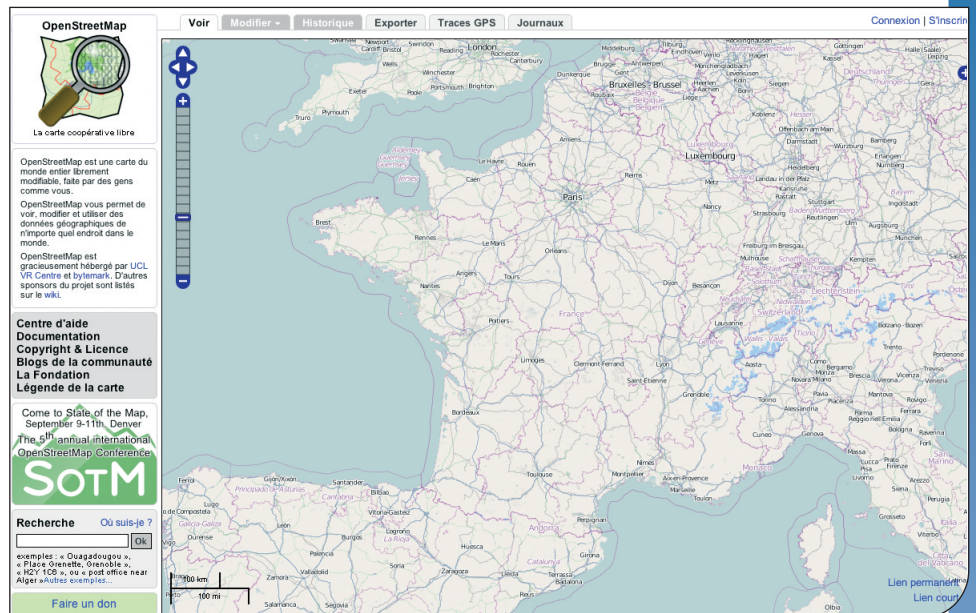
Lorsqu'on parle de logiciel libre, désormais il faut aussi penser à la donnée libre. Cette philosophie complémentaire est en vogue depuis 2010, au point de reléguer le Web 2.0 aux oubliettes. Nous verrons dans cet article comment aborder l'Open Data, l'utiliser et y participer.

Pour autant, ce n'est pas un enjeu nouveau, dans la mesure où un cadre juridique existe déjà depuis le 17 juillet ... 1978 (loi CADA, <http://www.legifrance.gouv.fr/affichTexte.do?cidTexte=LEGITEXT000006068643>) ! En 2005 est transposée en France une directive européenne datant de 2003 relative au droit de réutilisation des données publiques. En 2011, l'État Français active le projet Étalab, organisme chargé d'incuber la création d'un data.gouv.fr, la plateforme centralisée de distribution des données publiques. Un long chemin reste à parcourir, la vision politique et informatique de la distribution des données étant différente. Clairement, l'Open Data ne révolutionne pas nos vies, nous développons. Nous sommes habitués aux *mashups* , aux *apps* et à l'utilisation d'API pour servir ou être à la base de nos applications qu'ils soient clients, serveurs ou embarqués. En revanche si l'on regarde l'organigramme, cela change la vie des émetteurs des données, celle des services de la communication. Ils délèguent la technicité de la mise à disposition aux organes du SI mais gardent la main sur le choix et les modalités de diffusion de l'information ... dont la donnée n'est qu'une représentation informatique. Open Data reste et doit être un enjeu important, d'un point de vue philosophique, technologique et citoyen.

Les enjeux de la libération des données

Si en 1978 il manquait peut-être le vivier créatif et les moyens de mettre à disposition les données - les capacités de stockage, de calcul et de transport étaient loin d'égaliser la vitesse fibre - en 2011, Internet a eu le temps de s'ancrer dans les esprits depuis 15 ans.

L'enjeu fondamental de l'Open Data est la mise à disposition de données au grand public. Cela représente un enjeu dans la mesure où ces données n'étaient autrefois pas accessibles et qu'elles représentent une composante essentielle de la vie quoti-



dienne, car concernant tous les échelons de la société : finance, énergie, transports, services publics et territoriaux, économie, industrie, social, économie solidaire, justice, culture etc.

Ces données sont autant là pour offrir une transparence qu'un vivier d'innovations. Autrement dit, l'information n'est plus uniquement descendante (top-down et peu structurée) mais également remontante (bottom-up) voire même en réseau (crowd-sourcing).

Avant d'aller plus loin, accordons-nous sur la notion de donnée ouverte, c'est celle à conserver dans les esprits. Selon *Open Definition* (<http://www.opendefinition.org/>), une donnée est qualifiée comme ouverte si **n'importe qui est libre de l'utiliser, la réutiliser et la redistribuer** - et est soumis au moins de la citer et de la partager à conditions identiques.

Politique

L'Open Data est un moyen pour la sphère publique et politique de rendre des comptes à ses citoyens, et de devenir par la même occasion la politique au sens

large, à savoir fournir les moyens nécessaires à rendre une société organisée et structurée. D'un point de vue citoyen, c'est aussi un moyen de s'impliquer pour améliorer les usages, en créer de nouveaux mais aussi pour indiquer une direction à des dirigeants pas toujours en phase avec les besoins quotidiens.

La ville de Rennes fait partie des villes pionnières à mettre en œuvre la philosophie Open Data avec <http://www.data.rennes-metropole.fr/>. Y sont partagées des données liées aux transports, au budget, entre autres. L'adjoint au maire, Yves Préault, expliquait très simplement faire *"l'hypothèse que des applications seront créées pour analyser, par exemple, les consommations d'énergie ou d'eau dans les établissements scolaires, les ZAC, les équipements sportifs ou culturels"*.

Collaboration

Les données publiques peuvent aussi être issues d'un effort collectif pour influencer sur une situation considérée comme anormale, ou par désir de transparence. En ce sens, OpenStreetMap (<http://www.>

openstreetmap.org/) sert ce but, en permettant à quiconque d'enrichir les cartes, de les rectifier et de contribuer de nouvelles fonctionnalités... avec des données cyclables à vélo par exemple, via OpenCycleMap (<http://www.opencyclemap.org/>).

Dans un autre registre, celui du *crowd-sourcing*, l'initiative Prix de l'eau (<http://prix-deleau.org>) vise à mettre en exergue le prix théorique de l'eau au robinet, et celui réellement facturé. La démarche part de l'hypothèse que les habitants paient trop cher, et seront enclins à partager les différents montants, afin de faire pression sur les élus, eux-mêmes chargés de faire pression sur l'opérateur de distribution d'eau.

Compréhension

Qui dit données, dit consolidation, analyse et visualisation. Nous avons besoin d'aller au-delà des nombres, d'en lire une synthèse pour se forger une opinion et ensuite prendre une ou plusieurs décisions.

L'initiative NosDéputés.fr (<http://www.nosdeputés.fr/>) va dans ce sens : face à l'opacité du travail et de la présence parlementaire de nos élus, le collectif Regards Citoyens (<http://www.regardscitoyens.org/>) a décidé de monter ce site Web, et de mettre en forme de manière plus accessible des données publiques. Bilan : on peut prendre connaissance rapidement d'une synthèse de l'activité et de la présence de nos élus, de leurs dossiers et de leurs prises de parole. Une manière de visualiser si l'élu est en phase avec ses objectifs annoncés, et s'il travaille réellement pour ses citoyens.

Stimulation de l'innovation

Enfin, mettre à disposition des données publiques c'est fournir des données, exploitables, dans le but d'en trouver des usages nouveaux, des interprétations ou des analyses différentes.

Ces innovations viennent en soutien ou au contraire, deviennent un socle pour d'autres innovations, d'autres usages et de futurs investissements. Cette stimulation peut passer par des concours et dotations, afin d'amorcer ou d'accélérer la création d'applications et d'usages de ces données. Les retombées économiques et sociales, que ce soit par la création et la croissance d'entreprises, le maintien et la création d'emplois permettent de financer de nouvelles recherches ou la mise à disposition de nouvelles données, par le biais de système fiscal (taxes, imposition).

Ouvrir ses données

Maintenant que nous connaissons les enjeux de l'ouverture des données, et avant de s'attaquer au développement, sachons ce qu'est une donnée ouverte de qualité.

Cinq niveaux de partage

Dans un document datant de 2006 (<http://www.w3.org/DesignIssues/LinkedData.html>), Sir Tim Berners-Lee explique qu'Internet est particulièrement adapté à l'accès aux données, notamment via HTTP et REST, respectivement transport et représentation de données.

Il qualifie également 5 niveaux de partage des données, par ordre croissant d'efforts, de qualité, de réutilisabilité des données :

1. Disponible sur le Web (peu importe le format) mais avec une licence libre
2. Disponible dans un format structuré interprétable par une machine
3. Disponible dans un format structuré ouvert interprétable par une machine (CSV au lieu de Microsoft Excel par exemple)
4. Disponible dans un format structuré utilisant les standards du W3C (RDF et SPARQL) pour identifier les objets et pour faciliter l'accès aux données par des tiers
5. Disponible dans un format structuré utilisant les standards du W3C (RDF et SPARQL) pour identifier les objets et pour faciliter l'accès aux données par des tiers ; et les lier à des données tierces pour fournir un contexte supplémentaire

Les formats ouverts, et standardisés, sont bien évidemment à l'honneur et l'on peut dès lors distinguer 2 catégories de données : les données ouvertes et les données liées, par des relations, au travers du format RDF et de SPARQL, son langage de requête (accessible en tant que service Web).

Mettre à disposition des fichiers structurés réutilisables

L'objectif premier est la mise à disposition de données, dans un format lisible selon une norme ouverte et ne dépendant pas d'un format propriétaire.

Autrement dit, partager un fichier PDF est en soi une donnée libérée mais pas utilisable informatiquement parlant, sans pas-

ser par des processus complexes.

Des formats à privilégier sont CSV, XML, JSON, KML. Des bibliothèques en lecture/et ou écriture sont disponibles pour une très vaste majorité des langages existants, qu'ils soient client ou serveur. Vous souhaitez en plus les rendre lisibles par un humain ? Une feuille de style XSL fera l'affaire ou alors, utilisez REST pour proposer différents formats d'accès.

La mise à disposition de fichiers statiques est loin d'être chose nouvelle, ou particulièrement coûteuse. Leur génération s'effectue via des ETL (comme Talend) ou avec des scripts maison, déployés pourquoi pas avec un serveur d'intégration continue (comme Jenkins) afin de faciliter le contrôle de production et de mise en ligne des données.

Exposer des services Web

Les services Web permettent d'aller au-delà du caractère statique des fichiers en ajoutant une dimension de critère : l'information est récupérée à la demande, sur des identifiants bien précis, restreinte à une plage de données ... de son choix.

REST est l'architecture de mise à disposition privilégiée : une adresse par ressource, dont la représentation dépend ensuite des opérations et des paramètres renseignés. Elle reste simple, tant à implémenter qu'à consommer. Les documentations d'utilisation peuvent être générées automatiquement et ce, afin d'en faciliter l'usage par les consommateurs de services Web. Un bon moyen de contrôler l'accès et les consommations de données est de passer par l'identification du consommateur : OAuth ou une clé d'API, à défaut, plus simplement, sont deux mécanismes connus et là encore, dans la mise en œuvre et la consommation de la donnée.

YQL Open Data Tables

YQL (<http://developer.yahoo.com/yql/>) est un service Web de Yahoo! permettant de requêter d'autres services à la manière d'une requête SQL. Leur paradigme est "SELECT * FROM Internet", autrement dit un mécanisme d'accès unifié à des services Web hétérogènes, tout en donnant la possibilité technique de croiser les résultats ... avec des jointures.

Leur API est extensible via des tables communautaires, les *Community Open Data Tables* (<http://www.datatables.org/>). Ces



fichiers XML décrivent les paramètres entrants, sortants ainsi que l'opération à réaliser auprès d'un emplacement donné, que ce soit un autre service Web, un fichier statique ou même une page HTML. Autrement dit, constituer une Open Data Table et ne diffuser officiellement que via ce canal revient à utiliser YQL comme proxy. Résultat : vous offrez une visibilité sur les données publiées (via le catalogue communautaire), vous simplifiez l'accès aux données par le biais d'une API connue et surtout, YQL agit comme un proxy, réduisant la charge de vos propres serveurs. (Voir encadré bas de page).

Bâtir sur de la donnée libre

Il y a trois choses à considérer lorsque l'on souhaite développer une application basée sur des données libres :

1. le besoin à combler
2. les données existantes permettant d'y répondre
3. l'aspect financier, du coût estimé aux gains espérés

Les données de transport font partie des données favorites des développeurs, d'une part car elles sont relativement faciles à obtenir (même par des moyens détournés) et d'autre part car les fournisseurs de ces données n'ont et ne peuvent pas satisfaire tous les besoins de leurs usagers.

Dans mon cas, utilisateur de vélo en libre service de la Communauté urbaine de Bordeaux _ le VCub (<http://vcub.fr/>), je cherchais à pouvoir consulter facilement et rapidement la disponibilité des vélos.

Problème : la liste des stations (<http://www.vcub.fr/stations/liste>) ne me permettait pas de voir en un coup d'œil mes stations favorites (celles que j'utilise le plus fréquemment) et leur état. Deuxième problème : je ne pouvais pas le faire sans sortir de mon parcours, c'est-à-dire sans me rendre sur le site du fournisseur, via mon ordinateur ou mon téléphone mobile.

Mon objectif était clair : pouvoir accéder à mes stations favorites via un dashboard, à savoir Netvibes. Cela se traduit par la réalisation d'un widget UWA, embarquable sur un site Web ou un autre portail de widgets comme iGoogle ou encore MyYahoo!.

Identifier les données

Trois solutions s'offrent à nous pour obtenir l'information source :

- **Scraper** la source HTML de la liste des stations
Inconvénient majeur : on est fortement soumis à la mise en forme de la page, pouvant être altérée à tout instant.
- Récupérer la source de données de la cartographie des stations.
Inconvénient : là encore on dépend entièrement de la techno mise en œuvre, pouvant changer à tout instant, et ne contenant pas nécessairement toutes les informations nécessaires.
- Travailler à partir d'un flux GeoRSS.
Avantage indéniable : RSS est un format ultra connu et étendu de manière rétro-compatible avec une couche de données GPS.

Avant d'utiliser le flux (<http://www.vcub.fr/stations/feed/rss>), je m'interroge sur la compatibilité des navigateurs Web à pouvoir accéder aux *namespaces* XML. D'Internet Explorer à Firefox, les comportements ne sont pas les mêmes donc j'opte finalement pour une transposition du fichier GeoRSS en JSON (<http://www.vcub.fr/stations/feed/json>). Je gagne en poids de transfert mais aussi en facilité de lecture : je manipule des objets JavaScript et je n'ai pas besoin d'effectuer des requêtes complexes, juste une lecture séquentielle.

Le développement peut commencer.

Développer l'application

Le développement lui-même repose sur des standards :

- HTML
- CSS

- JavaScript
- Surcouche UWA documentée (<http://dev.netvibes.com/doc/uwa/documentation>)

Pour faciliter les livraisons (gestion de branches, développement et production), la collecte d'idées et d'anomalies mais également l'hébergement final du widget, je confie la gestion du code source à Git, via Github : <https://github.com/oncletom/uwa-vcub>. De la donnée libre naît une application libre, gratuite, ouverte aux contributions et en totale transparence avec les standards employés. Elle n'est plus maintenue ? Elle peut être dérivée. Une personne souhaite contribuer ? Elle peut être dérivée. Une personne souhaite l'utiliser comme base pour une autre ville ou comme socle d'un widget différent, utilisant les mêmes données ? Elle est dérivable. En d'autres termes, la mise en œuvre des données n'est pas et ne doit pas être liée à un langage, tant que celui-ci ne sert que d'interpréteur de données, interchangeable si besoin est. Le plus important réside dans l'utilisation des données, ce que le service apporte et de ce fait, de la création de richesse engendrée : qu'elle soit fonctionnelle ou en termes d'usage.

• Formater ses données :

RDF, RDFa, JSON-LD (<http://json-ld.org/>)

• Exposer des Web services sémantiques

Sparql
<http://www.w3.org/wiki/SparqlImplementations>

• Frameworks de développement :

Jena [Java], CubicWeb [Python], ARC2 [PHP]

■ Thomas Parisot

Consultant Web indépendant

Gérant de la société CyneticMonkey

<http://cyneticmonkey.com>

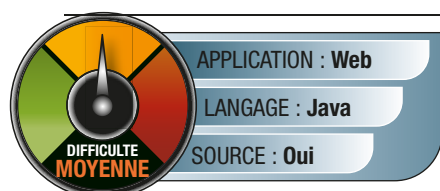
<http://case.oncle-tom.net>

```
USE 'http://www.datatables.org/amazon/amazon.ecs.xml' AS amazon.ecs;
USE 'http://www.datatables.org/bitly/bit.ly.shorten.xml' AS bit.ly.shorten;
USE 'http://www.datatables.org/delicious/delicious.feeds.popular.xml' AS delicious.feeds.popular;
USE 'http://www.datatables.org/delicious/delicious.feeds.xml' AS delicious.feeds;
USE 'http://www.datatables.org/dopplr/dopplr.auth.xml' AS dopplr.auth;
USE 'http://www.datatables.org/dopplr/dopplr.city.info.xml' AS dopplr.city.info;
USE 'http://www.datatables.org/dopplr/dopplr.futuretrips.info.xml' AS dopplr.futuretrips.info;
USE 'http://www.datatables.org/dopplr/dopplr.traveller.fellows.xml' AS dopplr.traveller.fellows;
```

Exemple d'invocation de tables multiples open data (yql) en "environnement". Un fichier environnement est un fichier texte contenant une liste de USE et de SET, avec extension .env. Exemple provenant de <http://developer.yahoo.com/yql/guide/yql-opentables-import.html#yql-opentables-import-environment>

Click : le Web facile et léger !

Avec l'essor de l'Open Source, l'univers Java voit fleurir de plus en plus de frameworks web. Si beaucoup ne décolleront jamais vraiment, il est bien souvent enrichissant de s'intéresser à certains d'entre eux qui peuvent permettre de nouvelles approches. Le framework Click fait assurément partie de cette catégorie de solutions. Tour d'horizon d'un outil qui a pour but de rendre le développement web en Java simple et rapide.



Né en 2003, Click ferait presque figure d'ancien dans le monde des frameworks web Java. Il faut dire que depuis l'apparition de Struts en 2000, le nombre de ceux-ci a connu une croissance exponentielle. Rentré dans le giron de la fondation Apache au cours de l'année 2009, puis promu au rang de top projet de la fondation au mois de Novembre de cette même année, Click présente de fait un certain nombre de garanties quant à sa pérennité. S'appuyant sur l'API Servlet de JEE, Click est un framework web moderne qui vise avant tout à simplifier le développement web en Java. La première release candidate de sa nouvelle version 2.3.0 est sortie à la fin du mois de Janvier de cette année.

ARCHITECTURE ORIENTÉE COMPOSANTS ET PAGES

Autre framework web de la fondation Apache, Tapestry avait été un des premiers à proposer une approche orientée composants. Suivie par les applications clients lourds, cette dernière avait déjà fait preuve de la productivité qu'elle apportait. Les pages et les composants fournissant une bonne encapsulation des concepts du Web, l'idée de base avec Click a donc été de prendre cette approche orientée composants et pages tout en la rendant plus accessible et plus simple à utiliser pour pouvoir développer rapidement des applications web se basant sur un modèle de programmation événementiel. Pour ce faire, les choix réalisés auront été radicaux. Tout d'abord, exit les pages JSP pour le rendu et exit le modèle MVC. En effet, ce dernier est un pattern plutôt orienté client lourd pour les créateurs de Click et il se révèle bien souvent inadapté voire trop lourd dans un contexte Web où le contrôleur et la vue sont généralement confondus. Afin de coller au plus près à l'esprit du Web et de rester le plus simple possible, Click est un framework stateless. Le côté serveur d'une application Click traite donc chaque requête provenant de la partie cliente de manière indépendante, ce qui rend les applications Click facilement scalables de fait. En outre, la configuration du framework est pensée pour être la plus transparente possible pour le développeur. Ceci étant rendu possible par l'utilisation du pattern Convention over Configuration. Ainsi, la configuration d'une application Click se résumera à la création d'un fichier click.xml placé au sein du dossier WEB-INF/ de votre application web et à la déclaration de la servlet ClickServlet au sein du fichier web.xml :

```
<web-app>
  <servlet>
```

```
<servlet-name>ClickServlet</servlet-name>
<servlet-class>org.apache.click.ClickServlet</servlet-class>
<load-on-startup>0</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>ClickServlet</servlet-name>
  <url-pattern>*.htm</url-pattern>
</servlet-mapping>
</web-app>
```

Le fichier click.xml est le cœur d'une application Click. Il permet notamment de spécifier les pages de l'application ainsi que le mode dans lequel celle-ci doit être lancée. Ce mode va définir le niveau de log et celui de cache de l'application. Le mode développement va permettre d'obtenir des rapports d'erreurs poussés à l'écran lorsque le chargement des pages ne se réalisera pas correctement. Enfin, la simplicité de Click passe le plus souvent par une absence d'abstractions complexes au sein de son API qui viendraient nuire à la courbe d'apprentissage et de maîtrise du framework. Le but de Click est donc avant tout de se concentrer sur la résolution la plus aisée possible des problèmes actuels se posant dans le développement d'applications web.

DES PAGES PRÉPONDÉRANTES

La notion de pages est au cœur des applications web. Au sein du framework Click, les pages vont permettre le traitement des requêtes HTML ainsi que le rendu des réponses HTML. Comme expliqué précédemment, Click est stateless ce qui implique qu'à chaque requête les pages et composants impactés sont recréés. Pour Click, une page logique regroupe une classe Java héritant de la classe Page du framework et un template Velocity qui va permettre de rendre la page au format HTML. Une page logique est ensuite définie au sein du fichier click.xml en précisant le chemin vers le template ainsi que la classe Java correspondante, de la manière suivante :

```
<page path="mapage.htm" classname="pages.MaPage" />
```

Cependant, en respectant le principe de Convention over Configuration appliqué dans Click, il n'est pas nécessaire de définir chaque page logique au sein du fichier de configuration click.xml. Il peut être suffisant de définir le package contenant les classes Java associées aux pages logiques Click comme suit :

```
<pages package="fr.myapp" />
```

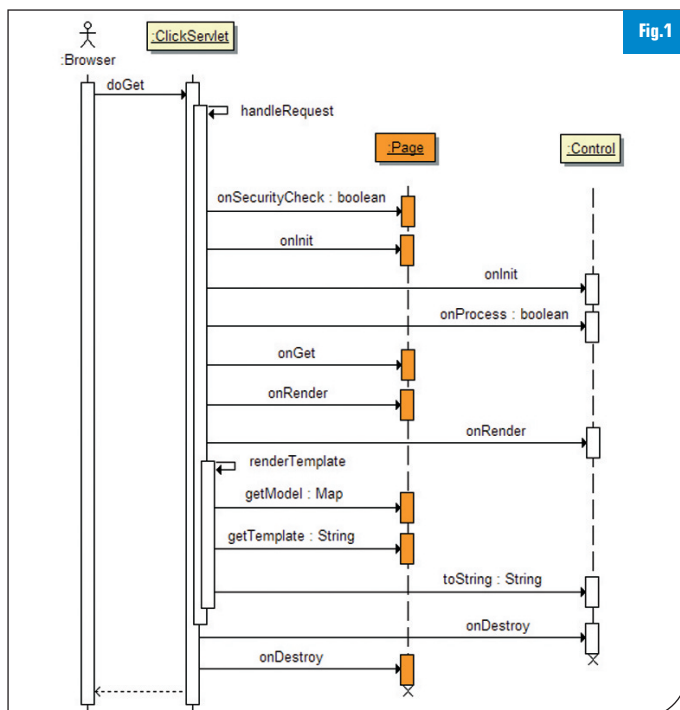
Avec cette définition, lorsqu'un utilisateur essaiera d'accéder à la page mapage.htm, Click va réaliser un mapping automatique en allant chercher une classe nommée MaPage dans l'arborescence des packages à partir de fr.myapp. S'il trouve une page MaPage au

sein du package `fr.myapp.pages`, il ira ensuite chercher `mapage.htm` au sein du dossier `pages` dans le `WEB-INF/` de votre application web. Ce mapping automatique s'avère très pratique afin de limiter les déclarations de page au sein du fichier `click.xml` et surtout afin de limiter la configuration de votre application. Il faut également noter qu'au cas où des pages logiques seraient définies entièrement et qu'en plus la définition d'un package de pages serait fait via la balise `pages`, Click utilisera en priorité la définition entière d'une page logique. Enfin, ce mapping automatique peut être désactivé le cas échéant.

La classe `Page` amenée par Click fournit un certain nombre de méthodes, vides par défaut, qu'il est possible d'overrider afin d'implémenter certaines fonctionnalités comme la gestion de la sécurité par exemple. Le cycle d'exécution d'une page au cours d'une requête HTTP de type GET est détaillé via le diagramme de séquence de la [Fig.1](#). Enfin, puisque les pages Click sont représentées en partie par des classes Java il est aisé de créer des pages réutilisables mais également des pages templates pour l'ensemble d'une application web. Cela se matérialise par la création d'une hiérarchie de pages. Chacune des pages de l'application héritant ensuite des pages templates de l'application.

DES COMPOSANTS STANDARD

Les composants constituent la seconde pièce maîtresse du framework Click. Ils ont à leur charge leur propre rendu et doivent également réagir aux interactions utilisateur. De fait leur rôle est plus étendu que celui dévolu à de simples composants et c'est pour cela que dans la terminologie Click, ils sont appelés contrôles. Le fait que les contrôles Click puissent gérer leur propre rendu permet de simplifier le travail du développeur puisqu'il n'est pas nécessaire d'avoir à définir deux fois le rendu d'une interface utilisateur. La définition d'un contrôle de l'interface utilisateur réalisée au sein de la classe Java associée à la page Click sera ensuite utilisable via une variable de type `$myForm` par exemple dans le template Velocity correspon-



Séquence d'exécution d'une page au cours d'une requête HTTP

dant. Le rendu d'un contrôle est réalisé via l'appel à sa méthode `toString()`. Néanmoins, il reste possible le cas échéant de contrôler directement le rendu d'un contrôle au sein du template de page associé.

Le framework fournit en standard plus d'une quarantaine de contrôles qui correspondent aux éléments HTML majeurs et qui permettent de construire rapidement des pages HTML sans avoir à étendre le framework. Pour des besoins plus poussés, les contrôles Click sont facilement extensibles puisque basés sur une approche orientée objet. Ainsi, il est aisé de créer une hiérarchie de contrôles réutilisables avec des relations d'héritage ou de composition. Un bon exemple pour ce type de création de contrôles est représenté par le contrôle `CreditCardField` du framework qui hérite du contrôle `TextField` tout en possédant des contrôles de type `Select` pour spécifier le type de carte souhaité.

Enfin si Click est stateless, il supporte néanmoins les contrôles de type stateful, ce qui peut être utile dans certains cas pour des applications web avec des pages complexes comprenant plusieurs formulaires et tableaux et qui nécessitent de maintenir leur état entre différentes interactions de l'utilisateur.

MOTEUR DE TEMPLATES

Pour le rendu des pages, Click utilise le moteur de templates Velocity. Son intégration avec ce dernier est transparente pour le développeur puisque toute la configuration Velocity est gérée par le framework. Le développeur a juste à se contenter de travailler sur ses templates et à manipuler les différents objets mis à disposition par Click en entrée du modèle du template Velocity concerné. Par défaut, le développeur peut ainsi interagir avec les objets classiques comme le contexte de l'application, les objets `request`, `response` ou `session` notamment. En outre, Click place dans le modèle du template l'ensemble des champs publics de la classe de la page logique associée ainsi que les variables annotées avec l'annotation `@Bindable` du framework. Ce comportement par défaut peut bien entendu être modifié si le besoin sur l'application s'en fait sentir.

Le framework offre la possibilité d'utiliser d'autres moteurs de templates que ce soit Freemarker ou bien JSP. La principale raison qui a poussé les développeurs de Click à opter pour Velocity réside dans la facilité d'apprentissage et d'utilisation de ce dernier puisque le nombre d'instructions qu'il comporte est assez réduit. En lisant un template, on devine ainsi aisément ce qu'il est censé faire, ce qui apporte un gain de productivité certain pour le développeur.

INSTALLATION ET CONFIGURATION

Le framework se présente sous la forme de 2 jars (`click-xxx.jar` et `click-extras-xxx.jar`) qu'il suffit d'ajouter dans le dossier `WEB-INF/lib` de son application web. La configuration de Click est facile et légère. Elle consiste en la déclaration de la servlet `ClickServlet` au sein du fichier `web.xml` et à la création d'un fichier `click.xml` à la racine du dossier `WEB-INF/`. Le plugin Eclipse ClickIDE qui vise à fournir un environnement de travail pour développer des applications Click est un axe de travail important pour les développeurs du projet. Facilitant la création de projets Click et de nouvelles pages, il fournit également un éditeur de templates Velocity qui apporte l'auto-complétion des instructions de ce dernier ainsi que celle des variables du modèle de la page ajoutées automatiquement par Click. Bref, il s'agit d'un outil intéressant pour gagner en productivité lors du développement d'applications Click.

MISE EN PRATIQUE

Après la théorie, place à la pratique ! Pour ce faire, nous allons réaliser une application web permettant de gérer une base d'utilisateurs. La gestion de la couche services est déléguée au conteneur léger Spring dont l'intégration avec Click réside dans l'utilisation de la servlet SpringClickServlet en lieu et place de la servlet ClickServlet dans le fichier web.xml. L'intégration entre Spring et la couche de persistance étant un sujet largement décrit dans la littérature, notre application se contentera d'un stockage mémoire des utilisateurs au sein d'une map. Les différentes pages de notre application se baseront sur un modèle commun au sein duquel seront définis des éléments tels que son menu ou son header. Click répond à cette problématique via la mise en place d'une hiérarchie de pages. Ainsi, nous allons créer une classe TemplatePage dérivant de la classe Page du framework. Elle va override la méthode getTemplate() ce qui lui permettra de définir le template Velocity qui servira de modèle aux différentes pages de l'application :

```
public class TemplatePage extends Page {
    // contrôles de la page
    private PageLink addUserLink = new PageLink("addUser", EditUser.class);
    private PageLink listUsersLink = new PageLink("listUsers", ListUsers.class);

    public TemplatePage() {
        // ajout des contrôles à la page car ils sont private ici
        addControl(addUserLink);
        addControl(listUsersLink);
    }

    @Override
    public String getTemplate() {
        return "/pages/template/template.htm";
    }
}
```

Les liens du menu de notre application sont définis au sein de ce template via des objets PageLink faisant référence à des classes associées à des pages logiques Click. Ces objets étant définis comme des champs privés, il faut les ajouter au sein de la page via la méthode addControl. La page d'ajout / modification d'un utilisateur est associée à la classe EditUser. L'utilisation du contrôle Form va permettre l'implémentation de cette fonctionnalité au sein d'un formulaire où les informations sur l'utilisateur devront être saisies. Parmi celles-ci, la saisie de la ville de provenance utilisera le contrôle AutoCompleteTextField de Click qui offre en standard un champ d'auto-suggestion effectuant une requête AJAX au fur et à mesure de la saisie utilisateur afin de proposer la liste des villes correspondante. Afin de bénéficier de l'injection de dépendances des services Spring au sein de notre classe EditUser, nous la déclarons en tant que bean managé via l'annotation @Component. Ainsi, les services userService et cityService, annotés par @Resource, seront directement injectés à l'exécution dans la classe EditUser. La différenciation du comportement de la page associée à la classe EditUser se fera grâce à l'utilisation d'un paramètre id passé à la page via la méthode GET. Ainsi, la page sera en mode édition si l'id passé dans l'URL correspond à celui d'un utilisateur existant. Le pattern Convention over Configuration appliqué

par Click permet une récupération transparente de la valeur du paramètre passé par l'URL pour peu qu'un champ de même nom soit annoté via @Bindable au sein de la classe associée à la page appelée. Click étant stateless par nature, la sauvegarde de l'id de l'utilisateur sur lequel les modifications devront être faites est réalisée au sein d'un champ caché ajouté au formulaire via le contrôle HiddenField. Le contenu de la classe EditUser est consultable sur notre site.

On remarque que les listeners d'action peuvent être créés soit en utilisant des inner classes anonymes implémentant l'interface ActionListener de Click soit en définissant le nom d'une méthode de la classe qui servira de handler via une chaîne de caractères. Plus longue, la première solution permet une vérification à la compilation et évite d'éventuelles erreurs à l'exécution qui pourraient survenir avec la seconde. La copie de données entre le contrôle Form et un POJO est prise en charge via des méthodes spécifiques au framework. En outre, les formulaires créés via Click bénéficient d'une validation automatique des champs et d'une gestion en standard des messages d'erreur éventuels sur ces derniers. Enfin, les contrôles Click ayant à leur charge leur propre rendu, le template Velocity correspondant à la page logique EditUser se révèle trivial :

```
#if ($msg)
    <div class="infoMsg"> $msg </div>
#end
$addUserForm
$jsElements
```

Notre page EditUser est basée sur un modèle de page commun définissant le menu de l'application, son header et son footer. Les autres pages présentées dans le menu de notre application ne sont pas détaillées dans cet article par manque de place. Cependant, elles suivent le même modèle en termes d'architectures que la page d'ajout / modification présentée. Le lecteur pourra consulter le code source complet de l'application sur le site du magazine afin de faire connaissance également avec le contrôle Table de Click utilisé dans la page listant les utilisateurs (figure 3). Lui aussi se révèle très puissant puisqu'il fournit en standard la pagination et le tri des données notamment.

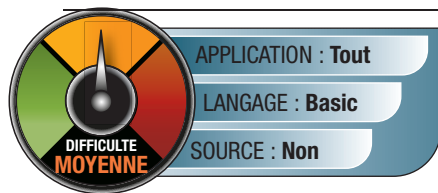
CONCLUSION

Au sein de la nouvelle génération de frameworks de développement web Java, Click se démarque puisqu'il ne propose pas de génération automatique de code par exemple. Au contraire, il se concentre sur une simplification de la couche présentation et un rapprochement vers les concepts initiaux du Web. Ainsi, son architecture stateless orientée composants et pages fournit une bonne encapsulation des concepts du Web. La prise en main et l'utilisation s'en trouvent grandement facilitées pour le développeur. En outre, ce dernier bénéficie des nombreux composants fournis en standard par le framework et voit donc sa productivité nettement améliorée. Les premières applications d'entreprise utilisant Click commencent à poindre et les premiers retours sont plus qu'intéressants pour l'avenir du framework. Seul l'avenir nous dira si Click peut arriver à s'imposer comme une alternative solide pour les applications d'entreprise mais quoi qu'il en soit, il faudra surveiller Click dans les années à venir !

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

La programmation avec PureBasic

J'étais très sceptique à l'idée d'utiliser ce langage à des fins professionnelles, très réticent même. Jusqu'au moment où j'ai eu l'occasion de découvrir des softs rédigés avec ce langage. Là, j'ai carrément été convaincu du bien fondé de ce système pour l'écriture d'applications vraiment professionnelles. En fait, je crois avoir retrouvé dans l'esprit, ce qu'était Turbo-Pascal dans les années 80 ! Bien entendu, tout cela au goût du jour avec les technologies actuelles ! Ce produit est simple à mettre en œuvre, léger, et demande peu de ressources au niveau du PC. Il pourrait même très facilement être utilisé sur un simple netbook !



APPLICATION : Tout
LANGAGE : Basic
SOURCE : Non

Vous avez tout, absolument tout pour commencer à programmer et pour un tout petit prix : 79 €. Vous disposez d'un éditeur de code performant, d'un éditeur visuel pour

créer vos fenêtres sous Microsoft Windows, d'un débogueur facile à utiliser, d'une aide contextuelle complète avec des exemples que vous pouvez très facilement récupérer par copier-coller...

Et tenez vous bien, pour ce prix, vous pouvez télécharger les versions 32 et 64 bits pour Windows, la version Mac OS X, les versions Linux 32 et 64 bits, et aussi la version Amiga....

Encore plus fort, vous disposez à vie des nouvelles versions et vous pouvez ainsi, sans supplément, avoir un environnement de développement toujours à la page, à la pointe de la technologie du développement ! [Fig.1].

En commençant à créer des programmes-exemples simples, on peut voir rapidement la puissance qui se cache derrière cet environnement de programmation ultra-puissant, car c'est bien d'un environnement de programmation très complet qu'il s'agit...

Mais quelles applications peut-on écrire avec PureBasic ? Et bien c'est simple, toutes sortes d'applications, des jeux, des applications de gestion, des applications d'informatique industrielles.... Facile à apprendre ? Oui, ce langage est simple à apprendre, de nombreux tutoriels en ligne existent, mais aussi l'aide intégrée qui est suffisante en elle-même dans bien des cas. De nombreux exemples de codes sont fournis avec le produit [Fig.2].

Bien que le pack de base soit suffisant, rien ne vous empêche d'ajouter des outils complémentaires à cet ensemble de développement au départ déjà très performant. Citons par exemple PureVision, un éditeur de fenêtres très complet qui permet de gérer le multi-fenê-

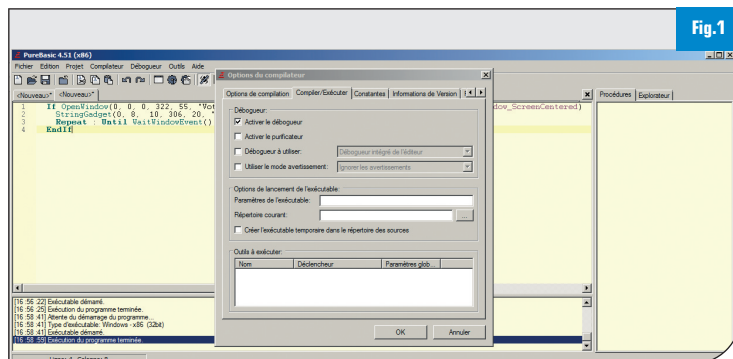


Fig.1

L'éditeur de PureBasic en action !

trage (produit payant) JaPBe qui est un éditeur de code source amélioré et totalement gratuit, Tailbite qui permet de rajouter des bibliothèques à l'environnement PureBasic....et plein d'autres choses encore....En fait, vous ajoutez si besoin toutes les bibliothèques que vous voulez, sachant qu'il y a déjà pas mal de choses !! De la gestion de la liaison série en passant par la génération de fichiers .PDF, sans parler de l'intégration complète des fonctions SQLite 3....vraiment, on est tous très gâtés dès le début ! PureBasic est fourni en effet en standard avec un ensemble de bibliothèques très complètes, qui permettent d'utiliser des bases de données, des bibliothèques de fonctions mathématiques, internet, et bien d'autres choses encore !

Les caractéristiques principales de PureBasic :

- Large panel de commandes internes permettant de créer tout type d'application ou de jeu.
- Support de tous les mots-clés du Basic standard, ce qui rend le langage très facile d'apprentissage...
- Compilateur très rapide qui crée des fichiers exécutables optimisés et compacts, ne nécessitant aucune DLL supplémentaire ! En effet, aucun fichier externe (DLL, exécuteur...) n'est nécessaire lors de la création du fichier exécutable
- Support des procédures pour une programmation structurée avec variables locales et globales
- Accès total aux API du système d'exploitation pour les programmeurs avancés.

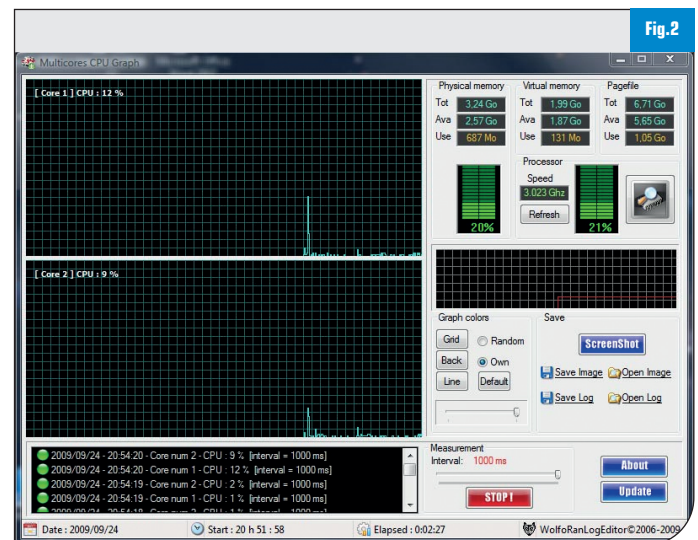


Fig.2

CPU Graph, une application écrite en PureBasic !!

- Utilisation optimale des possibilités de chaque machine par l'emploi de commandes spécifiques.
- Code source du programme immédiatement transposable sur Windows, AmigaOS ou Linux, et souvent Mac OS X...
- Éditeur de code et système de développement dédiés à PureBasic
- Debugger intégré permettant d'exécuter un programme pas à pas et de détecter facilement une erreur.

Franchement, si l'on fait le bilan de ce qui existe actuellement, difficile de trouver mieux, même en cherchant bien. Si vous voulez comme moi vous en convaincre, téléchargez la version de démonstration de ce produit, vous serez vous aussi rapidement conquis !
site : <http://www.purebasic.com>

PRÉSENTATION DE L'INTERFACE DE PROGRAMMATION

L'IDE de PureBasic est un environnement très complet qui permet la création et l'édition de fichiers sources, leur compilation, le débogage de l'application ainsi que la création de l'exécutable final.

La fenêtre principale de l'IDE se divise en 3 parties principales : [Fig.3].

1. La palette d'outils (menu en haut)

La palette contient plusieurs outils rendant la programmation plus facile et plus productive. Les outils affichés ici peuvent être configurés, déplacés ou même enlevés si besoin est. Voir Configurer l'IDE pour plus de renseignements.

2. La zone d'édition du code (en dessous de la barre d'outils)

Tous les codes sources sont affichés ici. Il est possible de passer de l'un vers l'autre à l'aide des onglets situés entre la barre d'outils et la zone d'édition.

3. Le rapport d'activité (situé en dessous de la zone d'édition, en bas)

Les erreurs de compilation ainsi que les messages d'erreur du débogueur seront affichés dans cette partie. Il peut être visible ou caché séparément pour chaque fichier.

Le reste de l'interface se compose du menu principal et de la barre d'outils. Cette dernière est simplement un raccourci des fonctions disponibles dans le menu et peut être complètement paramétrée (chaque fonction du menu peut être enlevée, déplacée ou ajoutée à

la barre d'outils). Pour savoir ce que fait un bouton, il suffit de laisser le pointeur de la souris dessus un court instant et une aide apparaîtra (et affichera la fonction du menu correspondante). Les commandes du menu sont expliquées dans une autre section.

Et pour créer les interfaces visuelles ?

Pour ma part, je préfère créer ces dernières directement dans le code, avec l'IDE de PureBasic, mais si vous préférez utiliser un éditeur d'interfaces, là, vous avez le choix et il n'y a pas une solution, mais plusieurs :

1. Le concepteur visuel d'interface fourni avec PureBasic, c'est le standard. Son utilisation est assez intuitive, simple et efficace [Fig.4]. Cette petite fiche n'a demandé que 5 minutes pour être conçue, c'est rapide, et le concepteur se charge ensuite de générer le code source « qui va bien » afin de pouvoir exploiter cette interface lors de la réalisation d'applications complètes [Fig.5 et 6].

2. Le concepteur visuel PureForm, gratuit lui aussi, téléchargeable sur le site : <http://gnozal.ucoz.com/#PureFORM>

C'est un concepteur d'interface un peu plus évolué, plus complet que le précédent, avec des fonctions supplémentaires au niveau du code généré.

3. Le concepteur PureVision, parfait si vous programmez des applications professionnelles, très complet mais payant (moins de 30 €) et de plus, il vous donne des bibliothèques supplémentaires pour vous permettre d'ajouter à vos applications des fonctionnalités avancées.

Ces 3 concepteurs d'interfaces sont écrits en PureBasic, et régulièrement, des mises à jour sont placées en ligne afin de tenir compte des évolutions de PureBasic, qui se dote d'année en année de fonctionnalités toujours plus abouties.

Mais quelle que soit votre décision, vous serez en mesure de créer des applications vraiment professionnelles, d'aspect soigné, compactes et ne nécessitant dans la plupart des cas aucune DLL ! Le code généré par PureBasic est natif, rapide et optimisé.

L'aide est conséquente, bien structurée, et largement agrémentée d'exemples complets.

De plus, des exemples, en dehors de l'aide sont là pour vous per-

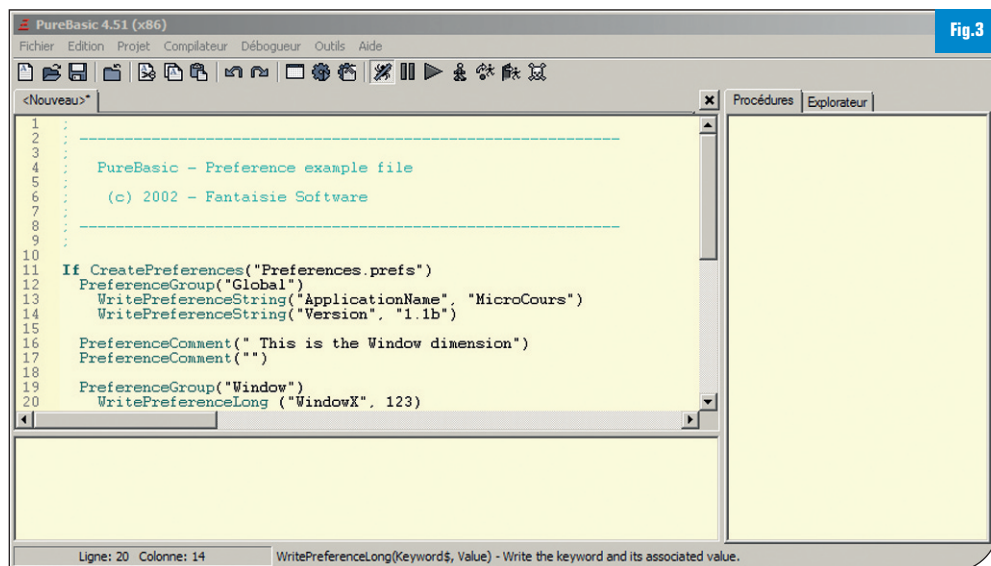


Fig.3



Fig.4



Fig.5

mettre de débiter dans les meilleures conditions...Et si cela ne vous suffisait pas, vous avez un forum tenu par des développeurs très sympathiques prêts à vous aider, à corriger votre code, et à vous donner tous les conseils indispensables pour mener à bien vos projets de développement. Et comme en achetant ce produit, vous bénéficiez des mises à jours gratuitement, - et pour toutes les plateformes utilisées par PureBasic (Windows, Mac OS X, Linux, Amiga) -, je peux vous dire que vous aurez la joie de découvrir à chaque fois de nouvelles fonctions intégrées dans le produit.

VOTRE PREMIER PROGRAMME PUREBASIC !

Restons dans la tradition ! Lançons l'environnement de développement. Voici un premier programme très simple en mode console, il suffit ensuite de le compiler avec le bouton circulaire en forme d'engrenage : [Fig.8].

Le code source de cette application très simple en mode console sera le suivant :

```
OpenConsole() ; ouvrir le mode console
PrintN("hello world") ; affichage
Repeat
  Until Inkey() <> "" ; si touche frappée, on ferme !
CloseConsole() ; fermer le mode console
```

N'en restons pas là, et faisons maintenant la même chose en développant cette fois une véritable application Windows.

Saisissons le code suivant dans une nouvelle fenêtre de l'éditeur :

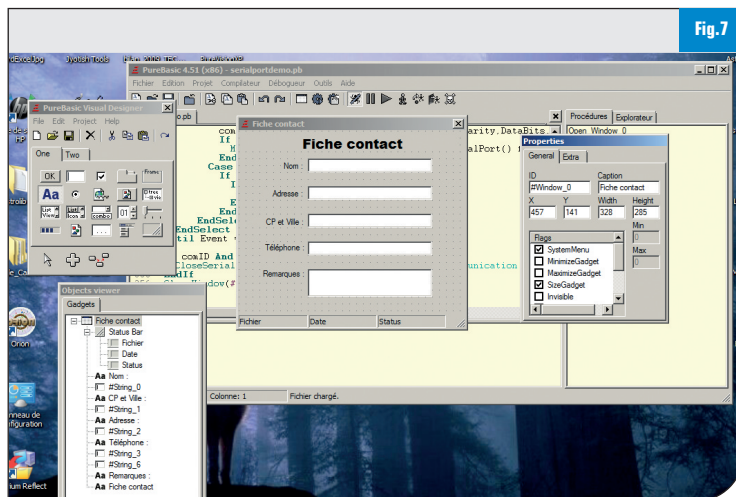
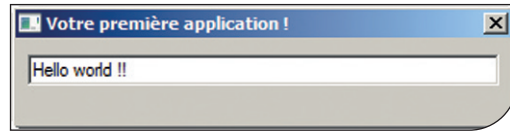


Fig.7

```
If OpenWindow(0, 0, 0, 322, 55, "Votre première application !",
#PB_Window_SystemMenu)
  StringGadget(0, 8, 10, 306, 20, "Hello world !!")
  Repeat : Until WaitWindowEvent() = #PB_Event_CloseWindow
EndIf
```

Et, de la même manière, après compilation : [Fig.9].



Et toujours dans la tradition : « Félicitations ! Vous venez de créer votre premier programme en PureBasic !! »

CONCLUSION

Voici un environnement complet de programmation, bon marché, qui ravira les débutants comme les professionnels. Un ensemble complet, encore peu connu, qui permettra au simple passionné de créer des merveilles, n'en doutons pas une seconde, et au professionnel confirmé de découvrir ici un ensemble léger, facile à mettre en œuvre, transportable sur une simple clé USB pour dépanner un client sur son site, très facilement. L'aide intégrée est très complète, un modèle du genre ! On y trouve toutes les explications relatives à une commande, des exemples, et tout cela dans un environnement « Made In France !! » de très haut niveau !

■ Philippe Georges
philippe.georges@sfr.fr

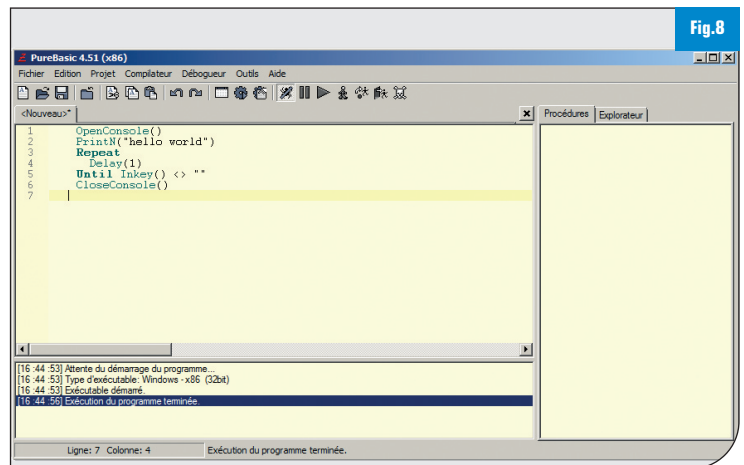


Fig.8

L'information permanente

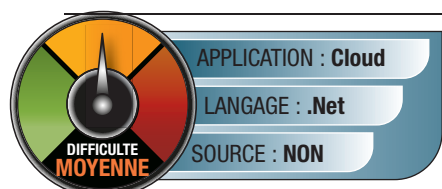
- L'actu de Programmez.com : le fil d'info quotidien
- La newsletter hebdo : la synthèse des informations indispensables.
Abonnez-vous, c'est gratuit !

www.programmez.com



Subversion dans les nuages

Une des fonctionnalités forts sympathiques de Windows Azure est de pouvoir y exécuter à peu près tout ce qu'on peut exécuter sur un serveur Windows. En résumé tout ce que vous pouvez faire tourner sur Windows Server, tournera sur Azure. C'est grâce à cela qu'on peut exécuter du Java sur Windows Azure. Dans cet article je vais vous présenter comment propulser votre serveur subversion dans les "nuages" grâce à Windows Azure.



APPLICATION : Cloud

LANGUAGE : .Net

SOURCE : NON

Pour ceux qui ne connaissent pas **Subversion**, c'est un système de gestion de versions centralisé, très populaire et open source, sous licence Apache.

Commencez par vous procurer les binaires d'un serveur Subversion pour Windows. Et là, vous avez l'embarras du choix <http://subversion.apache.org/packages.html> pour cet exemple j'avais pris la version 1.5.6 de collabnet. Pas besoin d'installer le serveur sur votre poste il vous faut juste extraire le dossier bin qui contient svnserve.exe et les dll associées. Ensuite on va uploader ce dossier dans les blobs de votre storage Azure. Pour cela j'ai utilisé le **cloudberry explorer** qui permet d'y naviguer à la manière d'un explorateur de fichier. Créez un container nommé "svn" et uploadez le contenu du dossier bin que vous avez extrait.

LE WORKER ROLE

Un Worker Role va récupérer les binaires stockés dans les blobs, les copier sur le local storage de la VM Azure, puis démarrer notre serveur SVN. Pour cela commençons par créer un nouveau Worker Role en utilisant les templates de Visual Studio.

Configuration du Worker Role

Ouvrez le ServiceDefinition.csdef

Rajoutez l'attribut enableNativeCodeExecution au WorkerRole, ainsi le WorkerRole peut exécuter du code non managé.

```
<WorkerRole name="WorkerRole" enableNativeCodeExecution="true">
```

A l'intérieur de cette balise insérez le code nécessaire pour déclarer l'utilisation des LocalResources

```
<LocalResources>
  <LocalStorage name="InstanceDriveCache" cleanOnRoleRecycle="false" sizeInMB="300" />
  <LocalStorage name="SvnFiles" cleanOnRoleRecycle="true" sizeInMB="300" />
</LocalResources>
```

Le LocalStorage que j'ai nommé SvnFiles sera pour les binaires de Svn. Et le InstanceDriveCache servira comme cache disque pour mon repository svn. A l'intérieur de la balise ConfigurationSettings ajouter :

```
<Setting name="SvnFilesContainerName" />
```

Cela permettra de déclarer le nom du container dans lequel j'ai stocké les binaires de SVN et de le changer par la suite sans modifier le

code de mon Worker Role. Et enfin dans les Endpoints rajouter comme cela un nouvel endpoint :

```
<Endpoints>
  <InputEndpoint name="WorkerIn" protocol="tcp" port="3690" />
</Endpoints>
```

Ainsi nous ajoutons un endpoint vers l'extérieur pour notre Worker Role, le port étant celui du protocole svn. Ouvrez le Serviceconfiguration.cscfg et ajoutez dans ConfigurationSettings :

```
<Setting name="SvnFilesContainerName" value="svn" />
```

pour déclarer le nom du container dans lequel j'ai stocké les binaires de svn. Vous pouvez aussi faire cette configuration en utilisant les écrans de propriété du WorkerRole (clic droit, puis propriété sur le WorkerRole dans le projet de configuration du service)

Le Worker Role

Un serveur SVN c'est bien, mais où seront stockés mes repository ? Nous allons utiliser un Windows Azure Drive (ou Xdrive). Le Windows Azure Drive c'est un disque NTFS mais dans les nuages. Il est stocké sous forme de blob dans le storage Azure et monté dans Azure par le Worker Role.

L'un des gros avantages du Windows Azure Drive est que mes repository SVN ne sont ainsi nullement liés à Azure. Le jour où je veux rapatrier mon serveur SVN et ses repository chez moi, c'est très simple il me suffit de recopier l'image disque (le .vhd) chez moi et de le remonter comme disque local (trois clics de souris sous Windows 7). Attention par contre en utilisant un Azure Drive on reste limité en termes de montée en charge, puisqu'on ne peut monter un Azure drive que sur une seule instance de notre Worker Role en lecture/écriture. Si on démarre plusieurs instances, notre Azure Drive ne sera disponible qu'en lecture sur les autres instances, et par conséquent notre serveur svn aussi. Donc la montée en charge se limite à augmenter la taille de la VM. Ouvrez le WorkerRole.cs dans la classe du WorkerRole, déclarez à l'intérieur de la classe cet objet :

```
private Process svnserve;
private string driveLetter;
private CloudDrive myCloudDrive;
```

L'objet svnserve nous servira à démarrer le processus du serveur SVN. Et le CloudDrive est l'objet correspondant au Windows Azure Drive. Dans la méthode OnStart() rajouter :

```
// Initialize config environment
```

```

        CloudStorageAccount.SetConfigurationSettingPublic(
            configName, configSetter) =>
        {
            configSetter(RoleEnvironment.GetConfiguration
                SettingValue(configName));
        });
    InstallAndStartSvn();

```

Les premières lignes sont pour initialiser la configuration, puis la méthode va copier et démarrer le serveur SVN. Voilà le contenu de la méthode InstallAndStartSvn : Les 2 premières lignes vont me permettre d'initialiser l'accès au blob

```

CloudStorageAccount storageAccount = CloudStorageAccount.From
    ConfigurationSetting("DataConnectionString");
CloudBlobClient blobfoldersClient = storageAccount.CreateCloud
    BlobClient();

```

Là, je récupère le chemin local à la VM de la LocalResource que j'ai déclarée dans ma configuration

```

String svnroot = RoleEnvironment.GetLocalResource("SvnFiles")
    .RootPath;

```

J'utilise à présent une classe qui va copier le contenu du container "Svn" dans le dossier correspondant à la LocalResource (La classe est disponible plus loin dans l'article)

```

StoreFoldersInBlobs.Folder SvnFolder = new StoreFoldersInBlobs
    .Folder(
        svnroot, blobfoldersClient.GetContainerReference(RoleEnvi
            ronment.GetConfigurationSettingValue("SvnFilesContainerName"))
    );
SvnFolder.CopyContainerToFolder();

Trace.WriteLine(" Copie du svn effectué :)");

```

Maintenant je monte mon Azure Drive.

```

MountAzureDrive();

Et je démarre mon serveur SVN.

int port = RoleEnvironment.CurrentRoleInstance.InstanceEndpoints
    ["WorkerIn"].IPEndpoint.Port;
svnserve = new Process()
{
    StartInfo = new ProcessStartInfo(Path.Combine(svnroot,
        @"svnserve.exe"),
        "-daemon" + " -root " + driveLetter + " -listen-port " + port)
    {
        UseShellExecute = false,
        WorkingDirectory = svnroot,
        RedirectStandardOutput = true,
        CreateNoWindow = true
    }
};
svnserve.Start();

```

```

Trace.WriteLine("Serveur Svn démarré");
}

```

A présent n'oubliez pas de rajouter dans la méthode Run du WorkerRole :

```

svnserve.WaitForExit();
Trace.WriteLine(" svn a fermé");

```

Sans quoi votre Worker Role redémarrerait tout de suite après avoir fini de démarrer, tournant en boucle indéfiniment de l'état Busy à Starting. Le code suivant va créer (s'il n'existe pas) et monter l'Azure Drive. Vous pouvez ainsi soit produire vos repository en local en réalisant votre vhd en local avec Windows 7, y créer un repository à l'aide de la commande «svnadmin create» y régler les droits appropriés en modifiant les fichiers de config du repository puis uploader votre vhd sur Azure toujours avec **le cloudberry explorer**. Le contenu de la méthode qui va monter le Windows Azure Drive pour le stockage du repository :

```

CloudStorageAccount storageAccount = CloudStorageAccount.From
    ConfigurationSetting("DataConnectionString");
LocalResource localCache = RoleEnvironment.GetLocalResource
    ("InstanceDriveCache");
CloudDrive.InitializeCache(localCache.RootPath + "cache", local
    Cache.MaximumSizeInMegabytes);
Trace.WriteLine(" cache créé");
// Just checking: make sure the container exists
CloudBlobClient blobClient = storageAccount.CreateCloudBlob
    Client();
blobClient.GetContainerReference("drives").CreateIfNotExist();

// Create cloud drive
myCloudDrive = storageAccount.CreateCloudDrive(
    blobClient
        .GetContainerReference("drives")
        .GetPageBlobReference("mysupercooldrive.vhd")
        .Uri.ToString()
);

try
{
    myCloudDrive.Create(64);
}
catch (CloudDriveException ex)
{
    Trace.WriteLine(ex.Message);
    // handle exception here
    // exception is also thrown if all is well but the drive
    // already exists
}
driveLetter = "";
try
{
    driveLetter = myCloudDrive.Mount(25, DriveMountOptions.Force);
}
catch (Exception e)
{
}

```

```
Trace.WriteLine(e.Message);
}
Trace.WriteLine("disque monté « + driveLetter);
```

Soit tout faire sur le cloud, en écrivant le code approprié. Voilà un exemple de code que vous pouvez rajouter dans votre Worker Role pour créer un repository après le démarrage du serveur :

```
Process svnad = new Process()
{
    StartInfo = new ProcessStartInfo(Path.Combine(svnroot,
        @"svnadmin.exe"), "create « + driveLetter + «test»)
    {
        UseShellExecute = false,
        WorkingDirectory = svnroot,
        RedirectStandardOutput = true,
        CreateNoWindow = true
    }
};
svnad.Start();
svnad.WaitForExit();
svnad.Dispose();
```

Vous pouvez aussi créer un Web Role avec une page d'administration qui exécutera ce code, qui permettra ainsi de produire de nouveaux repository. Et réaliser d'autres pages web d'administration qui permettront de gérer les droits de vos repository SVN. La classe qui sert à copier le contenu du container vers la LocalResource :

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using Microsoft.WindowsAzure.StorageClient;

namespace StoreFoldersInBlobs
{
    public class Folder
    {
        private string LocalPath;
        private CloudBlobContainer container;

        private string GetLocalPath(string uri)
        {
            string path = LocalPath;
            // skip domain and container
```

```
// note that this would almost certainly break running
// against dev storage (where paths look different)
foreach (var segment in new Uri(uri).Segments.
    Skip(2))
{
    path = Path.Combine(path, segment);
}
return path;
}

public Folder(string _localpath, CloudBlobContainer _blob
container)
{
    LocalPath = _localpath;
    container = _blobcontainer;
}

public void CopyContainerToFolder()
{
    var cloudBlobs = container.ListBlobs(new BlobRequest
Options()
{ UseFlatBlobListing = true, BlobListingDetails = Blob
ListingDetails.Metadata }).OfType<CloudBlob>();
    foreach (var blob in cloudBlobs)
    {
        var path = GetLocalPath(blob.Uri.ToString());
        Directory.CreateDirectory(Path.GetDirectoryName
(path));
        blob.DownloadToFile(GetLocalPath(blob.Uri.To
String()));
    }
}
}
```

Maintenant vous êtes prêt à propulser votre serveur SVN dans les nuages.

Mais ce n'est qu'une première étape, il serait possible d'y ajouter un serveur de build, de test et de mettre tout votre processus d'intégration continue dans les nuages. Mais cela est une autre histoire. Si vous êtes intéressé par cette idée, par un serveur SVN dans les nuages ou que vous désirez mettre votre serveur d'intégration continue dans les nuages n'hésitez pas à **me contacter**.

■ Aymeric Weinbach

Consultant Indépendant, aymeric@zecloud.fr, MCTS Web et Sql Server, créateur de zecloud.fr la communauté Open Source Azure qui organise régulièrement des Azure camps

Nouveau : économisez jusqu'à 50%

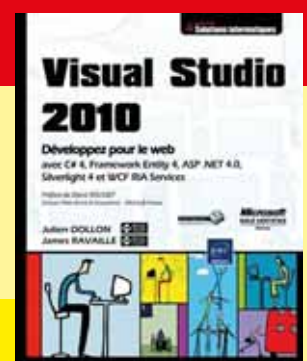
Abonnement 2 ans au magazine + 1 livre numérique ENI

• **79€ 2 ans au magazine** au lieu de 130,90 (valeur de 22 numéros) *Tarif France métropolitaine*
+ un livre d'une valeur de 23,9 € à 31,9 €, soit un total de 154,8 € à 162,8 €

• **89€ 2 ans au magazine + archives sur Internet et PDF + 1 livre numérique ENI**

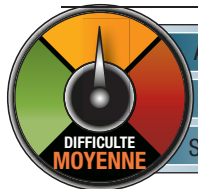
Livres à Choisir : • Visual Studio 2010 • PHP5.3 • Bing Maps • MySQL 5, Administration et optimisation
• Java et Spring, Concevoir, construire et développer une application Java/J2EE avec Spring.

Sur www.programmez.com/abonnement.php



Apprendre à construire et à utiliser des tables views

Dans cet article nous allons apprendre à manipuler cet objet à l'aide d'un exemple d'application permettant de gérer une liste de courses que nous enrichirons par étapes successives.



APPLICATION : iOS

LANGAGE : Objective-C

SOURCE : Oui

Pour utiliser une table view, il est nécessaire de créer une instance de la classe **UITableView**, de la configurer puis de l'ajouter à une autre vue en tant que sous-vue.

Chaque ligne d'une table view est considérée comme une cellule. Une cellule est une instance de la classe **UITableViewCell** et est composée de cinq différentes sous-vues :

- **backgroundView** : Le fond de chaque ligne
- **selectedBackgroundView** : Remplace la **backgroundView** lorsque la ligne est sélectionnée
- **image** : une image à la gauche de la cellule
- **accessoryView** : Une vue personnalisable à la droite de la cellule
- **contentView** : une vue personnalisable entre l'image et l'**accessoryView**

On entrevoit alors tout le potentiel de personnalisation de chaque cellule, réalisable en modifiant ces différentes vues. Le remplissage des données affichées se fait en respectant le protocole **UITableViewDataSource** ; le contrôle de l'apparence de la table se fait avec le protocole **UITableViewDelegate**.

CRÉATION D'UNE TABLEVIEW

Maintenant que nous avons introduit les principaux objets, commençons pas créer une table qui va afficher une liste de courses prédéfinies. Pour cela, nous allons implémenter les deux méthodes nécessaires du protocole **UITableViewDataSource** et utiliser l'application delegate comme source de données. Nous stockerons les données sous la forme de chaînes de valeurs dans une instance de type **NSMutableArray** nommée « mesCourses ». La syntaxe en Objective-C pour adopter un protocole consiste à définir directement après sa super classe le protocole suivant : **NSObject<UITableViewDataSource>**. L'exemple ci-dessous montre que la classe **TabViewAppDelegate** gère la vue et agit comme source de données principale. Nous définissons également une propriété pour la source de données nommée « **dataSource** ».

```
#import <UIKit/UIKit.h>
#import <Foundation/Foundation.h>

@interface TabViewAppDelegate : NSObject <UITableViewDelegate,
UITableViewDataSource> {
    UIWindow      *window;
    UITableView    *maTable;    // Ma table view
    NSMutableArray *mesCourses; // Ma liste de courses
}

@property(n nonatomic, assign) id <UITableViewDataSource> data
```

Source;

@end

L'initialisation des valeurs de notre liste de courses est placée dans la méthode **applicationDidFinishLaunching**.

Nous créons notre tableview et initialisons sa taille à celle de l'application afin qu'elle occupe l'espace maximal possible, puis nous complétons notre liste de courses.

Enfin nous indiquons que c'est notre application delegate qui sera la source de données de notre table view.

```
#import «TabViewAppDelegate.h»

@implementation TabViewAppDelegate

@synthesize window;
@synthesize maTable;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    UIWindow *window = [[UIWindow alloc] initWithFrame:[UIScreen mainScreen] bounds];

    CGRect frame = CGRectMake(0, 70, 320, 420);
    maTable = [[UITableView alloc] initWithFrame:[UIScreen mainScreen].applicationFrame style:UITableViewStylePlain];
    mesCourses = [[NSMutableArray arrayWithObjects:@"Soupe", @"Crudités",
        @"Pâtes",
        @"Pizza",
        @"Lentilles",
        @"Yahourt",
        @"Gâteau au chocolat",
        @"Crème brûlée",
        nil] retain];
    maTable.dataSource = self;
    [window addSubview:maTable];
    [window makeKeyAndVisible];
}
```

Compilons notre code et exécutons-le dans notre simulator iPhone. Nous constatons que notre liste de valeurs est vide. En effet nous n'avons pas à ce stade, implémenté les deux méthodes du protocole

UITableViewDataSource que sont `numberOfRowsInSection:` et `cellForRowAtIndexPath:`.

```
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:
InSection:(NSInteger)section{
    return [mesCourses count];
}
```

La méthode **numberOfRowsInSection** définit pour une table et une section, le nombre de lignes associées. Par défaut, nous n'avons qu'une seule section et nous retournons donc le nombre de lignes de notre liste de courses.

La méthode **cellForRowAtIndexPath:** est invoquée par la table pour récupérer la représentation de chaque cellule. C'est généralement dans cette méthode que nous personnalisons le contenu des cellules. Nous recherchons d'abord dans le cache si une cellule n'a pas déjà été créée pour cette ligne.

Si ce n'est pas le cas alors nous créons la cellule. Une fois la cellule récupérée ou créée nous lui affectons la valeur correspondante issue de notre liste de courses.

Vous constaterez que désormais à l'exécution la liste des courses s'affiche correctement dans notre simulateur.

AJOUT D'IMAGES

Chaque ligne de notre tableau contient un article à acheter, il peut être intéressant d'afficher la représentation visuelle de chaque article. Pour cela nous allons afficher une image à la gauche de chaque cellule.

```
NSString *imageName = @"salade.jpg";
cell.imageView.image = [UIImage imageNamed:imageName];
```

Pour mettre à jour l'image d'une cellule, il faut positionner dans la propriété `image` de l'`UIImageView` de la cellule une instance de type **UIImage**. Cette modification est à intégrer dans la méthode **cellForRowAtIndexPath**.

SUPPRESSION DE LIGNES

L'insertion, la suppression ou le déplacement de ligne se fait en basculant la table view en mode édition. Pour cela, nous allons ajouter un bouton « Supprimer » en haut de notre fenêtre permettant à l'utilisateur d'activer ce mode.

Nous créons donc un bouton nommé « Supprimer » que nous ajoutons à la fenêtre principale et définissons la méthode `suppAction` qui sera invoquée lorsque l'on cliquera dessus.

```
boutonSuppression = [[UIButton buttonWithType:UIButtonTypeRoundedRect] retain];
boutonSuppression.frame = CGRectMake(200.0, 25.0, 100, 35);
[boutonSuppression setTitle:@"Supprimer" forState:UIControlStateNormal];
[boutonSuppression addTarget:self action:@selector(suppAction:)
forControlEvents:UIControlEventTouchUpInside];
[window addSubview:boutonSuppression];
```

La méthode `suppAction` que nous avons définie est chargée d'assurer la bascule entre les deux modes édition/non édition.

Si le titre du bouton cliqué est « Supprimer » alors nous activons l'édition via la méthode **setEditing** avec le paramètre `editing` à YES et

animé à YES pour animer le changement d'état. Dans le cas contraire, nous passons en mode non édition.

```
-(void)suppAction:(id)sender{

    if(sender == boutonSuppression){
        if([boutonSuppression.currentTitle isEqualToString:@"Supprimer"]
        == YES)
        {
            [boutonSuppression setTitle:@"Retour" forState:UIControlStateNormal];
            [maTable setEditing:YES animated:YES];
        }
        else
        {
            [boutonSuppression setTitle:@"Supprimer" forState:UIControlStateNormal];
            [maTable setEditing:NO animated:YES];
        }
    }
}
```

Pour permettre à un utilisateur de supprimer une ligne de sa liste de courses, il est nécessaire d'implémenter les deux méthodes `tableView:editingStyleForRowAtIndexPath:` et `tableView:commitEditingStyle:forRowAtIndexPath:`. La méthode `editingStyleForRowAtIndexPath:` est appelée par la table view et retourne les actions autorisées par l'utilisateur (insertion, suppression) pour chaque ligne. Sans définition de cette méthode par défaut le mode d'édition est supprimé et donc toutes les lignes sont marquées avec un cercle rouge avec le signe -. La méthode **commitEditingStyle** permet d'être notifiée lorsque l'utilisateur a effectué une opération d'insertion ou de suppression d'une ligne.

Si le mode d'édition est de type `UITableViewCellEditingStyleDelete` alors nous procédons à la suppression de la ligne sélectionnée par l'utilisateur. La suppression d'une ligne se traduit par une suppression de celle-ci à la fois dans notre tableau de courses mais aussi dans la table view. L'appel à la méthode **removeObjectAtIndex** permet de supprimer la ligne de notre tableau `mesCourses`. La méthode **deleteRowsAtIndexPaths** permet de supprimer la ligne du table view avec un effet d'animation lors de la suppression (`withRowAnimation:UITableViewRowAnimationLeft`).

AJOUT DE SECTIONS

Un autre mode de présentation très pratique est l'utilisation de sections. Les sections permettent de regrouper des éléments d'une même catégorie ensemble. Nous allons créer trois sections : la première pour les entrées, la seconde pour les plats et la troisième pour les desserts. La méthode **numberOfRowsInSection** permet de définir par section le nombre de lignes correspondant.

```
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:
InSection:(NSInteger)section{
    if(section == 0){
        return 2;
    }
    else return 3;
}
```

Nous allons également définir une nouvelle méthode `titleForHeaderInSection:` pour ajouter un titre de début de section.

```
(NSString *)tableView:(UITableView *)tableView titleForHeaderInSection:(NSInteger)section{

    if(section == 0){
        return @"Les entrées";
    }
    else if(section == 1){
        return @"Les plats";
    }
    else return @"Les desserts";
}
```

Nous modifions enfin la méthode `cellForRowAtIndexPath` afin de retourner par section la valeur de cellule. Comme nous manipulons dans cet exemple un seul tableau de courses, nous ajoutons +2 et +5 à notre indice afin de récupérer les aliments correspondant aux sections plats et desserts. Dans l'absolu, il aurait été pertinent de définir un tableau de courses par section.

```
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{

    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:@"courses"];

    if (cell == nil) {
        cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:@"courses"] autorelease];
    }

    // Les entrées
    if(indexPath.section == 0) {
        cell.textLabel.text = [mesCourses objectAtIndex:indexPath.row];
    }

    // Les plats
    if(indexPath.section == 1) {
        cell.textLabel.text = [mesCourses objectAtIndex:indexPath.row+2];
    }

    // Les desserts
    if(indexPath.section == 2) {
        cell.textLabel.text = [mesCourses objectAtIndex:indexPath.row+5];
    }

    return cell;
}
```

AFFICHER DES DONNÉES HIÉRARCHIQUES

Dans certains cas, il peut être intéressant d'afficher des données sous une forme hiérarchique. Pour cela chaque cellule dispose de deux propriétés : `indentation level` et `indentation width`, permettant de gérer une indentation.

La multiplication des deux champs produit la valeur de décalage, en pixels, du contenu. Le contenu sera décalé vers la droite s'il s'agit de

valeurs positives et vers la gauche pour des valeurs négatives. En ajoutant ces deux lignes de code nous provoquons un décalage de 10 pixels entre chaque ligne de notre table, le résultat est un affichage des données en escalier.

```
cell.indentationLevel = indexPath.row;
cell.indentationWidth = 10.0f;
```

PERSONNALISATION D'UNE CELLULE

Rappel : une cellule d'une table view est constituée de plusieurs vues que l'on peut modifier afin de personnaliser le contenu d'une cellule. Nous allons modifier chaque cellule de la méthode `cellForRowAtIndexPath` afin de lui ajouter :

- un titre informatif correspondant à la section et la ligne courante
- une image à la droite de la cellule

```
UILabel *sectionLabel;
...
// Affichage de la section et de la ligne courante
sectionLabel = [[[UILabel alloc] initWithFrame:CGRectMake(180, 15, 90, LABEL_HEIGHT)] autorelease];
sectionLabel.backgroundColor = [UIColor clearColor];
sectionLabel.textColor = [UIColor grayColor];
sectionLabel.highlightedTextColor = [UIColor blueColor];
sectionLabel.font = [UIFont boldSystemFontOfSize:12];
sectionLabel.text = [NSString stringWithFormat:@"S=%c,L=%c",indexPath.section+'0',indexPath.row+'0'];
[cell.contentView addSubview:sectionLabel];
```

Nous créons un label de type `UILabel` pour contenir notre titre informatif de type « Section=1,Ligne=1 ». Nous adaptons sa taille pour un affichage réduit, définissons la couleur du texte à gris puis le rajoutons comme sous-vue à la vue `ContentView`. La `ContentView` est affichée entre l'image de gauche et l'`accessoryView` de droite.

```
// Ajout du bouton pour afficher le détail
UIButton *button = [UIButton buttonWithType:UIButtonTypeDetailDisclosure];
[button setTitle:@"Suite" forState:UIControlStateNormal];
[button addTarget:self action:@selector(Detail:) forControlEvents:UIControlEventTouchUpInside];
cell.accessoryView = button;
```

Nous créons un bouton de type `UIButtonTypeDetailDisclosure`. Ce bouton pourrait ainsi être utilisé dans le futur pour afficher un autre vue et visualiser des informations sur chaque plat. Nous affectons ce bouton à la vue `AccessoryView`. Rappel : l'`accessoryView` est la zone de droite d'une cellule. Nous exécutons notre exemple et constatons que, désormais, chaque cellule contient un titre informatif en gris et une image détail à sa droite.

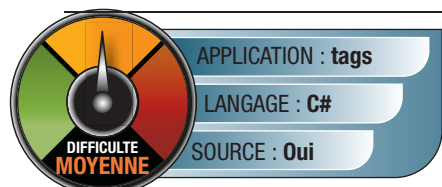
CONCLUSION

Dans cet article, nous avons vu comment créer, manipuler et personnaliser une liste d'éléments en utilisant l'objet graphique `UITableView`. Les possibilités sont infinies pour celui qui sait faire un peu de graphisme et souhaite proposer un design innovant pour sa future application iOS.

■ Christian Peyrusse

Connectez le monde au web avec Microsoft Tag

Nous sommes bien loin de Kinect, et pourtant, ce produit permet de « connecter » le monde réel au monde virtuel, et en particulier le web. Découvrez ce qu'est Microsoft Tag, comment l'utiliser et comment l'intégrer dans vos sites web et/ou applications.



Microsoft Tag est un ensemble de services autour d'une technologie de code barre 2D. Dans la lignée des QR Codes, les MS

Tag vous permettent une per-

sonnalisation et une flexibilité sans précédent. De plus, il suffit d'un Smartphone avec une caméra pour utiliser ce service. Il est composé :

- D'une technologie de code barre 2D, appelée HCCB (*High Capacity Color Barcode*),
- De logiciels de capture et de lecture pour téléphones mobiles,
- D'un service en ligne permettant de générer, modifier et suivre l'utilisation des tags.

Ce dernier point est l'un des principaux différenciateurs de ce produit. En effet, ce service en ligne vous permet de faire vivre vos tags (c'est-à-dire, de modifier des informations après la création, de les activer/désactiver, etc.), et d'avoir des statistiques détaillées sur leurs usages. A l'heure actuelle, plus de deux milliards de Microsoft Tags ont été générés.

POURQUOI PEUT-ON S'EN SERVIR ?

Tout d'abord, il faut savoir que vous pouvez stocker 4 types de données dans des Tags : messages (texte libre), URL, carte de visite et n° de téléphone. Le type de tag et les informations associées (comme votre nom et prénom dans le cas d'un tag Carte de visite) ne sont pas stockés dans le tag lui-même, mais sur les serveurs de Microsoft : le tag ne contient qu'un identifiant. Cela vous permet, même une fois le tag imprimé ou diffusé, de modifier les informations qu'il contient.

A savoir : Pour l'heure, toutes les fonctionnalités présentes dans Microsoft Tag sont, et resteront gratuites, y compris le logiciel de lecture. Certains services futurs pourraient, quant à eux, devenir payants.

COMMENT L'UTILISER ?

Il suffit de trois étapes pour utiliser Microsoft Tag : la création, la diffusion et l'analyse. La première chose à faire est de se rendre sur le Tag manager (<http://tag.microsoft.com>) afin de créer un compte gratuitement. A partir de ce compte, vous aurez la possibilité de gérer vos tags et afficher des rapports détaillés d'utilisation [Fig.1]. Une fois votre tag créé, vous pouvez en effectuer le rendu en cliquant sur le mini tag dans la colonne de droite. Le Tag manager vous permet non seulement de choisir la taille et le format de votre tag (au choix pdf, jpeg, png, gif, tiff ou wmf), mais également de personnaliser le rendu. Vous pouvez le rendre en couleur, en noir et blanc, ou le personnaliser avec une image de fond. Dans ce dernier cas, vous devrez passer par un logiciel de design pour ajouter l'image. Une fois rendu, il faut diffuser votre tag par tous les moyens possibles : magazines, flyers, sites internet, cookies au chocolat (cela a déjà été fait). Microsoft a développé un lecteur unique pour tous les tags, et ce sur une multitude de plateformes : Windows Mobile et Windows Phone 7 bien sûr,



Le même tag, rendu personnalisé, classique et noir et blanc

mais également iPhone, Android, Blackberry, Symbian et une version Java ME. Il suffit de se rendre sur <http://www.gettag.mobi> pour télécharger gratuitement le lecteur. Les lecteurs Tag sont simples d'utilisation : il suffit de « taguer » avec la caméra pour afficher son contenu ou être redirigé vers l'Url indiquée dans le Tag Manager. Une fois le tag diffusé, vous pouvez suivre en détail le scan de vos tags grâce à un ensemble de rapports : par tag, géographiquement sur une carte, les scans du jour, ou bien encore par catégorie.

MICROSOFT TAG POUR LES DÉVELOPPEURS

Le service Microsoft Tag peut donc être utilisé de manière autonome, sans faire appel à un logiciel ou un développeur. Cependant, vous pouvez également les intégrer à vos applicatifs existants. Microsoft fournit une API sous forme de service web SOAP pour accéder aux fonctionnalités du Tag Manager, comme la création de tag, le rendu image ou l'activation/désactivation des tags existants.

LES WEB SERVICES

Avant de pouvoir utiliser ces web services, vous devez demander l'accès à l'API en remplissant un formulaire sur le site de Microsoft Tag (<http://tag.microsoft.com/ws/accessrequest.aspx>). Vous recevrez rapidement un email contenant un jeton d'accès indispensable pour utiliser l'API. Le service web étant en SOAP, il est simple de l'utiliser depuis un grand nombre de langages. Nous allons nous pencher sur l'utilisation depuis deux langages : C# et PHP.

L'API C#

L'utilisation des API MS Tag est très simple : il suffit d'ajouter une référence web vers l'adresse du service (<https://ws.tag.microsoft.com/MIBPSERVICE.wsdl>), et de l'utiliser. L'exemple de code suivant permet de créer une catégorie et un tag de type Url dans cette catégorie (retrouvez un exemple complet dans les sources de l'article) :

```
using (MIBPContractClient tagServiceClient = new MIBPContractClient())
{
    // Chargement des autorisations
    UserCredential credential = new UserCredential();
    credential.AccessToken = System.Configuration.ConfigurationManager.AppSettings["MSTagApiToken"];

    // 1. Création d'une nouvelle catégorie
```

```

Category categorie = new Category();
categorie.Name = "C# Conference";
categorie.UTCStartDate = DateTime.UtcNow;

tagServiceClient.CreateCategory(credential, categorie);

// 2. Création d'un tag URL
URITag uriTag = new URITag();

uriTag.Title = "Bewise Labs";
uriTag.MedFiUrl = "http://labs.bewise.fr";
uriTag.UTCStartDate = DateTime.UtcNow;

tagServiceClient.CreateTag(credential, categorie.Name, uriTag);
}

```

A savoir : Lors de la première exécution, vous aurez certainement une erreur vous indiquant que la réponse reçue excède la capacité de traitement. Cela est dû à la configuration du proxy du service web. Rendez-vous dans le fichier de configuration de votre application et, dans la section `basicHttpBinding`, augmentez les valeurs de `maxBufferSize`, `maxReceivedMessageSize`, `maxArrayLength` et `maxNameTableCharCount`.

L'API PHP

Il est tout à fait possible de faire appel à des services SOAP en PHP, que cela soit avec les fonctions natives de PHP, ou avec des bibliothèques telles que `nusoap`. Vous pouvez également utiliser un projet de la communauté : **Microsoft Tag Php SOAP Interface** (<http://mstagrest-lib.codeplex.com/>). Distribué sous la forme d'un fichier php à inclure, il vous donne accès à un ensemble de classes PHP vous permettant d'accéder facilement aux services exposés par l'API, sans vous soucier des appels ou de la construction des messages SOAP.

L'exemple suivant reprend l'exemple de l'API C# vu précédemment :

```

<?php
require_once('MSTag.php');

$msTag = new MSTag();

// Chargement des autorisations
$MSTagAuthToken = ""; // TODO : Remplacer par votre token MS Tag
$userCredential = new UserCredential($MSTagAuthToken);

// 1. Création d'une nouvelle catégorie
$category = new Category();
$category->setName('C# Conference');
$category->setUTCStartDate();

$result = $msTag->CreateCategory($userCredential, $category);

// 2. Création d'un tag URL
$tag = new URITag();
$tag->setTitle('Bewise Labs');
$tag->setMedFiUrl('http://labs.bewise.fr');
$tag->setUTCStartDate();

$result = $msTag->CreateTag($userCredential, 'TestCategory2', $tag);

?>

```

Il vous est ainsi tout aussi facile de vous intégrer que votre application soit réalisée en C#/VB.net ou en PHP.

L'API MS TAG, UN CAS PRATIQUE

Nous allons maintenant utiliser l'API MS Tag dans un cas pratique. Pour cet exercice, nous allons prendre l'exemple d'un événement informatique. Notre application permettra de gérer à la fois les inscriptions des visiteurs et un concours. La partie inscription, pour notre cas pratique, restera simple : un seul formulaire, avec les informations de base. Une fois validé, le visiteur pourra imprimer son badge. D'un point de vue des MS Tags, nous allons tout d'abord les créer (comme nous l'avons vu précédemment), puis les afficher. La reconnaissance du tag se fera par les applications sur le téléphone. Le tag contiendra une url unique vers le site d'inscription indiquant si le visiteur possède un ticket gagnant ou non. Nous avons déjà vu tous les éléments nécessaires à la réalisation de cette application, à l'exception de l'affichage dans un site web du tag généré. Il y a de nombreuses manières d'adresser ce besoin. L'une des solutions est de créer un Handler pour afficher les tags en fonction de la catégorie et le nom.

```

public void ProcessRequest(HttpContext context)
{
    context.Response.ContentType = "image/jpeg";

    string tagCollection = context.Request.QueryString.Get("collection");
    string tagTitle = context.Request.QueryString.Get("title");

    if (string.IsNullOrEmpty(tagCollection) || string.IsNullOrEmpty(tagTitle))
        return;

    using (MIBPContractClient tagServiceClient = new MIBPContractClient())
    {
        // Chargement des autorisations
        UserCredential credential = new UserCredential();
        credential.AccessToken = System.Configuration.ConfigurationManager.AppSettings["MSTagApiToken"];

        byte[] tagImage = tagServiceClient.GetBarcode(credential,
            tagCollection,
            tagTitle,
            ImageTypes.jpeg,
            1,
            DecorationType.HCCBRP_DECORATION_NONE,
            false);

        context.Response.BinaryWrite(tagImage);
    }
}

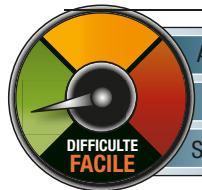
```

Microsoft Tag est à la fois une technologie très intéressante, ludique et simple d'emploi. Que ce soit pour l'utiliser de manière ponctuelle – en créant le tag depuis le tag manager – ou pour l'intégrer dans un applicatif, vous avez toutes les clés en main. Il ne vous reste plus qu'à trouver l'usage qui vous convient.

■ **Christophe Maneu**

Mettre en œuvre l'API Google Earth

Visiter Mars, la Lune, les constellations, ou plus simplement notre bonne vieille terre, d'un continent à l'autre, ou des villes d'un monument à l'autre, voilà ce que permet l'API Google Earth, avec simplement quelques lignes de Javascript et de XML



APPLICATION : Géoloc

LANGUAGE : Web

SOURCE : Oui

Google Earth est au départ une reconstitution de la terre sur la base de photographies aériennes et satellites. Puis Google a étendu cela à l'espace ainsi

qu'à la Lune et à Mars, à partir des données de la Nasa. Et Google ne cesse d'étendre son Google Earth. Ainsi, si nous ne parlons ici que de Google Earth 5, au moment de la rédaction de cet article Google proposait une version 6 en bêta qui intègre Street View. Au final l'ensemble devient un outil fascinant, programmable à travers une API, et qui peut permettre d'offrir à travers un navigateur des applications à vocation pédagogique, ou des jeux superbes, avec très peu d'investissement pour le programmeur. Découvrons ensemble les bases du maniement de l'API Google Earth.

1 PRÉ-REQUIS ET OUTILS

Pour programmer Google Earth, il faut avoir des bases de Javascript et de XML. Il est nécessaire de comprendre ce que sont des coordonnées géographiques, à savoir latitude et longitude et avoir quelques notions de géométrie, et c'est tout. Au niveau des outils, il suffit de télécharger et d'installer le plugin Google Earth depuis <http://www.google.fr/intl/fr/earth/index.html>. Pour écrire le code, n'importe quel éditeur capable de travailler avec HTML, XML et Javascript sera bienvenu. Enfin, on installera sur la machine de développement un serveur Web quelconque. Apache est a priori tout trouvé. Sous Windows, une solution rapide est d'installer un EasyPHP. Les exemples de cet article sont disponibles sur notre site et considèrent être déposés dans un répertoire 'ge' du serveur web. Vous devrez éventuellement adapter ceci à votre configuration.

2 HELLO WORLD !

C'est vraiment l'occasion ou jamais :) Pour commencer à travailler avec Google Earth, il faut obtenir une API Key. Pour cela il suffit de se rendre sur le site développeur de Google Earth, à <http://code.google.com/intl/fr-FR/apis/earth/> et de demander la clé. Pour l'obtenir, vous devez avoir un compte Gmail. Cette clé permet d'utiliser le système sans limitation dans le cadre d'applications non commerciales. Les applications étant destinées à être sur Internet, vous devrez donner votre domaine pour générer une clé qui ne fonctionnera qu'avec lui. Pour la phase de développement en local, donnez simplement <http://localhost>. Au moment de la mise en ligne de votre application, vous devrez demander une autre clé et modifier tout vos scripts. Une application Google Earth consiste en une page HTML. Cette page contiendra une balise DIV qui sera le réceptacle de plugin Google Earth. On placera aussi dans la page un code Javascript minimum pour charger et initialiser le plugin. Voici ce que cela donne :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
  Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
  transitional.dtd">
<html>
<meta http-equiv="content-type" content="text/html; charset=
  utf-8" />
<head>
<title>Programmez!</title>
<script src="http://www.google.com/jsapi?key=xxxx "></script>

<script>

var ge = null;

function init() {
  google.earth.createInstance('helloworld',
    initCallback, failureCallback);
}

function initCallback(object) {
  ge = object;
  <div id="helloworld" style="border: 1px solid silver;
    height: 500px; width: 500px;">
}

function failureCallback(object) {
  alert('Erreur lors de la création de l'instance de Google Earth');
}

google.load('earth', '1');
google.setOnLoadCallback(init);
</script>

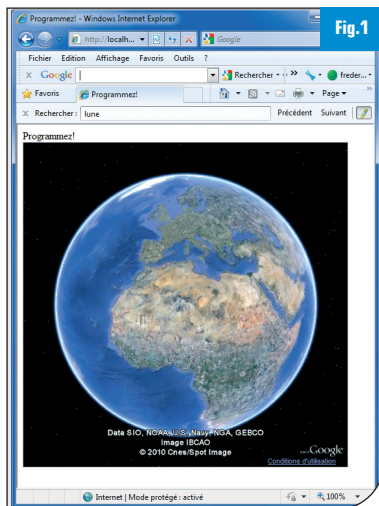
</head>
<div id="helloworld">
  <div id="helloworld" style="border: 1px solid silver;
    height: 500px; width: 500px;">
  </div>
</div>
</html>
```

Lorsque le navigateur charge cette page, il va exécuter le script inclus et donc la fonction google.load qui va charger le plugin. La valeur 1 passée en argument correspond à la version de l'API. Celui-ci doit ensuite être initialisé. On pourrait le faire dans l'événement onload de la balise body. Cependant Google recommande d'utiliser la méthode setOnLoadCallback qu'il fournit, recommandation que suit notre exemple. Ceci aboutit à l'invocation de la fonction init, qui est

de fait une fonction de rappel (callback), et qui crée une instance du plugin qui sera attachée à la balise div qu'on lui fournit en premier argument. Notre fonction *init* attend également en argument deux autres fonctions de rappel. La première recevra du système un objet qui encapsule tout Google Earth. Dans cette fonction, nous affectons cet objet à une variable pour usage ultérieur et nous rendons l'interface visible à l'utilisateur [Fig.1]. La seconde fonction de rappel sera invoquée en cas d'erreur.

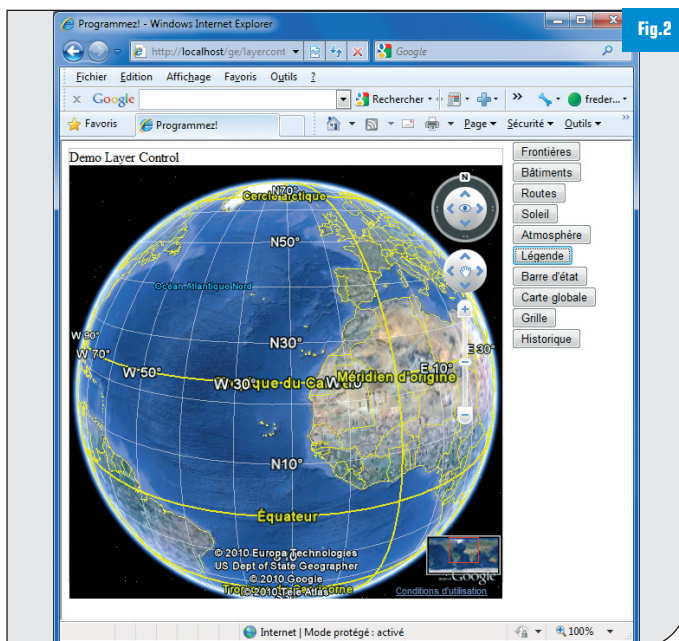
3 ENRICHIR L'INTERFACE AVEC LAYERS ET CONTRÔLES

En l'état le plugin est fonctionnel, mais son interface est un peu rudimentaire. Il est possible de faire tourner le globe terrestre avec le bouton gauche de la souris, de zoomer avec la molette ou déplacer la souris en maintenant le bouton droit enfoncé. Mais tout cela peut-être considérablement enrichi. Tout d'abord par un contrôle de navigateurs qui permettra de zoomer, de se déplacer à volonté. Le contrôle montre aussi une boussole, qui se révèle extrêmement utile lorsqu'on se déplace près du sol. La direction selon laquelle on se déplace, est également la direction selon laquelle on regarde, et



Un hello World qui mérite bien son nom :-)

cela s'appelle heading dans le jargon de Google Earth. Il est aussi possible de faire apparaître des surcouches de représentation qui répondent au terme générique de layers. Ainsi il est possible de délimiter les frontières des pays, d'afficher ou non un effet lumineux dû à l'atmosphère terrestre, ainsi que la grille de coordonnées géographiques composée par les méridiens et parallèles. Apparaissent renforcés le méridien de Greenwich,



Les surcouches de vue d'ensemble.

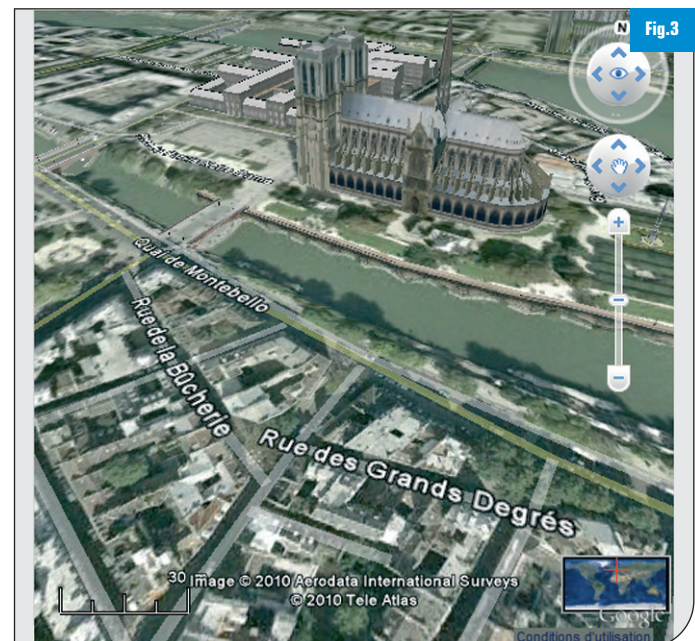
l'équateur et les parallèles du tropique du Cancer et du tropique du Capricorne. Il est encore possible d'afficher un planisphère miniature que GE appelle une carte globale ou Overview map. Tout ceci comme illustré. [Fig.2]. Il est encore possible de faire se lever le soleil. Cette opération fait également apparaître un curseur de temps, et apporte un effet jour-nuit en fonction de l'instant défini avec le curseur. Un effet qui peut se révéler excellent pour un jeu qui dépend du temps. Toujours plus fort, il est encore possible de faire apparaître une échelle, des routes et même des bâtiments en 3D, et enfin une barre d'états qui donne, entre autres, les coordonnées géographiques et l'altitude. Ces détails n'apparaîtront que lorsque vous serez près du sol. Les bâtiments n'apparaissent en 3D que s'ils sont modélisés dans GE. Sinon ils restent en photographies. L'illustration montre la cathédrale Notre-Dame de Paris ainsi que les rues environnantes. Même le nom des rues y est [Fig.3]. Pour terminer avec les layers, remarquons qu'il est possible d'obtenir des affichages historiques, c'est-à-dire l'état de la terre, des villes, etc. en fonction de ce qui est disponible dans les bases de données de Google selon certaines périodes. Lorsque l'on demande l'affichage historique, un curseur de temps apparaît qui permet de sélectionner la période, opération qui peut se faire également par programmation. Nous renvoyons le lecteur à la documentation pour cela ainsi qu'au code `layercontrol.html` disponible sur notre site.

4 LES MARQUES

Google Earth permet d'identifier des endroits précis en les pointant par une marque, qui par défaut a un look de punaise. L'illustration montre une marque posée sur Paris. Pour cela on utilise les coordonnées géographiques de la ville :

```
// Marquer Paris 48N50 0h09E
function setMark() {
    var latitude = 48.833;
    var longitude = 2.25;

    placemark = ge.createPlacemark({Paris});
```



La cathédrale Notre-Dame de Paris, modélisée dans Google Earth.

```

var point = ge.createPoint({});
point.setLatitude(latitude);
point.setLongitude(longitude);
placemark.setGeometry(point);
ge.getFeatures().appendChild(placemark);

placemark.setName('Programmez!');
placemark.setDescription('Abonnez-vous! :-)');
}

```

Les coordonnées géographiques, latitude et longitude, peuvent être obtenues sur un atlas. Et même de Google Earth, ce que nous verrons plus loin. Google Earth attend des coordonnées exprimées en degrés et décimales de degrés, alors que les atlas donnent les coordonnées en degrés et minutes. Il faut donc faire une petite conversion pour des résultats rigoureux. Il arrive aussi que les atlas donnent la latitude en degrés minutes, et la longitude en temps. Ainsi pour Paris nous pouvons avoir 48°50 Nord et 02°09 Est. Tout ce qu'il faut savoir dans ce cas est que la terre fait un tour sur elle-même en 24 heures, soit 15 degrés à l'heure. Une simple règle de trois permettra d'obtenir la bonne valeur. La description définie dans le code ci-dessus apparaît dans une bulle lorsqu'on clique sur la marque. Il est néanmoins possible de définir et faire surgir une bulle plus explicitement :

```

function showBalloon() {
    var balloon = ge.createFeatureBalloon({});
    balloon.setMaxWidth(200);
    balloon.setFeature(placemark);
    ge.setBalloon(balloon);
}

```

Et enfin de changer la représentation de la marque en fournissant une image alternative. [Fig.4].

```

function showProgrammez() {
    var icon = ge.createIcon({});

```

```

    icon.setHref('http://localhost/ge/prog.jpg');
    var style = ge.createStyle({});
    style.getIconStyle().setIcon(icon);
    placemark.setStyleSelector(style);
}

```

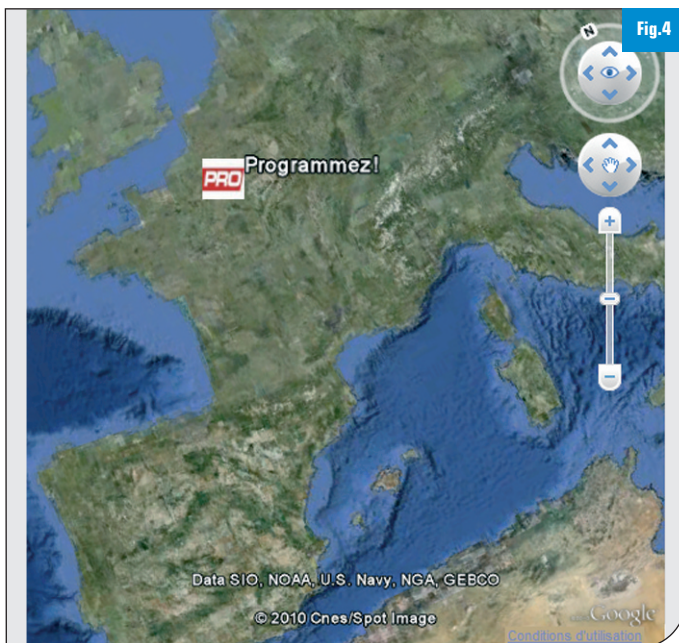
5 CAMERA ET LOOKAT

Il est maintenant temps de commencer à se déplacer. Ce qui revient, avec Google Earth à modifier le point d'observation et sa direction. Nous avons pour cela deux outils à notre disposition, la caméra et le lookAt. Une caméra définit le point d'observation. Par défaut une caméra regarde le long de l'axe des Z, «vers le bas» mais ceci peut être modifié en faisant varier son inclinaison (tilt). Par défaut cette valeur est 0.

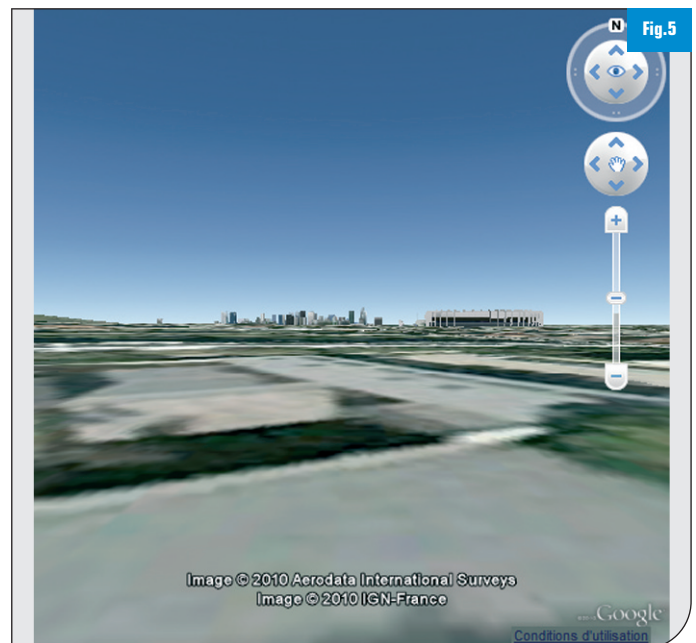
Une valeur de 90° met la caméra à l'horizontale. Le *tilt* ne doit pas être confondu avec le *roll* qui incline la caméra autour de l'axe de visée. On peut également définir l'altitude de la caméra. Une valeur négative nous emmène sous la terre ou sous le niveau de la mer le cas échéant. Avec le lookAt, ce n'est pas le point observé qui est défini mais le point d'observation. Tandis que faire varier les paramètres d'une caméra fait voir «autre chose», faire varier un lookAt fait voir la même chose, mais depuis un autre endroit. Autrement dit le lookAt se déplace.

La direction de la visée du lookAt s'appelle le heading. L'exemple caméra.html, qui est rudimentaire, requiert de votre part d'actionner les boutons dans l'ordre de haut en bas. Moyennant quoi, vous ferez une plongée sur Paris. Puis vous amèneriez la caméra à l'horizontale, puis à une hauteur de 2 m au dessus du sol. Après quoi, faites apparaître les bâtiments et vous aurez le Parc des Princes dans votre champ de vision. [Fig.5] Un peu plus loin au fond, à peine visible: Roland-Garros :-). C'est tout pour la caméra.

Bouton suivant, pour le panoramique et le zoom, un lookAt est utilisé. Attention avec le zoom. Il se répercute sur l'altitude et vous pouvez vous retrouver sous terre. N'oublions pas qu'avec le lookAt c'est le point d'observation qui est déplacé. Le code camera.html, disponible sur notre site montre comment manipuler ces concepts.



Programmez! réside à Paris.



Vue imprenable sur le Parc des Princes

6 LES ÉVÉNEMENTS

Notre avant-dernier exemple montre comment installer des gestionnaires d'événements. Avec Google Earth, un événement contient une foule de données intéressantes. Ce point n'est pas vraiment documenté, mais il est facile d'examiner l'objet événement sous un débogueur pour savoir ce qu'il a dans le ventre. Les événements sont de la programmation Javascript classique. Notre exemple `event.html` récupère les coordonnées des endroits sur lesquels on clique. Voici le code intéressant :

```
function eventHandler(event) {
    var lat = event.getLatitude();
    var long = event.getLongitude();
    camera.setLatitude(lat);
    camera.setLongitude(long);
    ge.getView().setAbstractView(camera);

    var pos = 'Latitude: ' + lat + ' Longitude: ' + long;
    document.getElementById('position').innerHTML = pos;
}
```

7 UN PETIT TOUR DANS PARIS

Nous avons récupéré ainsi les coordonnées du Parc des Princes [Fig.6], de Roland-Garros [Fig.7], de la Tour Eiffel et de l'Arc de Triomphe. [Fig.8] Notre dernier exemple vous emmènera en balade et vous fera survoler ces endroits. Avec un changement de heading dans le cas de l'Arc de Triomphe pour une meilleure observation. Les points du parcours sont définis dans un fichier KML qui est fondamentalement du XML. KML offre un très grand champ de possibilités. Il mériterait un article à lui tout seul. Nous l'utilisons ici au minimum de ses possibilités. Dans ce fichier chacun des points du parcours est décrit dans les moindres détails. Coordonnées, vitesse de déplacement, altitude, heading, zoom, etc. En voici un extrait :

```
<!-- Arc de Triomphe -->
<gx:FlyTo>
  <gx:duration>10.0</gx:duration>
  <LookAt>
    <latitude>48.872413635253906</latitude>
    <longitude>2.296112060546875</longitude>
    <altitude>100.0</altitude>
    <heading>330.0</heading>
    <tilt>80.0</tilt>
    <range>0.0</range>
    <altitudeMode>relativeToGround</altitudeMode>
  </LookAt>
</gx:FlyTo>
```

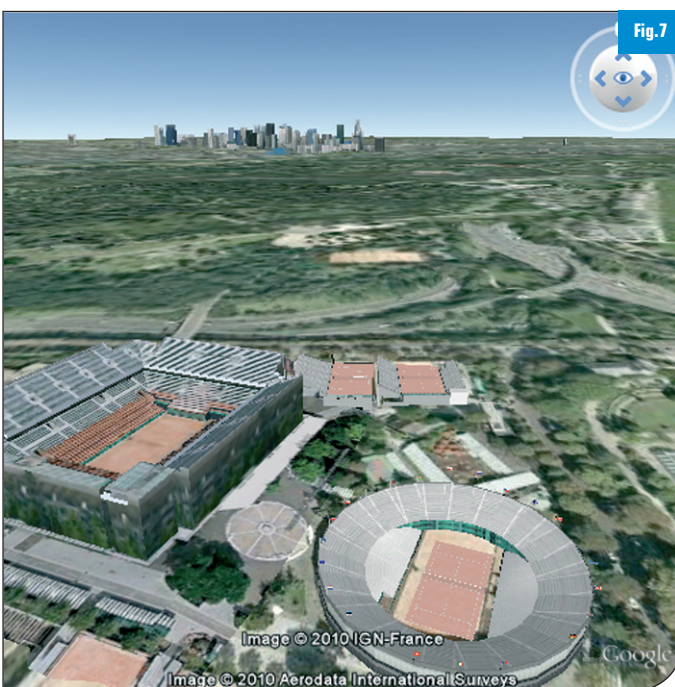
Lancer la promenade est tout simple :

```
function initTour() {
    var href = 'http://localhost/ge/paristour.kml';
    google.earth.fetchKml(ge, href, kmlFinishedLoading);
    ge.getLayerRoot().enableLayerById(ge.LAYER_BUILDINGS, true);
}

function kmlFinishedLoading(object) {
    ge.getTourPlayer().setTour(object);
}
```

Mais la mise au point d'un tel parcours peut se révéler très énervante. En effet, et sauf erreur de ma part, le plugin Windows, avec lequel j'ai travaillé, gardait systématiquement le premier fichier KML en cache, malgré des chargements explicites, et effectués avec succès, de nouvelles versions. Seul moyen contre ce comportement : tuer le processus de plugin avec le gestionnaire de tâche à chaque essai...

■ Frédéric Mazué
fmazue@programmez.com



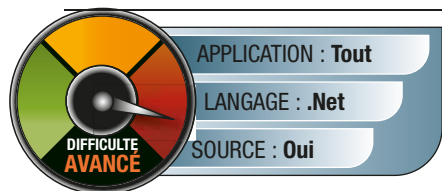
La visite continue par Roland-Garros



Et enfin l'Arc de Triomphe.

TPL Dataflow, la programmation concurrente par flux de données

Après avoir étudié la programmation asynchrone sous .NET avec Async, nous abordons aujourd'hui la programmation concurrente avec TPL Dataflow.



Le mois dernier nous avons étudié la programmation synchrone sous .NET avec Async, une extension au compilateur C# qui permet de faire de la programmation asynchrone en la pensant synchrone, donc en gommant une très grande partie de la difficulté d'écriture du code. Cette approche efficace, qui devrait faire partie de C# 5 a été élaborée avec TPL Dataflow (TDF), une nouvelle librairie .NET pour le travail avec les flux de données en environnement concurrent.

1 LE CONCEPT

Les concepteurs de cette librairie font remarquer que l'écriture du code concurrent conduit toujours aux mêmes problèmes. Le code concurrent est verbeux, compliqué, et nécessite la mise en place de verrous, de sections critiques, de sémaphores, etc. Outre les difficultés de mise au point, un tel code est par nature sujet aux deadlocks et autres races conditions, donc est tout simplement susceptible de ne pas fonctionner. Pour eux, un des problèmes de fond est qu'avec les langages et frameworks actuels, y compris .NET, le programmeur se concentre sur le flux d'exécution, comme par exemple lorsqu'il écrit des boucles parallèles, alors qu'il pourrait être plus intéressant de penser l'application en termes de flux de données (Dataflow). De même, alors qu'il est plus avantageux de travailler avec des entrées/sorties asynchrones, le programmeur aura malgré tout tendance à penser et travailler «synchrone» en raison de la complexité d'écriture du code asynchrone. Le mois dernier nous avons vu combien Async rendait facile l'écriture de code asynchrone. Ces facilités restent disponibles pour le travail avec TPL Dataflow, ce dernier apportant des types de haut niveau pour encapsuler les flux de données, tout en se basant lui-même sur les types de haut niveau de .NET comme le sont les tasks, les concurrent collections et autres tuples. Les types de TPL Dataflow travaillent de manière asynchrone et concurrente sans que le programmeur ait à se soucier de cette question. La communication entre ces types et avec eux se fait par passage de message, d'une manière qui rappelle beaucoup ce que nous avons vu dans Programmez! 120 avec Concurrency & Coordination Runtime (CCR), le framework robotique de Microsoft. De fait, TPL Dataflow est une évolution de CCR dont il reprend les concepts clés, pour un domaine d'application plus large, au-delà de la robotique. Dans l'article «CCR, une API de programmation concurrente pour .Net» de Programmez! 120, nous avons vu que CCR gommait les difficultés en transformant habilement la programmation concurrente en programmation séquentielle. Nous allons faire le même constat avec TPL Dataflow.

2 A LA BASE DEUX INTERFACES

A la base de TDF nous trouvons deux interfaces, au sens programmation objet du terme: `ISourceBlock<T>` et `ITargetBlock<T>` dont les méthodes permettent la communication de données et la connexion entre les instances des types qui les implémentent. Les implémentations de `ISourceBlock` sont des producteurs de données. Leur méthode spécifique `LinkTo` permet de les connecter à des implémentations de `ITargetBlock` qui seront des récepteurs de données. Ceux-ci ont la méthode `Post` qui permet aux producteurs de données ou au programmeur de poster des données dans les récepteurs. Cette méthode `Post` présente une très forte similitude avec la méthode `Post` des ports de CCR. Le programmeur a toute liberté d'implémenter lui-même ces interfaces pour des cas pointus. Toutefois les types intégrés, les 'Block' proposés par TDF devraient convenir dans la plupart des cas. Découvrons ces types par la pratique. Async, examiné le mois dernier, doit être installé sur votre machine.

3 CALCULER LES CARRÉS D'UNE SUITE DE NOMBRES

Pour ce premier exemple très simple, nous voulons afficher les valeurs des carrés d'un flux de nombres entiers, au fur et à mesure de leur production. Voici le code :

```
using System;
using System.Threading;
using System.Threading.Tasks.Dataflow;

namespace Carre
{
    class ProducteurEntier
    {
        private const int tempo = 1000;
        private const int nb = 11;
        ActionBlock<int> action_carre;

        public ProducteurEntier(ActionBlock<int> action_carre)
        {
            this.action_carre = action_carre;
        }

        void Worker()
        {
            for(int i=1; i<nb; i++)
            {
                Thread.Sleep(tempo);
                Console.WriteLine("Valeur: {0}", i);
                action_carre.Post(i);
            }
        }
    }
}
```

```

    }
}

public void Run()
{
    new Thread(Worker).Start();
}

class Program
{
    ActionBlock<int> action_carre;

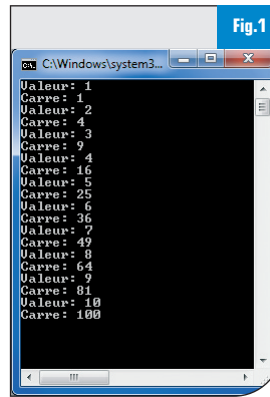
    Program()
    {
        action_carre = new ActionBlock<int>(i =>
        {
            int carre = i * i;
            Console.WriteLine("Carre: {0}", carre);
        });
    }

    void Run()
    {
        ProducteurEntier pi = new ProducteurEntier(action_carre);
        pi.Run();
    }

    static void Main(string[] args)
    {
        Program p = new Program();
        p.Run();
        Console.ReadLine();
    }
}

```

Pour notre propos, nous utilisons le type `ActionBlock`. Ce type implémente l'interface `ITargetBlock<T>`. Il s'agit donc d'un récepteur de données. Les données lui sont fournies par sa méthode `Post`. `ActionBlock` traite les données, une à la fois, dans l'ordre de leur réception. Comme le nom l'indique, une action peut être définie, qui sera appliquée à chacune des données. Dans notre exemple, nousinstancions `ActionBlock` pour un type entier et nous donnons une fonction anonyme au constructeur. Ceci n'est pas sans rappeler les `map` sur listes des langages fonctionnels. Dans notre exemple, l'instance de l'`ActionBlock` est passée au constructeur d'une classe qui produit 10 entiers consécutivement dans un thread. Le code de cette classe est tout à fait trivial. On remarque seulement que le thread poste les entiers au rythme de leur production via la méthode `Post` de l'`ActionBlock`, comme mentionné plus haut. Si cet exemple est rudimentaire, il mérite malgré tout quelques remarques. La première est que la production des nombres entiers et le calcul et l'affichage de leurs carrés résident dans deux threads distincts. Cependant le passage des valeurs de l'un à l'autre dans une file de messages (via `Post`) assure que les deux travaillent en bonne entente et qu'il n'y a pas de désynchronisation [Fig.1]. Du point de vue de la fonction anonyme passée au constructeur de l'`Action-`



Production d'entiers et affichage de leurs carrés dans deux threads synchronisés.

`Block`, les valeurs arrivent séquentiellement, tout simplement. Remarquons encore que nous avons dit plus haut que les `Blocks` pouvaient être chaînés les uns avec les autres. L'exemple ci-dessus ne montre encore rien de tel et notre `ActionBlock`, en tant que `ITargetBlock`, se comporte comme un point terminal. Rien n'empêcherait pourtant notre fonction de poster dans un autre block, même si dans ce cas, il serait sans doute mieux d'utiliser un block mixte plutôt qu'un `ActionBlock` peut effectuer des actions complexes. Ainsi nous pourrions pousser des URL au lieu de pousser des entiers dans le block, afin de télécharger une image et d'effectuer un traitement :

```

var downloader = new ActionBlock<string>(url =>
{
    byte[] imageData = Download(url);
    Process(imageData);
});

downloader.Post("http://msdn.com/concurrency");
downloader.Post("http://blogs.msdn.com/pfxteam");

```

Ceci est bien, mais si le téléchargement de certaines images est retardé, c'est tout le flux de données qui sera retardé. C'est ici que peut intervenir `Async` vu le mois dernier, afin d'effectuer tous les traitements de manière asynchrone :

```

var downloader = new ActionBlock<string>(async url =>
{
    byte [] imageData = await DownloadAsync(url);
    Process(imageData);
});

downloader.Post("http://msdn.com/concurrency ");
downloader.Post("http://blogs.msdn.com/pfxteam");

```

4 LES BLOCKS MIXTES

Nous allons reprendre et modifier l'exemple précédent. La sortie produite par ce programme sera la même, mais nous allons en compliquer un peu la «tuyauterie», pour le plaisir de jouer avec d'autres blocks. Cette fois, nous chaînerons des blocks et nous découvrirons que si un block peut être producteur ou receveur, rien n'empêche qu'il soit mixte, c'est-à-dire les deux à la fois. Avec toute la palette de blocks proposée par TPL dataflow, on retrouve le plaisir de jouer au Lego :) Le code complet de cette exemple, baptisé `CarreTransform` est disponible sur notre site. Le producteur d'entiers est le même que précédemment, à un détail près :

```

public ProducteurEntier(BufferBlock<int> buffer_block)
{
    this.buffer_block = buffer_block;
}

```

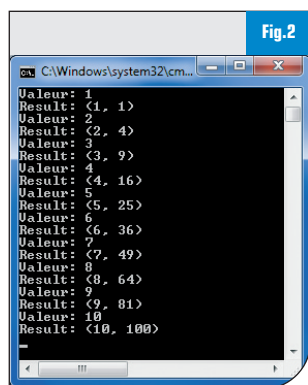
Nous employons au début de la chaîne, un `BufferBlock` au lieu d'un `ActionBlock`. `BufferBlock` a ceci de très sympathique qu'il ne fait rien, mais qu'il le fait de façon mixte car il implémente `ISourceBlock<T>` et `ITargetBlock<T>`. `BufferBlock` est un simple tuyau. On y place des données à l'entrée, on récupère les données à la sortie. Ce n'est au final qu'une simple pile FIFO (First-in First out, ou premier entré, premier sorti). `BufferBlock`, en dépit de sa simplicité, est d'un très grand intérêt car il permet de bufferiser les messages en attente de traitement que le lecteur trouvera sur notre site.

Dans ce code, le `BufferBlock` est instancié en premier, puis vient l'instanciation du type `TransformBlock` qui est mixte. Immédiatement après, nous invoquons la méthode `LinkTo` du `BufferBlock` pour lui attacher notre `TransformBlock` qui joue ici le rôle de récepteur. Le `TransformBlock` est instancié avec une fonction anonyme en argument de son constructeur. Cette fonction calcule le carré des valeurs qui arrivent dans le flux et y réinjecte le résultat avec l'instruction `return`. Enfin nous retrouvons notre vieille connaissance l'`ActionBlock`. Le `TransformBlock` se lie à lui par sa méthode `LinkTo` et tient à cet endroit le rôle d'émetteur. Avant de quitter cet exemple, remarquons bien que le `BufferBlock` y est inutile. Nous l'y avons mis pour le plaisir de jouer avec et de chaîner des `Blocks`.

5 DIVERGENCES ET CONVERGENCES DE FLUX

TDF ne limite pas le travail à un flux de données unique. Au contraire, il est possible de joindre des flux de données et même de créer des flux divergents. L'exemple `CarrJoin`, disponible sur notre site, illustre ceci. Dans cet exemple, nous commençons la construction du flux avec un `BroadcastBlock`. Celui-ci a pour propriété de dupliquer les données qu'il reçoit en autant de copies que de `Blocks` auquel il est lié. Ce qui au final va créer autant de flux. Dans notre exemple le flux initial se sépare en deux flux avec notre `BroadcastBlock`. Un flux qui s'écoule paisiblement dans un `BufferBlock`, sans rien faire d'autre,

et un flux dans lequel, à nouveau, nous calculons le carré des valeurs. Puis nous rejoignons nos deux flux en un seul en les faisant se lier à un `JoinBlock`, qui a pour caractéristique de construire un `Tuple` avec les valeurs qu'il reçoit. `Tuple` que nous affichons à la fin du flux. Nous remarquons que tout se passe bien [Fig.2]. Et avec cet exemple nous commençons à bien percevoir la puissance de TDF, dont les `blocks` sont autant de tâches indépendantes, que le programmeur n'a pas à gérer, et qui se synchronisent automatiquement.



Des tuples constitués par la jonction de flux automatiquement synchronisés.

6 THE SANTA CLAUS PROBLEM

C'est de saison, car cet article a été écrit au moment de Noël. The Santa Claus Problem, ou problème du Père Noël est un problème classique de programmation concurrente. Il s'énonce ainsi. « Le Père dort en attendant d'être réveillé par la totalité de ses neufs rennes qui reviennent de vacances, ou par trois de ses dix elfes qui viennent lui soumettre une nouvelle idée de jouet après un temps de réflexion. Si le père Noël est réveillé par les rennes, il les attelle à son traîneau et va distribuer des jouets, puis il renvoie les rennes en vacances. S'il est réveillé par des elfes, il étudie leur idée pendant

une réunion, puis renvoie les elfes au bureau d'études. Enfin s'il est sollicité par les rennes et un groupe d'elfes simultanément, il doit donner la priorité aux rennes. » Dans ce problème les neuf rennes et les dix elfes sont des entités indépendantes, a priori modélisés par autant de threads. Le père Noël est éventuellement un thread lui-même. Lorsque le Père Noël, est sollicité par les rennes et les elfes simultanément, nous avons un accès concurrent. Ce problème classique de programmation se résout bien évidemment avec n'importe quel langage doté de la prise en charge de la programmation concurrente. Ainsi, il existe des solutions en ADA et Java qui utilisent les sémaphores. On trouvera aussi, à <http://www.crsr.net/Notes/SantaClausProblem.html>, une solution en Haskell et une en Erlang. De l'humble avis de votre serviteur, la solution en Erlang est la plus élégante. Erlang est un langage orienté processus tout désigné pour ce genre de problème. Les processus communiquent par passage de messages et c'est ce qui en fait la solution la plus proche de celle avec TDF que nous allons écrire maintenant. Toutes les solutions citées sont totalement concurrentes, jusqu'au père Noël. Avec TDF, en pensant notre code par flux de données, nous allons finalement rendre séquentielles les arrivées concurrentes de données. Bien sûr on peut émettre la critique que ceci dénature le problème, qui n'est plus tout à fait un vrai problème de programmation concurrente. Mais l'autre point de vue est que cette approche, si elle sacrifie à la pureté, est d'une extrême simplicité, c'est son imbattable avantage. La partie des flux proprement dite n'a demandé qu'une quinzaine de minutes pour l'écriture, et a fonctionné dès le premier essai. Notre code, `SantaClaus`, disponible au complet sur notre site, commence par définir deux classes `Renne` et `Elfe` avec une routine de thread pour chacun, qui les fait agir à l'issue d'un délai aléatoire. Les neuf et dix instances respectivement sont activées au début du programme. Chaque thread, à l'issue du délai, poste un message soit à la porte des rennes, soit à la porte des elfes. Disons que ces portes, matérialisées par des `BatchBlock` sont les portes du bureau du secrétaire du Père Noël, et que celui-ci va mettre un peu d'ordre dans tout ce chaos. Les `BatchBlocks` ont l'excellente idée de recevoir des messages jusqu'à une quantité déterminée lors de leur instanciation. Ils libèrent alors les données rangées dans un tableau. Leur seule présence résout le problème de constituer des groupes de dix rennes ou de trois elfes. Il n'y a rien à faire d'autre. Les données entrent ainsi dans le flux. Voici l'extrait de code correspondant :

```
enum Personnel { Rennes, Elfes };

struct SantaAction
{
    public Personnel personnel;
    public int[] elfes;
};

class SantaClaus
{
    private const int nb_rennes = 9;
    private const int nb_elfes = 10;
    private const int groupe_elfes = 3;
    private Renne[] rennes;
    private Elfe[] elfes;
    private ActionBlock<int> porte_des_rennes;
    private ActionBlock<int> porte_des_elfes;
```



```

private BatchBlock<int> rassemblement_des_rennes;
private BatchBlock<int> rassemblement_des_elfes;
private ActionBlock<int[]> tous_les_rennes;
private ActionBlock<int[]> tous_les_elfes;
private BufferBlock<SantaAction> buffer_santa_reveil;
private ActionBlock<SantaAction> santa_reveil;

public SantaClaus()
{
    rennes = new Renne[nb_rennes];
    elfes = new Elfe[nb_elfes];
    buffer_santa_reveil = new BufferBlock<SantaAction>();

    //// Traitement du flux des rennes
    porte_des_rennes = new ActionBlock<int>(id =>
    {
        Console.WriteLine("Le renne: {0} frappe a la porte des rennes", id);
        rennes[id].Knocking = false;
        rassemblement_des_rennes.Post(id);
    });

    rassemblement_des_rennes = new BatchBlock<int>(nb_rennes);

    tous_les_rennes = new ActionBlock<int[]>(tableau =>
    {
        Console.WriteLine("Tous les rennes sont prêts");
        foreach (int i in tableau)
        {
            Console.Write("{0} ", i);
        }

        Console.WriteLine("");
        SantaAction sa = new SantaAction();
        sa.personnel = Personnel.Rennes;
        buffer_santa_reveil.Post(sa);
    });

    rassemblement_des_rennes.LinkTo(tous_les_rennes);
    //// Fin traitement du flux des rennes

    //// Ici le Pere Noel agit
    santa_reveil = new ActionBlock<SantaAction>(sa =>
    {
        if (sa.personnel == Personnel.Rennes)
        {
            Console.WriteLine("Le Pere Noel attelle ses rennes");
            Console.WriteLine("Puis distribue ses jouets....");
            Thread.Sleep(1000);
            Console.WriteLine(
                "Distribution terminee, le Pere Noel libere ses rennes");
            for(int i=0; i<nb_rennes; i++)
                rennes[i].Knocking = true;
        }
        if (sa.personnel == Personnel.Elfes)
        {
            Console.WriteLine("Le Pere Noel recoit un groupe d'elfes");
            Console.WriteLine("Puis il etudie leur proposition....");
            Thread.Sleep(1000);
            Console.WriteLine(

```

```

                "Etude termine, le Pere Noel libere ses elfes");
            for (int i = 0; i < groupe_elfes; i++)
            {
                elfes[sa.elfes[i]].Knocking = true;
                Console.WriteLine("{0} ", sa.elfes[i]);
            }
            Console.WriteLine("");
        }
    });
    //// Fin du travail du Pere Noel
    buffer_santa_reveil.LinkTo(santa_reveil);

    //// Traitement du flux des elfes
    porte_des_elfes = new ActionBlock<int>(id =>
    {
        Console.WriteLine("L'elfe: {0} frappe a la porte des elfes", id);
        elfes[id].Knocking = false;
        rassemblement_des_elfes.Post(id);
    });

    rassemblement_des_elfes = new BatchBlock<int>(groupe_elfes);

    tous_les_elfes = new ActionBlock<int[]>(tableau =>
    {
        Console.WriteLine("Un groupe d'elfes est pret");
        foreach (int i in tableau)
        {
            Console.Write("{0} ", i);
        }
        Console.WriteLine("");
        SantaAction sa = new SantaAction();
        sa.personnel = Personnel.Elfes;
        sa.elfes = new int[nb_elfes];
        sa.elfes[0] = tableau[0];
        sa.elfes[1] = tableau[1];
        sa.elfes[2] = tableau[2];
        buffer_santa_reveil.Post(sa);
    });

    rassemblement_des_elfes.LinkTo(tous_les_elfes);
    //// Fin traitement du flux des elfes
    }
    // etc.
}

```

L'organisation de ce code est simple. Au départ nous avons deux flux parallèles de données différentes, rennes et elfes. Dans chaque, flux, lorsqu'un BatchBlock libère un tableau de données, nous modifions celles-ci en un type de données commun à travers un ActionBlock. Puis nous postons la nouvelle donnée dans un BufferBlock. C'est cet endroit du code qui séquence les arrivées des groupes de rennes et d'elfes. A la sortie de ce BufferBlock, le Père Noël est réveillé et il agit selon les caractéristiques de la donnée, groupe de rennes ou groupe d'elfe. Et ici, le Père n'est jamais sollicité de façon concurrente, bien qu'au départ des flux, rennes et elfes s'agitent dans le plus grand désordre. Cette approche aplanit totalement la difficulté de la programmation concurrente. On remarquera que nous n'avons utilisé ni verrou ni sémaphore, ni rien qui y ressemble.

■ Frédéric Mazué - fmazue@programmez.com

MOBILITÉ

Développer pour le web mobile



Difficulté : **

Editeur : Pearson

Auteur : Maximiliano Firtman

Prix : 42 €

Le web mobile est désormais banalisé et son utilisation s'accroît avec les ventes de smartphones. Mais quand on est développeur, comment appréhender le web mobile, comment créer un site ou l'adapter ? Comment tester ? Faut-il se focaliser sur une ou deux plateformes ou au contraire être le plus généraliste possible ? Et il y a l'embarras du choix sur le système, le type de navigateur. Ce livre vous montrera comment construire une application standard et la rendre opérationnelle sur des appareils précis. Vous découvrirez également les particularités et les pièges de la mise en oeuvre des applications mobiles avec HTML, CSS et d'autres outils web standard. Il explore également les variantes des différentes plateformes, les difficultés des navigateurs mobiles, les design patterns Ajax pour le Web mobile, et bien d'autres points encore. Rapidement, vous serez capable d'utiliser les API Web 2.0 dans des applications pour l'App Store, l'App World, l'OVF Store, l'Android Market et d'autres boutiques en ligne.

WEB

Sites web avec Joomla 1.6



Difficulté : ***

Editeur : Dunod

Auteur : collectif

Prix : 22 €

Joomla est un des CMS open source les plus connus avec Drupal. Et il peut servir à tout : site web, réseau social, blog, etc. Ce livre se veut 100 % pratique pour savoir comment installer, configurer, créer ses premiers sites. Joomla nécessite une dose de maîtrise pour l'exploiter au mieux. Sa vraie puissance se cache sous le capot. En complément, le CD-Rom embarque la version 1.6, les sources et outils utilisés dans le livre, un template prêt à l'emploi. Et vous pourrez bénéficier d'un an d'hébergement gratuit (sous conditions).

LIVRE DU MOIS

Les tests logiciels

Difficulté : *** - Editeur : Lavoisier

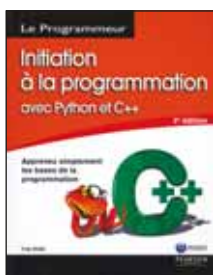
Auteur : Bernard Homès - Prix : 79 €

Les tests devraient être une préoccupation des développeurs et il faudrait les réaliser tout au long du développement. Mais encore aujourd'hui, le test est souvent la portion congrue des projets informatiques et parfois même, réduits à peu de choses. Qui le test coûte cher en temps et en argent. Bernard Homès, spécialiste reconnu du test logiciel en France, revient aux fondamentaux. L'auteur les présente clairement et particulièrement les processus organisant et structurant les tests ainsi que la dimension éthique indispensable. Puis nous rentrons dans le cycle de vie et les techniques pour définir, exécuter les batteries de tests. Car le test vit et évolue avec le projet, les versions. L'ouvrage se conclut sur deux chapitres importants : les outils et l'automatisation et les modèles de tests. Un livre indispensable à lire et à relire. Seul regret : le prix.



LANGAGE

Initiation à la programmation avec Python et C++ 2e édition



Difficulté : **

Editeur : Pearson

Auteur : Yves Bailly

Prix : 24 €

Vous souhaitez programmer ? Pourquoi ne pas s'initier à Python et au C++ ?

L'auteur aborde les notions fondamentales des langages, les spécificités : boucles, objet, variables, interface, notion de 3D, les outils, la récursivité dans la programmation, etc. L'auteur expose de très nombreux exemples de code, expliqués et très clairs. Une bonne initiation.

3D

3DVIA Studio



Difficulté : ***

Editeur : Eyrolles

Auteur : Jacques Vernus

Prix : 35 €

Outil de développement de contenu 3D interactif, 3DVIA permet de réaliser des applications temps réel (jeux, simulations, etc.) avec une qualité de rendu très élevé. Développé et mis à disposition gratuitement par Dassault Systèmes pour Windows et bientôt pour Mac OS X (sortie prévue courant 2011), 3DVIA Studio propose, dans la lignée de Virtools Dev, de créer

des "expériences 3D interactives" et de les partager sur le portail communautaire 3dvia.com. En effet, grâce à la programmation schématique, qui s'appuie sur la mise en oeuvre d'algorithmes de façon visuelle et intuitive (plutôt que sur un langage tel que le C++), il n'est plus nécessaire d'être un développeur chevronné pour créer des expériences 3D. Une bonne initiation à la 3D et à ses techniques !

3D

CSS avancées



Difficulté : ***

Editeur : Eyrolles

Auteur : Raphaël Goetter

Prix : 38 €

CSS est devenu emblématique du développement web.

Et avec les évolutions HTML 5 et CSS 3, CSS est plus que jamais incontournable ! Bien maîtrisé, CSS garantit la qualité de rendu quel que soit le navigateur, le système. Mais on n'en sait jamais assez sur CSS. Grâce à ce livre, vous irez plus loin en apprenant à faire usage tout autant des technologies avant-gardistes de CSS 3 et HTML 5 que de pratiques avancées, concrètes et mal connues déjà utilisables en production, et ce, pour l'ensemble des médias reconnus par les styles CSS (écrans de bureau ou mobiles, messageries, mais aussi impression, médias de restitution vocale, projection et télévision). Vous maîtriserez les arcanes de CSS 2.1.

ATTAQUES XSS, INJECTION SQL...

VOTRE SITE WEB EST EN DANGER !

AUDITEZ-LE MAINTENANT

WebSure Agency



Analyse automatisée, rapide et à très bas coût destinée aux Web Agency souhaitant s'assurer de livrer des sites Web sans failles.

WebSure Indeeep Audit



Analyse semi-automatisée menée par un Expert conjointement à une recherche approfondie des failles et leur position.

WebSure Audit



Analyse semi-automatisée menée par un Expert utilisant les dernières technologies de scan de sites web.

WebSure lance le tout premier service d'audit de site Internet à bas coût pour toutes les entreprises. Aucune société n'est à l'abri de voir son site et ses données volées ou détruites par des personnes mal intentionnées. Faites confiance à de réels experts pour vous prémunir de tous les risques qui pèsent sur votre réputation et vos informations cruciales.

Pour plus d'informations, visitez le site :



www.websure.fr





J'ai mon imagination pour seule limite. Je pense Windows Azure.

Windows Azure est la plateforme applicative qui vous permet de déployer et exécuter vos applications dans le cloud. Son modèle de programmation vous permet de concevoir des applications indépendantes de l'infrastructure. Ainsi, en cas de pics de demandes, notre plateforme s'adapte dynamiquement à vos besoins.

Avec Windows Azure, libérez votre imagination et restez maître du code. C'est ça la puissance du Cloud.

Pensez Windows Azure.

**Essayez la plateforme Windows Azure
gratuitement* sur
www.windowsazure.fr**

* 1 mois d'utilisation offert. Offre valable jusqu'au 13 mai 2011 minuit.