

QUELLES PLATES-FORMES pour les **MOBILES ?**

- TabletPC
- PocketPC
- SmartPhone
- Palm OS
- i-mode
- Java

EXCLUSIF

INFOPATH

Sur OFFICE 2003 :
XML pour
utilisateurs
avancés

PHP 5.0

Une refonte
complète

● **Eclipse 2.1**

● **C#**

Les boîtes
de dialogues
prédéfinies

● **ASP.NET :**

l'authentification
par Passport

● **J2EE**

avec

JBoss

● **Persistance
des données**

BELGIQUE 6,45 € - SUISSE 12 FS
LUXEMBOURG 6,45 € - CANADA 8,95 \$ CAN

M 04319-53-F: 5,95 €



IBM choisit Qtopia pour les PDA Linux

Après Sharp pour le Zaurus, IBM vient de jeter son dévolu sur la couche graphique Qt/Embedded et la suite collaborative Qtopia de Trolltech.

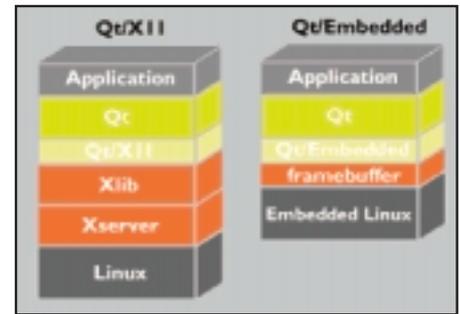
"Les entreprises cherchent à déployer les mêmes applications sur un parc hétérogène de PDA. Linux et Java sont donc deux candidats de choix" explique Michael Karasick, Responsable du groupe de travail Développements embarqués au sein d'IBM. La plateforme cible de l'éditeur s'appuie sur un microprocesseur PowerPC 405LP à faible consommation d'énergie et un système d'exploitation Linux optimisé. L'éditeur a retenu le framework de développement et le système

coût de déploiement de Qt/Embedded s'établit entre 6\$ pour 100 périphériques à 3\$ pour 10 000.

Qtopia : une alternative à Palm OS et à Microsoft PocketPC

"Nous espérons imposer Qtopia comme le système d'exploitation de référence des PDA Linux" nous déclarait récemment Eirik Eng, Président et fondateur de Trolltech. Il faut dire que l'éditeur y a tout intérêt puisque 25% de son chiffre d'affaires - 8 millions d'euros en 2002 - sont déjà réalisés dans l'embarqué. Trois arguments prèchent en la faveur de Qt/Embedded : les performances de C++, la possibilité pour un éditeur de maîtriser (personnaliser, adapter, relooker, etc.) parfaitement toutes les couches de Linux - Qt/Embedded - Qtopia, et un millier d'applications professionnelles déjà disponibles.

Qtopia est donc l'arme de Trolltech pour conquérir le marché de l'embarqué. Les fabricants désirent en effet un OS complet qui embarque avec lui toutes les applications bureautiques standards : e-mail, agenda, traitement de texte, etc. Qtopia répond à cette définition. Il embarque un lecteur MPEG3/4, un gestionnaire d'e-mail, un client web, des jeux, un agenda, un bureau entièrement personnalisable, etc. Des propriétés qui ont déjà séduit Sharp dont le SL-C700 vient de gagner le " Best In Show Award " des



PDA lors du salon Consumer Electronics Show de Las Vegas. Qtopia fonctionne sous différentes distributions de Linux embarqué, telles que le Consumer Electronics Edition (CEE) de MontaVista ou l'OpenPDA de Metrowerks. Qtopia supporte au travers de Linux embarqué une vaste variété de microprocesseurs allant de StrongARM à Xscale en passant par PowerPC, x86, MIPS ou Dragonball. Qtopia est disponible à la fois sous licence GPL et commerciale. La licence de développement coûte 195\$ pour les applications commerciales. ■

Pour plus d'informations : www.trolltech.com

David Thévenon



de fenêtrage fournis par Qt/Embedded, ainsi que l'environnement de développement d'application graphique Qtopia de Trolltech.

Standard de fait en l'absence de concurrent, Qt/Embedded propose un framework de développement d'application et un gestionnaire de fenêtres pour les applications embarquées Linux. Avantage de taille, Qt/Embedded repose sur Qt, un framework de développement C++ multi plate-forme, qui offre la portabilité de Java - écrire une fois, exécuter partout -, les performances en plus. La majorité des applications QT existantes sont donc portables sur Qt/Embedded par simple re-compilation. 150 000 développeurs ont déjà développé des milliers d'applications Qt. Le

Vous utilisez LabVIEW™ ? Pensez aux toolsets !

Modules spécialisés, complémentaires à l'environnement LabVIEW, les toolsets permettent de tirer avantage de la puissance de la programmation graphique pour répondre rapidement à des besoins précis :

- Enterprise Connectivity Toolset... pour la communication avec les bases de données, l'analyse statistique (SPC/MSP) et l'exploitation optimale d'Internet
- Report Generation Toolset... pour créer et éditer des rapports professionnels
- PID Control Toolset... pour la régulation PID et la logique floue
- Sound and Vibration Toolset... pour l'analyse acoustique et de vibrations
- Signal Processing Toolset... pour l'analyse temps-fréquence, la conception de filtres numériques et la conception par ondelettes
- System Simulation and Design Toolset... pour la simulation, la conception et l'analyse de systèmes de contrôle linéaires et non-linéaires

ni.com/frinfo

Pour davantage d'informations sur les toolsets LabVIEW, visitez ni.com/frinfo et tapez l'info-code 838px.

NATIONAL INSTRUMENTS™

01 48 14 24 24

Fax : 01 48 14 24 14 • ni.france@ni.com

© 2002 National Instruments Corporation. Tous droits réservés. Service de produits de sociétés affiliées aux technologies appliquées par leurs propriétaires respectifs.

JAVA

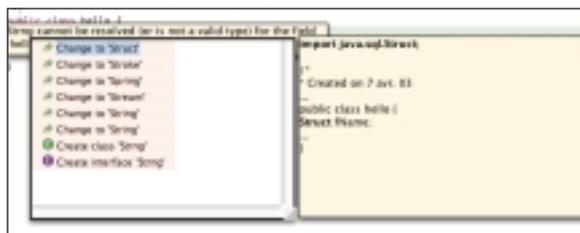
Le printemps réussit à Eclipse !

On attendait avec impatience la nouvelle version de l'IDE Eclipse. Disons-le tout de suite, la version 2.1 est un bon cru ! Cette plate-forme de développement prenant de plus en plus d'importance sur le marché des IDE, Java commence à prendre le chemin de la maturité.

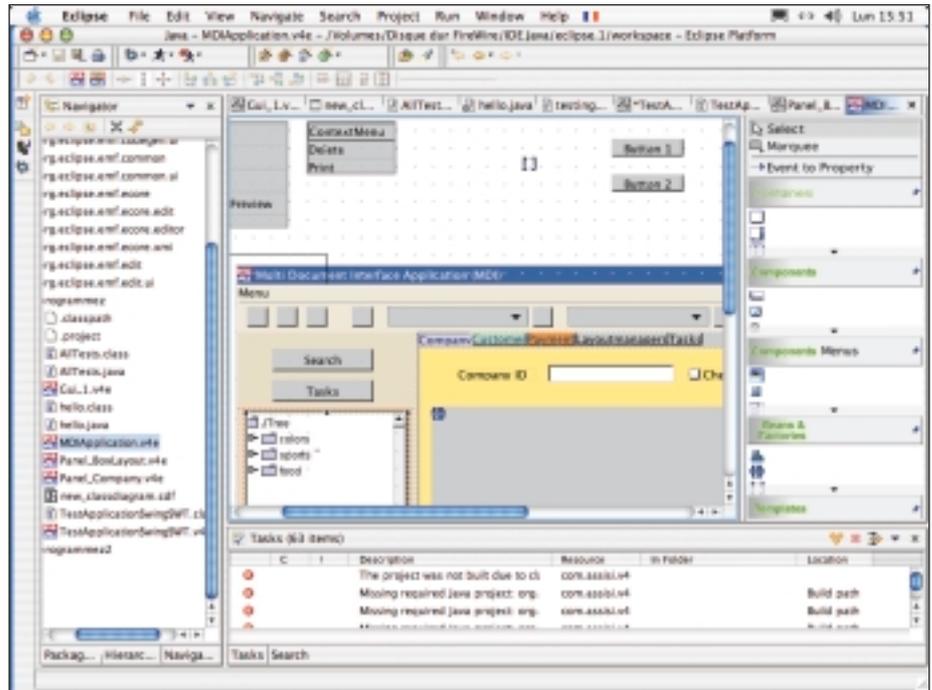
Pour certains, Eclipse reste un IDE modulaire, composé d'une plate-forme et de modules que l'on ajoute, manquant de puissance et de fonctionnalités. Pour d'autres, au contraire, Eclipse c'est l'avenir. Il existe déjà plusieurs centaines de plug-ins Eclipse. La version 2.1 apporte de très nombreuses améliorations, nouveautés et corrections par rapport à la version précédente.

Une aide au développement efficace et discrète

L'éditeur se dote de petites fonctions très pratiques. Par exemple, quand on tape une commande / fonction Java, pour en savoir plus, il suffit d'appuyer sur la touche F2 pour obtenir un petit descriptif. On dispose aussi d'une vérification à la volée du code, indiquant la présence de warning ou d'erreurs (un rectangle rouge s'active en haut à droite de la fenêtre d'édition). La fonction recherche se complète d'une option rechercher toutes les occurrences, pratique quand



le projet est très volumineux. Autre plus, la fonction Quick Fix a été améliorée. Quick Fix permet de visualiser les modifications effectuées dans un code. Sur la vérification du code et debug, on notera une meilleure intégration de Junit. On pourra aussi apprécier le nouveau Threads and Monitors. Cela facilite la visualisation et le suivi des threads de son application. Pour la navigation, on dispose maintenant dans le Workbench d'un historique de navigation dans l'éditeur. En clair, cela signifie que l'on a des boutons " précédents / suivants ". Ainsi, Quand on clique sur l'option Back, on visualise la liste du fichier ouvert et on peut afficher celui de son



choix. La partie refactoring a aussi eu droit à quelques ajouts.

Eclipse reste un IDE de référence pour coder rapidement en Java mais il manque toujours un vrai RAD intégré. Pour cela, il faut passer par des plug-ins externes qui sont parfois payants. Idem pour les fonctions UML. Si la

modularité est pratique, elle a l'inconvénient d'obliger à aller chercher soi même les modules. Nos premiers tests sur la version 2.1 sous MacOS X et la dernière JDK disponible (1.4), montrent des améliorations de performances mais pas toujours très significatives. L'organisation même de l'éditeur et du gestionnaire de projet peut étonner, comparé à un JBuilder. Par contre, la richesse fonctionnelle est impressionnante. Mais cette richesse n'est pas toujours bien exploitée par une ergonomie un peu brouillonne et lourde. Ici aussi une machine puissante, avec beaucoup de mémoire vive et un grand écran seront

nécessaires. Bref, cette nouvelle cuvée est plutôt intéressante mais on reste encore un peu sur notre faim et les développeurs débutants en Java préféreront sans doute un IDE plus complet comme un JBuilder.

En route vers Eclipse 2.2

La prochaine version majeure d'Eclipse sera estampillée 2.2. Les premiers développements ont débuté en avril 2003. La première version téléchargeable doit être disponible début mai. La version finale ne devrait pas être livrée avant début 2004. La compatibilité avec les versions 2.0 et 2.1 sera assurée. De nombreuses améliorations seront apportées aux librairies SWT et au rendu HTML. On aura droit à une meilleure gestion des plug-ins, tout comme l'apparition de fonctions asynchrones, évitant de bloquer l'utilisateur dans la mise en œuvre de certaines fonctions. L'interface utilisateur sera en partie remaniée en profondeur pour mieux prendre en compte les capacités de chaque système (ce ne sera pas un luxe). Le support de J2SE 1.5 (nom de code Tiger) est programmé. Tout va dépendre de la disponibilité de chaque environnement. Ce n'est qu'un maigre échantillon de la 2.2. Programmez ! vous tiendra au courant régulièrement des développements autour d'Eclipse. ■

François Tonic

Environnement Novell parie sur les Web Services et Java ?

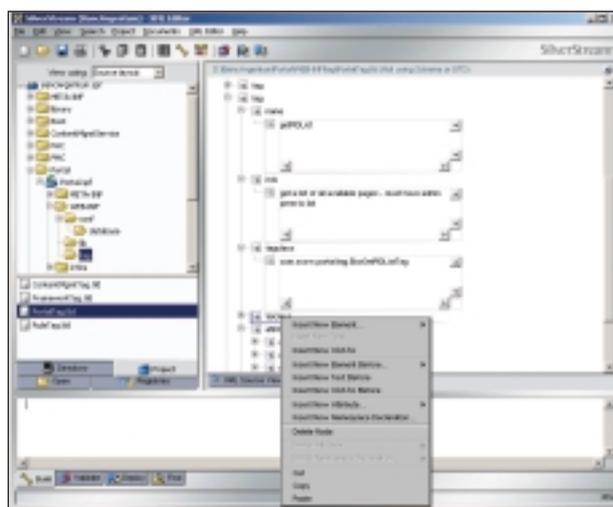
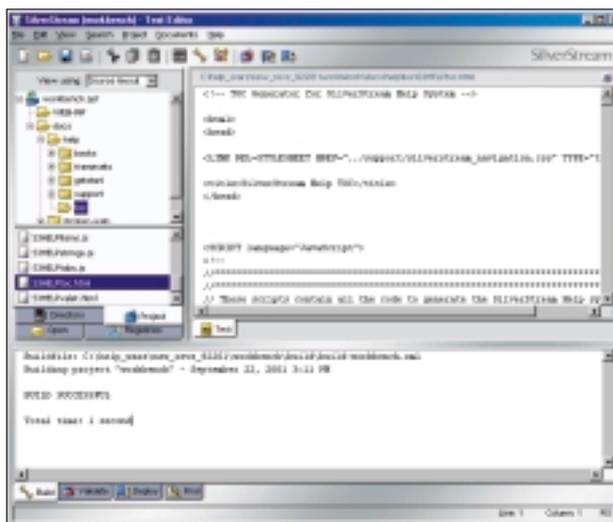
L'éditeur Novell était plus connu pour son annuaire et son système réseau Netware que pour ses outils de développement / intégration. Aujourd'hui, Novell met le paquet autour de son environnement exteNd. Les Web Services font partie de la nouvelle offre de l'éditeur, avec pour cœur de l'architecture, Java.

Novell n'est pas à l'origine de l'environnement exteNd. En réalité, ce produit avait été créé par SilverStream, éditeur racheté par Novell. Cette plate-forme se compose de cinq grands composants :

- Application Server : c'est un serveur d'applications incluant dans un même ensemble, exteNd Workbench, jBrokerWeb, jBroker MQ, jBroker ORB. Tout cela doit permettre de créer des applications Web et de les déployer rapidement en environnement J2EE, tout en supportant totalement les Web Services.
- ExtéNd Composer : outil d'intégration visuel XML.
- ExtéNd Director : il s'agit d'un environnement serveur de portails.
- ExtéNd WorkBench : environnement de développement J2EE et Web Services.
- JBroker Messaging Platform : cet outil de messaging s'appuie totalement sur les spécifications de Java Messaging Service (JMS).

Développement et intégration

Toute cette offre doit permettre 3 choses : disposer d'outils visuels, assurer une interaction et une intégration. Le tout orienté service. Le lien entre interaction et intégration se fait par des Web Services. ExtéNd est donc principalement une plate-forme de développement et de déploiement et aussi d'intégration. Mais cette plate-forme n'est qu'une composante de la nouvelle stratégie Novell. Au-dessus, on trouve Nsure pour la sécurité et encore au-dessus, on a Nterprise pour ce qui est réseau inter plates-formes. ExtéNd doit faire le lien entre les différentes solutions de l'entreprise et les données. Novell le présente un peu comme un Hub logiciel d'intégration XML, Java et Web Services. Avec les



outils de cette nouvelle plate-forme, on dispose de tout ce qu'il faut à une entreprise pour reprendre l'existant. Par exemple, avec le Composer, on peut parfaitement reprendre les données venant d'un mainframe, du messaging JMS ou encore d'XML. En fait, le cœur de la plate-forme est XML et les Web Services. Ainsi on peut transformer les fonctions / composants métiers en simples Web Services, ou en XML. Créer des applications Web ou Web Services ne suffit pas, il faut ensuite les

déployer. Comme exteNd est une architecture Java / J2EE, il est facile de l'intégrer à des serveurs d'applications de type IBM, BEA... Ca, c'est la partie intégration. Pour la partie interaction, il y a le exteNd Director. Pour être réellement multi-canal, Novell a du prendre en

compte tous les types de technologies d'affichages serveurs : JSP, servlet, portlet, .NET, client Java. Director tente de proposer dans un seul et même environnement tout le nécessaire pour créer des applications web complexes. Selon les besoins de chacun, Director se décline en trois versions : Platform Edition, standard Edition et Entreprise Edition. Bien entendu, on peut y faire aussi du Web-to-Host classique avec son architecture mainframe, de l'EAI ou encore du B2B. Au niveau annuaire, on aura les implémentations de : LDAP, UDDI et eDirectory.

L'omniprésence de Java et XML

Passons au plat de résistance, exteNd Workbench 4.0. Cet IDE permet de créer des Web Services, des EJB, des pages JSP, des documents XML, des JavaBeans, etc. L'ambition est de proposer, via une multitude d'assistants, une création rapide et assistée, avec une partie du code génération automatiquement. Pour le travail en équipe, tout est fait pour une bonne intégration en reconnaissance CLS, PVCS ou encore ClearCase. Comme nous sommes en environnement Java, le produit implémente JAX RPC 1.0. Pour le

côté middleware, il y a jBroker MQ, ORB et Web. Il supporte JMS, Corba et les services Web et s'utilise avec le serveur d'applications Novell, ou tout seul.

Bref, avec cette offre plutôt complète, Novell tente de se repositionner en profitant des développements de SilverStream. Reste maintenant à convaincre les entreprises et développeurs.

François Tonic

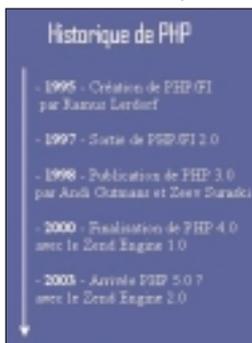
PHP 5 : une refonte complète

Dans quelques mois va sortir une nouvelle mouture du moteur PHP. Les changements depuis PHP 4 sont très importants, il ne s'agit pas d'une simple mise à jour mais d'une refonte complète du moteur.

Grâce à cette évolution majeure, ce qui était jugé par certains comme un langage de pages Web personnelles pourrait bien retrouver le crédit qu'il aurait toujours dû avoir du côté des applications professionnelles.

Historique de PHP

Le langage PHP date de 95, il s'agissait au départ d'un script Perl qui interprétait le code. PHP alors servait uniquement pour faire un système de gabarits de pages Web. La version 3 amène en 98 un vrai moteur de script tout à fait fonctionnel. En 2000 le moteur voit arriver une nouvelle version, PHP 4. Les performances sont au rendez-vous et la modularité permet l'apparition d'extensions, pour gérer tout ce qui peut l'être, de la connexion LDAP jusqu'aux interfaces GTK, en passant par la correction orthographique. Pourtant, le langage traîne encore, à tort, une réputation de script pour pages Web personnelles. Cette prochaine version du moteur devrait faire la coupure avec toutes ces critiques, en implémentant les mécanismes qui lui manquent pour une programmation orientée objet, tout en gardant la simplicité tant appréciée.



Quelles évolutions ?

Le changement le plus important est la refonte totale de la gestion des objets. Vous retrouverez maintenant une démarche plus complète, qui tient tout à fait la comparaison avec les langages orientés objet comme Java. Il ne s'agit pas d'ailleurs uniquement de syntaxe, cette évolution permettra de se connecter à d'autres composants comme COM, Java ou .NET. Nous étudierons en détail



dans la suite ce nouveau modèle objet. Les évolutions ne sont pas pour autant limitées à la POO. On trouve par exemple une gestion d'erreurs, repensée sous forme d'exceptions, des contraintes de type dans les paramètres de fonctions, ou une gestion XML unifiée.

La compatibilité du code PHP 4 est une des priorités des développeurs et l'essentiel des scripts existants devraient pouvoir être utilisés sans modification. Seules quelques rares applications nécessiteront l'utilisation d'un mode de compatibilité : les scripts PHP 4 qui utilisent la démarche objet et se reposent sur le mécanisme de copie des objets (voir plus loin pour les détails techniques).

État actuel du développement

La réflexion sur le nouveau moteur a commencé au milieu de l'année 2001, c'est à dire il y a déjà deux ans. Une série d'idées et de buts avaient été exprimés, vous pouvez encore les trouver sur <http://zend.org/engine2/ZendEngine-2.0.pdf>. Le développement du moteur interne de PHP 5, le Zend Engine 2 (le premier étant celui de PHP 4), est plus récent. Les principales fonctionnalités sont déjà intégrées, mais beaucoup de choses sont encore à implémenter, voire en discussion. Tous les choix n'ont pas encore été faits et il est impossible d'annoncer avec certitude les solutions qui seront retenues pour certains détails. Des versions de tests sont dis-

ponibles sur <http://snaps.php.net/>. Toutefois, il faut préciser que ce sont des snapshots réguliers du CVS : Elles ne sont destinées qu'à des tests et à donner un aperçu du moteur, il ne s'agit même pas de versions bêta. Il est dur d'estimer la date de sortie de PHP 5. Une date vers novembre semble réaliste mais bien entendu, donnée à titre indicatif. La date de sortie n'est pas planifiée et peut varier de plusieurs mois par rapport à cette estimation..

La modèlè objet de PHP 5

Le nouveau modèle objet est probablement le changement le plus important. Le but n'est pas de faire un Java mais vous retrouverez la plupart des repères habituels des langages orientés objets.

PHP 4 n'avait qu'une démarche objet très réduite : il était possible de définir des classes contenant des propriétés et des méthodes. Les classes permettaient l'utilisation d'un constructeur, l'héritage simple et l'appel de méthodes de manière statique. Aucun principe d'héritage multiple ou d'interface n'était disponible, ni aucun destructeur ou mécanisme de sûreté. En outre un gros reproche fait aux objets de PHP 4 est qu'ils sont par défaut passés par valeur et non par référence (il fallait utiliser l'opérateur & pour faire une référence).

Eric Daspet

eric.daspet@dreams4net.com

LES NOUVEAUTES de PHP 5

Voici une liste non exhaustive des changements prévus :

Structure générale

Les objets supportent destructeurs et constructeurs. Auparavant seuls les constructeurs existaient. Les destructeurs gagnent de plus une syntaxe plus cohérente, grâce au préfixe réservé " __ ". Les méthodes utilisées sont __construct() et __destruct().

```
class foo {
    function __construct() {
        echo 'objet créé' ;
    }
    function __destruct() {
        echo 'objet détruit' ;
    }
}
```

Support des espaces de nom. Le mot clé " namespace " peut désormais délimiter un ensemble de classes, fonctions, variables et constantes. Le mot clé " import " permet d'importer dans le cadre général, tout ou partie des structures contenues dans un espace de nom. On retrouvera donc des possibilités de gestion de modules intéressantes, sans risques de conflits de noms.

```
namespace foo {
    var $variable ;
    const A = 1 ;
    const B = 2 ;
    function fct() { ... }
    class bar { ... }
}
$result = foo::fct() ;
import class bar from foo ;
import const * from foo ;
$objet = bar( A, B) ;
```

Variables et fonctions statiques dans les classes. Les propriétés et méthodes pouvant être appelées de manière statique, sont maintenant déclarées explicitement comme telles.

```
class foo {
    static var $variable ;
    static function bar() { ... }
}
foo::$variable = 2 ;
foo::bar() ;
```

Utilisation des objets

Chargement automatique des classes instanciées. La fonction __autoload() permet de définir un chargement automatique des classes instanciées, mais non existantes.

```
function __autoload($classname) {
    $file = $classname.'.class.php' ;
    if ( file_exists($file) ) include_once($file) ;
}
$objet = new foo ; // charge le fichier foo.class.php s'il existe
```

Les objets sont passés par référence par défaut. Dans PHP 4 les objets étaient passés par valeurs et il fallait utiliser le préfixe & pour utiliser les références, ce qui pouvait se retrouver complexe à gérer. Dans PHP 5 les objets sont passés par référence par défaut.

```
$source->value = 1 ;
$ref = $source ;
$ref->value = 2 ;
echo $source->value ; // affiche 2 dans PHP5 et 1 dans PHP4
La méthode __clone() permet de copier explicitement un objet et ses valeurs.
$source->value = 1 ;
$copy = $source->__clone() ;
$ref->value = 2 ;
echo $source->value ; // affiche 1
```

Dé-référencement des objets. Il est possible d'utiliser directement un objet retourné par une fonction. Ainsi il est possible de faire fonction()->méthode() si la fonction retourne un objet.

Destruction des objets. Pour éviter que les multiples références vers un objet ne rendent difficiles la libération de la mémoire par l'utilisateur, la fonction delete() permet de détruire l'objet passé en paramètre et pas uniquement la référence en cours, comme avec unset().

Exceptions. C'est une des très grosses arrivées dans PHP 5. Les mécanismes habituels de try() et catch() pourront être utilisés. Les Exceptions pourront être des objets dérivant de la classe " Exception ". Il s'agit de l'avènement d'une gestion d'erreur complète dans le langage.

```
class myexcep extends Exception { ... }
try {
    if ($a < 1) throw new myexcep( 'A est
```

```
inférieur à 1') ;
} catch( myexcep $e ) {
    echo $e->getMessage() ;
}
```

Sûreté de fonctionnement

Sûreté de fonctionnement, via des contrôles d'accès. Les méthodes et propriétés implémentent les contrôles d'accès " private ", " protected " et " public " permettant de limiter la réutilisation de fonctions internes.

```
class foo {
    private $variable ;
    protected function bar() { ... }
}
```

Classes abstraites et interfaces. Le nouveau moteur permet l'utilisation de classes abstraites et interfaces afin de garantir la cohérence des objets.

```
abstract class ab_foo {
    abstract function ab_bar() ;
}
interface in_foo {
    function in_bar() ;
}
class foo extends ab_foo implements in_foo {
    function ab_bar() { ... } ;
    function in_bar() { ... } ;
}
```

Contraintes de type. Il est possible d'imposer aux paramètres d'une fonction d'appartenir à une classe spécifique.

```
function (maclasse $param) { ... }
```

Intégration

Intégration des classes utilisateur dans les structures natives. En implémentant dans vos classes les interfaces de l'espace SPL, il est possible de permettre aux fonctions natives PHP d'utiliser vos objets. Par exemple l'implémentation de l'interface SPL::iterator permet d'utiliser de traverser vos objets avec foreach(), reset() et each().

Interception des appels aux méthodes et propriétés. Les appels aux propriétés ou fonctions non existantes de vos classes peuvent être interceptées, si vous définissez les méthodes __get() (lecture d'une propriété), __set() (affectation d'une propriété) et __call() (appel d'une méthode).

E. D.

France 3 : STREAMING ET PHP



A ce jour, les émissions de France 3 sont archivées principalement sur des cassettes vidéo. Lorsqu'un journaliste veut consulter un sujet susceptible de l'intéresser, il doit commander une copie de la cassette qui lui parviendra au bout de quelques jours. Compte tenu de l'implantation régionale des rédactions, les sujets potentiellement importants peuvent être nombreux.

L'application Browsing utilisée par France3.

Il pourrait également visualiser des sujets en les faisant transiter sur le réseau haut débit, conçu pour échanger les vidéos entre régions, en 8Mb/s, mais cela encombrerait le réseau.

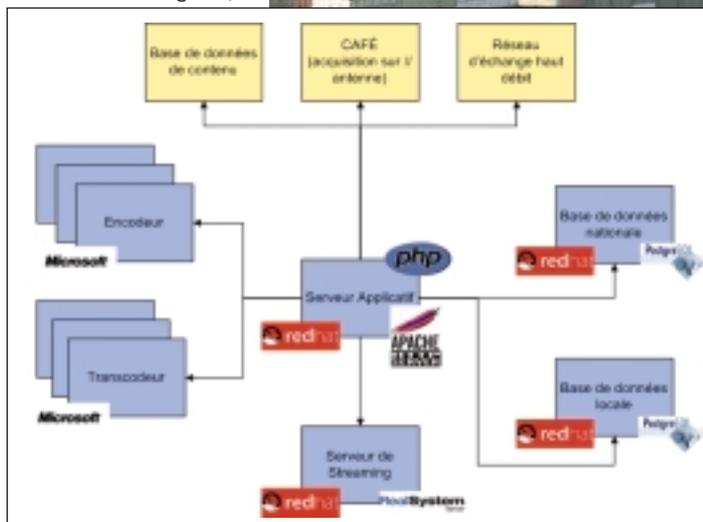
L'un des objectifs principaux du nouveau système était de permettre aux journalistes de visualiser sur l'Intranet les vidéos diffusées récemment, provenant de n'importe quelle antenne régionale, tout en consultant la base de donnée attachée.

Aujourd'hui, Browsing tourne sur Paris et Limoges. Une fois le système abouti, il sera généralisé sur toutes les antennes régionales.

Les contraintes du projet ?

Le système demandé devait être fortement modulaire pour permettre d'ajouter des

France Télévision © France 2 - Xavier Gassmann



briques de services, il devait pouvoir communiquer avec les différents Systèmes d'Information de France3. Il devait également piloter des encodeurs, des transcodeurs, permettre l'utilisation d'un serveur de streaming, gérer une base nationale, permettre un couplage sur les nombreuses antennes de France3,.... Sur le Browsing, un titre et un descriptif sont attachés à chaque sujet, constituant une base de données.

Les technologies utilisées

Le cœur du système est le serveur Applicatif. D'un côté il est à l'écoute des SI externes (en jaune), de l'autre il commande les différentes machines du Browsing (en bleu).

Parmi les systèmes inclus dans le Browsing :

- les encodeurs qui permettent de transformer des sujets vidéos passant à la télé (parallèle antenne) en des sujets realMedia,
- les transcodeurs qui transforment des sujets haut débit (MPEG2TS - 8mb/s) en RealMedia,
- le serveur de streaming qui diffuse les vidéos
- les bases de données.

Le serveur applicatif qui devait être d'une stabilité à toute épreuve est basé sur un serveur Linux couplé avec Apache et PHP.

Les serveurs de base de données et de streaming sont aussi sur une architecture Linux.

Les encodeurs et transcodeurs ont été réalisés sur une plate-forme Windows.



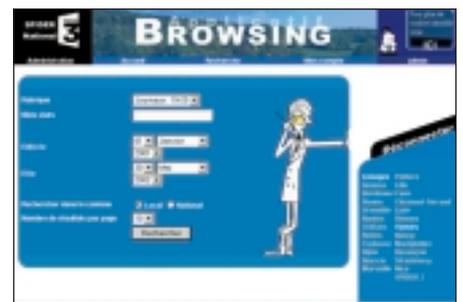
PHP afin de relier les différents SI

Consultant chez Osiacom, Cyril PIERRE de GEYER est également membre fondateur de l'Association Française des Utilisateurs de PHP. Il souligne la maturité de cette plate-forme, pour l'informatique professionnelle.



Quelles applications "pointues" de PHP avez-vous utilisées ?

Le code du Browsing a été fortement axé sur de l'Objet, afin de proposer la structure modulaire demandée. Nous générons des flux XML qui permettent de piloter les transcodeurs/encodeurs. La gestion des sessions et l'interaction en PHP et Flash ont aussi constitué des points assez poussés.



Pourquoi ces choix technologiques ?

Nous avons voulu une application fiable, qui soit capable de vivre d'elle-même, sans avoir besoin de la relancer régulièrement. C'est pour cette raison que nous avons choisi Linux. Nos serveurs sous Linux n'ont pas eu besoin d'être relancés depuis un an. L'argument de la gratuité des licences était aussi un "plus" mais pas primordial. Les principaux aspects envisagés étaient la sécurité et la stabilité.

En tant que membre fondateur de l'Afup et fortement impliqué dans le PHP, n'étiez-vous pas de parti pris dans le choix de ces technologies ?

Un parti pris non, une préférence peut être. A chaque problématique sa solution, dans ce cas le PHP s'imposait. Vous savez PHP est un langage dont les bases sont faciles à apprendre et par conséquent, beaucoup de monde "bidouille" en PHP. Il ne faut pas en

La force de PHP est dans ses bibliothèques aussi, et nous avons utilisé pleinement les fonctionnalités de la bibliothèque PEAR pour gérer des templates (dissociation du contenant et du contenu).

La plus grande difficulté n'a pas été de trouver des modules existants mais de les choisir car l'offre de qualité est conséquente.

Combien d'intervenants sur ce projet ?

L'ensemble du projet a été coordonné par Denis Vergnaud de France3.

Actuellement l'outil est disponible dans une version 2 et nous travaillons à l'associer à encore plus

de Système d'Information pour enrichir les informations textuelles.

Pour raisonner en jour homme je dirais, sans avoir les chiffres exacts, que l'ensemble a représenté par moins de 600 jours/homme, et 6 mois pleins, jusqu'ici chez Osiacom.



Stéphane Pons

déduire pour autant que c'est limité à une utilisation pour "pages perso". Les 2700 fonctions disponibles sur PHP lui permettent de générer du PDF, se connecter à quasiment toutes les bases de données, générer du XML (voir www.afup.org)

Les Outils

- Le serveur Web: il tourne sous Apache. "Tout simplement parce que c'est le leader du marché, loin devant IIS de Microsoft", indique C. Pierre de Geyer
- Les langages et outils utilisés sont nombreux : C, Bash, PHP, CronTable (lancement à date et heures prédéfinies), Samba (pour le partage des fichiers), ProFTP,...
- Toute l'intelligence du système est basée sur PHP
- Zend Accelerator pour optimiser les performances.
- SGBD : des bases PostgreSQL, pour leur puissance et leur stabilité.
- Le serveur de Streaming est un Real Server.
- Les transcodeurs et encodeurs tournent sous Microsoft Windows.

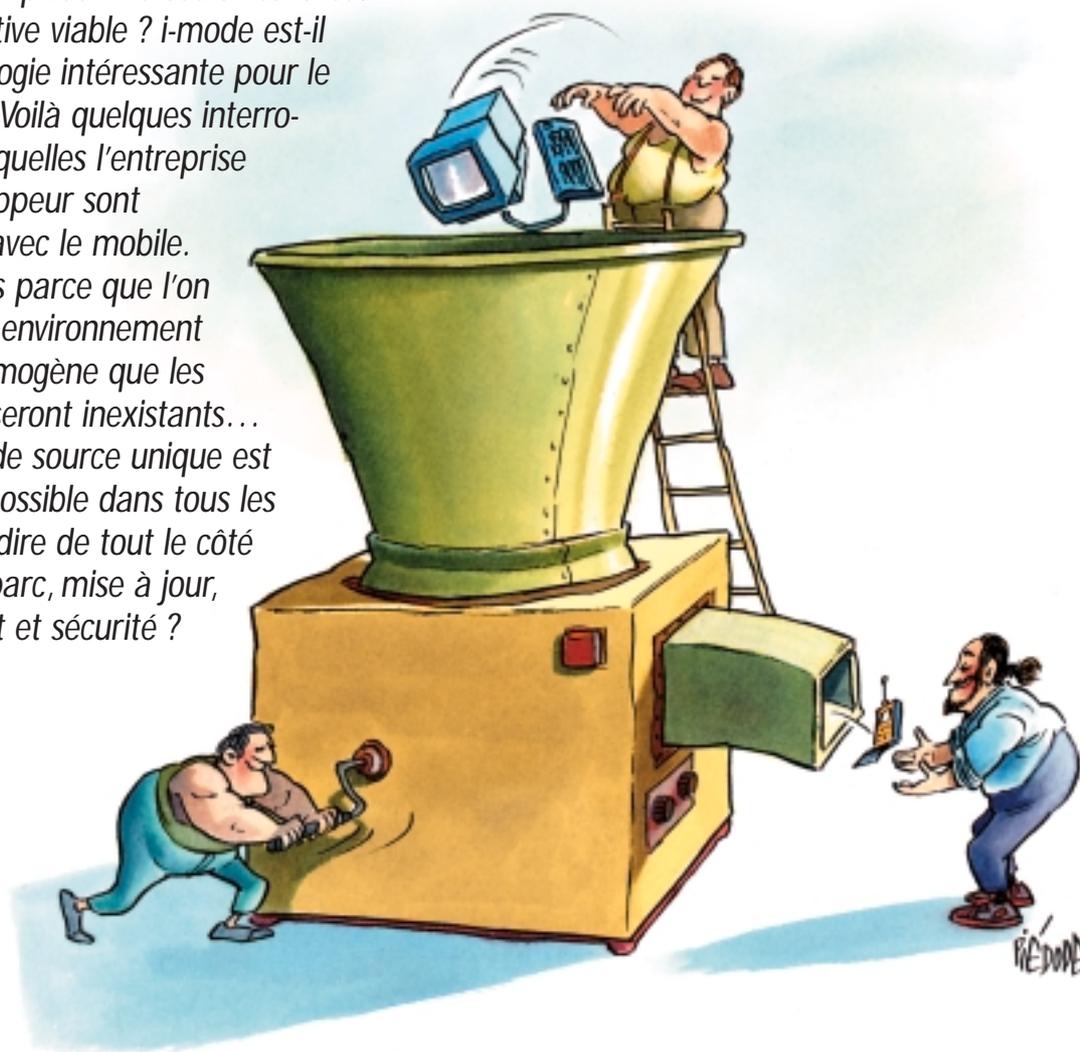
Quelles plates-formes pour les mobiles ?

PAR FRANÇOIS TONIC

Aujourd'hui, les collaborateurs peuvent être équipés de terminaux mobiles différents : PDA, Smartphone, voire de TabletPC. Être mobile, cela signifie aussi avoir un environnement applicatif adapté aux différentes plates-formes mobiles. Quelle plate-forme choisir ? Quel matériel choisir ? De quelle manière adapter une application et les accès aux données ? Comment définir une politique de sécurité ? Dans le cas d'une problématique eBusiness la question se pose aussi. Vise-t-on uniquement le téléphone ou aussi le PDA ? Et dans ce cas, quelles technologies dois-je prendre en compte.

Si vous avez des applications Windows, est-il intéressant d'utiliser une plate-forme mobile estampillée Microsoft ? Java est-il une alternative viable ? i-mode est-il une technologie intéressante pour le téléphone ? Voilà quelques interrogations auxquelles l'entreprise et le développeur sont confrontés avec le mobile.

Ce n'est pas parce que l'on possède un environnement système homogène que les problèmes seront inexistantes... Avoir un code source unique est loin d'être possible dans tous les cas. Et que dire de tout le côté gestion de parc, mise à jour, déploiement et sécurité ?



1 • NOUVEAUX MARCHÉS ET TERMINAUX : MULTIPLICATION DES OFFRES ET CASSE-TÊTE POUR LES RESPONSABLES ?

En quelques mois, nous avons vu apparaître plusieurs générations de terminaux mobiles, de technologies et de nouveaux marchés. Cette situation offre davantage de possibilités pour le eBusiness, mais risque à terme de complexifier encore un peu plus l'architecture matérielle et logicielle de l'entreprise et le travail du développeur. Essayons de faire le point.

La mobilité vise les salariés, les commerciaux, les clients ou encore les prestataires ou fournisseurs. Selon le type d'application, il y aura donc plus ou moins de cibles à prendre en compte.

Une offre très diversifiée

Dans le marché de l'informatique mobile (au sens large du terme), il existe plusieurs types de matériels, de systèmes et de langages.

- **PDA** : les PDA sont les offres les plus classiques de l'informatique mobile depuis bientôt dix ans. Il faut distinguer le PDA standard et le PDA hybride, dit PDA communicant. Il existe trois grandes plates-formes : les PocketPC fonctionnant avec Windows CE, les Palm et compatibles fonctionnant sous PalmOS et enfin les PDA Linux utilisant un système Linux dédié (ex. : Yopy).

- **Téléphone** : actuellement il existe trois grands groupes de smartphone : les téléphones utilisant Windows, SymbianOS, ou encore PalmOS. Linux est en retard mais commence à prendre pied sur ce nouveau marché. SymbianOS est le leader du domaine, même si Microsoft fait le forcing.

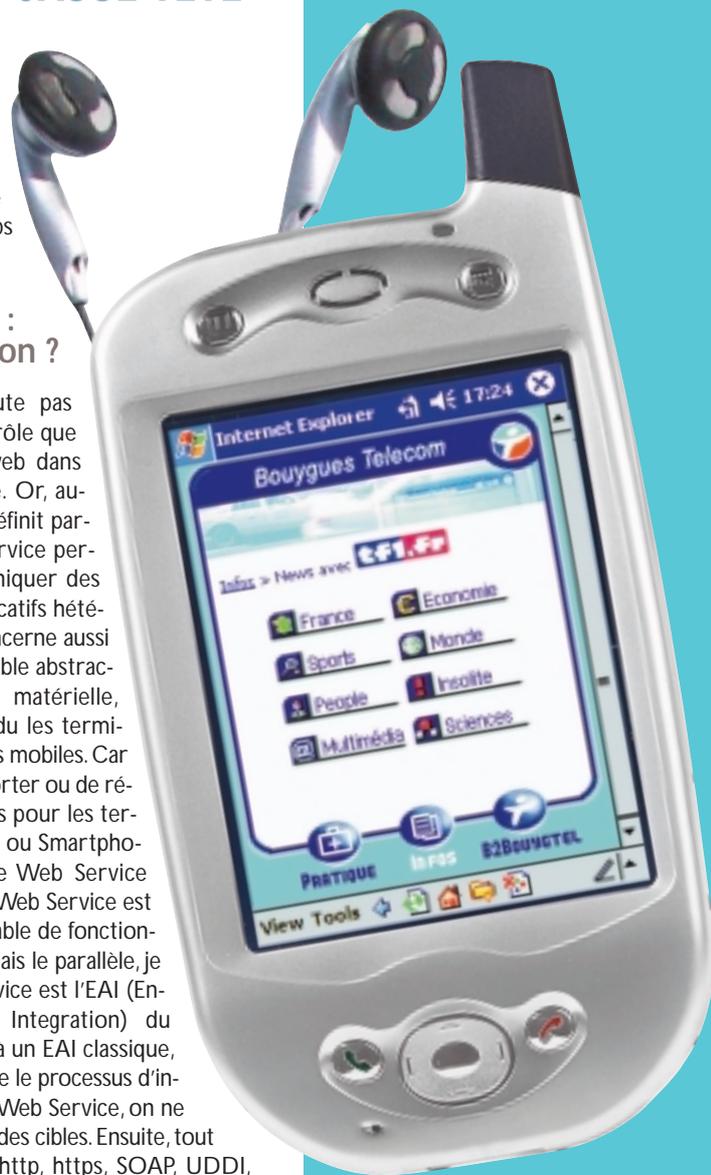
- **Portable nouvelle génération** : les ordinateurs portables de type Tablettes PC sont les petits derniers du marché. Ils offrent plus de fonctions que les portables classiques. Pour le moment, seuls des modèles Windows sont disponibles sur le marché.

Comme partiellement indiqué ci-dessus, la diversité de l'offre matériel se double d'une diversité système, technologique et aussi applicative. Cela ne fa-

cilite donc pas les choix. Et trouver la plate-forme adéquate peut prendre un temps certain.

Web Services : l'ultime solution ?

On n'avait sans doute pas imaginé au départ le rôle que jouerait le service web dans l'informatique mobile. Or, aujourd'hui ce rôle se définit parfaitement. Le Web Service permet de faire communiquer des environnements applicatifs hétérogènes. Mais cela concerne aussi le matériel. Cette double abstraction, logicielle et matérielle, concerne bien entendu les terminaux et les applications mobiles. Car au lieu d'adapter, de porter ou de ré-écrire des applications pour les terminaux mobiles (PDA ou Smartphone en tout genre), le Web Service facilite le dialogue. Le Web Service est portable et donc, capable de fonctionner partout. Si je risquais le parallèle, je dirais que le Web Service est l'EAI (Enterprise Application Integration) du futur. Contrairement à un EAI classique, le Web Service simplifie le processus d'intégration. Car, avec le Web Service, on ne se soucie pas de la ou des cibles. Ensuite, tout cela transite sur du http, https, SOAP, UDDI, WSDL, XML, etc. Bref, au lieu de vouloir à tout prix adapter le code d'une application sédentaire à du mobile ou de concevoir des clients applicatifs, le Web Service remplace toutes ces notions avec une



**MINI-BAROMÈTRE
DES TECHNOLOGIES
/ PRODUITS**

- *En forme*
Web Services
J2ME
i-Mode
TabletPC
SymbianOS
Bluetooth
.NET Compact Framework
802.11b / g
XML
IPv6
- *Les mitigés*
Linux
PalmOS
UMTS
EAI
Intel Pentium M (Centrino)
Smartphone Windows
- *En baisse*
WAP
WEP
802.11a

approche homogène partout et quels que soient les terminaux et systèmes. Or, le fait que le Web Service puisse être l'EAI du futur fait frémir un certain nombre de sociétés. Car finalement, mettre en œuvre un EAI global génère un budget très intéressant pour les éditeurs. Réaliser des Web Services nécessite moins de compétences et surtout de budget ! De toute manière l'EAI n'a jamais été une solution souple et de simplification telle qu'on pouvait l'entendre de la bouche de responsables. Mais gardons nous tout de même d'aller trop vite. Le Web Service ne peut encore s'apparenter à un EAI mais il en prend tout de même la route. Cependant, si vous devez utiliser des Web Services pour intégrer le mobile, il faut créer des Web Services "neutres" et éviter toute technologie propriétaire. Évitez tout ce qui est spécification .NET, J2EE, etc. En faisant cela, vous disposerez des composants "universels". Toute une phase de test sera bien entendu nécessaire. Le Web Service ne saura, par contre, remplacer une application entière. Si vous devez impérativement disposer d'applications mobiles, le Web Service ne sera sans doute pas la meilleure solution. Donc là, l'adaptation ou la réécriture de l'application sera nécessaire. Défaut du Web Service : avoir de préférence une connexion au réseau sans fil ou téléphonique permanente... Si ce n'est pas le cas, le Web Service n'aura que peu d'intérêt, sauf s'il est épaulé par un client mobile travaillant hors connexion et dans ce cas le Web Service peut servir d'interface, voire d'outil de synchronisation.

Et la sécurité ?

Tout ce qui est connexion distante, réseau sans fil, pose bien entendu le problème de la sécurité. Les réseaux de type Wifi (mais aussi les connexions Bluetooth) constituent une grosse faille de sécurité, malgré les progrès réalisés depuis quelque temps. Il y a aussi le problème de compatibilité entre les matériels provenant de différents constructeurs. Les incompatibilités entre les normes 802.11 a, b ou g, posent aussi problème. Si 802.11g peut utiliser le matériel estampillé 802.11b, le 802.11a est incompatible. Même si les mesures de sécurité doivent ralentir une application ou la vitesse de synchronisation, elles sont indispensables. Dans le côté classique, on peut utiliser du SSL, clé de cryptage 128 bit, https, authentification forte, signature électronique, etc.



On pourra aussi mettre en œuvre des architectures de type VPN appliquées aux mobiles (= mVPN, m pour mobiles). Le WEP ne garantit qu'une sécurité primaire dans une configuration Wifi. Si une politique de sécurité est mise en place dès l'adoption du projet mobile, cela minimise les problèmes de sécurité. Vous devez ABSOLUMENT prendre en compte dès le départ la sécurité et prévoir toutes les techniques et technologies adéquates. Cela signifie aussi avoir une application mobile propre, qui ne risque pas d'interférer avec les mesures de sécurité. Toutes les données sensibles stockées sur le mobile doivent être impérativement cryptées. L'accès au mobile ou à l'application se connectant directement au réseau interne doit être sécurisé, pour une connexion sûre et une authentification forte !

Des offres d'environnement de développement très larges

Tous les principaux éditeurs proposent des extensions mobiles à leurs produits. Cette démarche permet de faciliter le portage des applications et des données vers les terminaux mobiles. Microsoft propose une panoplie complète d'outils et son SQL Server possède une déclinaison pour Windows CE. Sybase fait de même. Borland propose un module dédié au développement des applications mobiles via un RAD, JBuilder MobileSet, tout comme Sun. La sortie, il y a peu, du nouveau processeur Intel Centrino dédié aux portables, a déjà permis à IONA de sortir le Mobile Orchestrator. Il doit permettre l'amélioration du travail des utilisateurs mobiles, grâce à un environnement utilisant du Web Service.

Il faut une attitude active !

Quand on introduit l'information mobile et les applications mobiles dans l'entreprise, il faut avoir une démarche active et non passive. Il est important de suivre tout le processus du " projet mobile ". Si vous avez un existant applicatif et qu'il doit pouvoir être utilisé sur les terminaux mobiles, prenez une plate-forme compatible ou aisément intégrable. Il faut aussi penser les méthodes de synchronisation et de mise à jour / mise à niveaux, des applications et du système mobile. Si déployer des mobiles ne pose pas de difficulté, la gestion du parc peut se révéler plus délicate. Car tous les processus de télé-distribution posent problème. Il ne faut pas modifier la sécurité du réseau d'entreprise, prendre en compte la bande passante quand on utilise les connexions sans fil, éviter le redémarrage du terminal après chaque opération, etc. Certains éditeurs (Xcellent, Marimba ou encore CA) proposent de tels environnements d'administration. Si le parc mobile est important, ce type d'outils semble incontournable. ■

FT

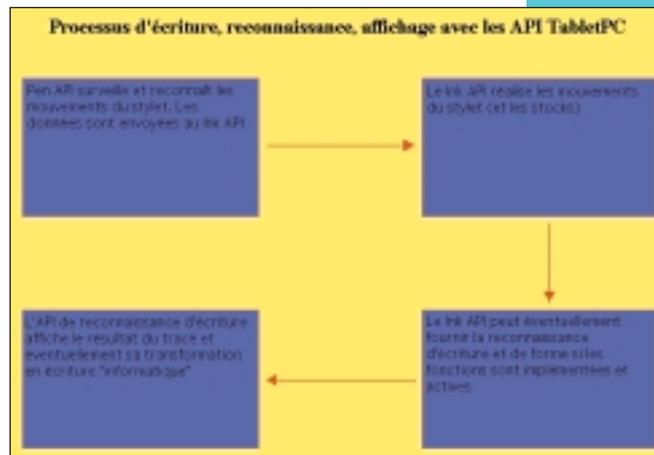
D
O
S
S
I
E
R

2 • LES NOUVELLES APPROCHES MICROSOFT POUR LE MOBILE

TabletPC : repenser l'application

Les TabletPC fonctionnent avec un Windows XP légèrement modifié pour pouvoir exploiter les fonctions spécifiques à ce type de terminaux. Microsoft propose un TabletPC Platform SDK. Ce kit permet d'implémenter dans son projet les fonctions propres aux TabletPC. La plus visible est, bien entendu, la reconnaissance d'écriture (Ink). Une API Ink permet de prendre en considération cela. La prise en compte des fonctions Ink n'est pas obligatoire, même dans un contexte TabletPC. Ink s'appuie sur la reconnaissance des mouvements du stylet et non sur les formes dessinées. Donc, en implémentant Ink, on doit gérer les mouvements. Le SDK se découpe en deux grandes sections : InkControl, classes .NET / API Win32. Les InkControl permettent une gestion rapide des fonctions Ink. On a deux types : InkEdit pour la reconnaissance texte et InkPicture pour les images. Dans InkPicture, on peut ajouter la gestion de couches. Au-delà des fonctions Ink, il faut aussi prendre en compte la gestion du stylet (ou Pen). On peut parfaitement dans une application, ajouter la gestion du Pen et du Ink, en détectant la présence du module Ink dans le système client. Une application existante peut parfaitement être adaptée pour prendre en compte le SDK TabletPC.

Le TabletPC oblige à une profonde réflexion du développeur par rapport aux portages des applications vers TabletPC. Ce n'est pas parce que l'on peut utiliser les notions d'encre et d'écriture que l'on doit transformer toute l'interface. Il faut au contraire repenser l'interface et trouver un équilibre entre le stylet et l'encre et l'interface classique. On doit se demander : où l'encre est-elle indispensable ? Et cela n'est pas forcément la tâche la plus évidente. De plus, n'oubliez pas l'utilisateur. Il faut qu'une zone InkEdit soit parfaitement identifiable ! Mélanger deux types d'interfaces, pose nécessairement un problème de différenciation. Une chose aussi à prendre en compte. Quand on adapte une application existante, il y a également le problème des gestions " propriétaires ". TabletPC ne gère pas de la même manière certains événements, comme le double-clic. Le développeur devra donc bien prendre en compte les différences de gestions événementielles. D'autre part, vous savez sans doute que l'on peut utiliser l'écran du TabletPC selon deux orientations. Donc, il ne faut pas oublier de prendre en compte cela et de tester son interface dans les deux orientations... De plus, si le terminal TabletPC n'a pas de clavier, le système possède des mouvements de stylet prédéfinis,



permettant d'accéder à des applications ou à des fonctions. Donc, il ne faut pas que des " raccourcis-stylets " de son application heurtent les raccourcis déjà réservés.

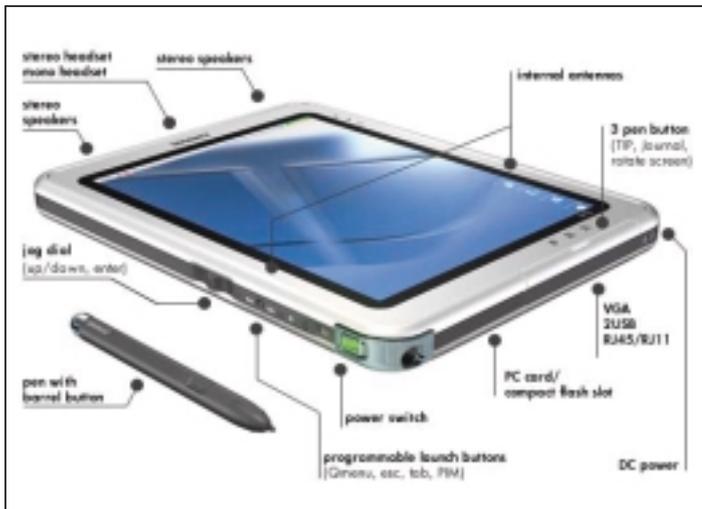
InkEdit est un dérivé du RichTextBox, donc une zone de saisie. Cette " box " permet d'utiliser l'encre et de mettre en œuvre le moteur de reconnaissance d'écriture. Notons que la reconnaissance peut être, soit automatique (via la propriété `RecognitionTimeout` qui permet aussi de paramétrer le délai), soit manuelle. Mais il peut aussi être utilisé par une saisie au clavier. InkPicture permet de " capturer " l'encre sans reconnaissance d'écriture. Cette zone peut aussi servir à afficher des images. L'utilisation de ces deux objets est standard pour la déclaration et construction, tout comme pour l'initialisation

```
Exemple d'une déclaration et construction :
private Microsoft.Ink.InkEdit inkEdit1;
private Microsoft.Ink.InkPicture inkPicture1;

this.inkEdit1 = new Microsoft.Ink.InkEdit();
this.inkPicture1 = new Microsoft.Ink.InkPicture();
```

Fondamentalement, il n'y a donc peu de différences entre la programmation et appels des API Windows et le TabletPC. Pour ce qui concerne la prise en compte et la gestion de l'encre, on passe par les classes du Managed API (classe `Tablet`, `Cursor`, `CursorButton`, `InkOverlay`, `InkCollector`, etc.). Finalement, selon le type d'utilisation de l'application, on implémente ou non la gestion de l'encre numérique. Dans ce cas, on met en place uniquement le `InkControl`. Cette solution permet de créer facilement et sans contrainte, une application générique pour Windows et TabletPC. Pour tester une application contenant du code TabletPC, on peut utiliser un desktop (équipé du SDK). Par contre, certaines fonctions seront absentes.

La plate-forme TabletPC propose aussi d'autres



SDK et technologies. Outre le Ink-to-Text Conversion (moteur de reconnaissance d'écriture), on peut accéder aux fonctions vocales avec le Speech Recognition (ou SAPI). Bien entendu, le support multi langage n'a pas été oublié. Actuellement six langues sont supportées, d'autres suivront. MS Office n'a pas été oublié. Un pack spécial (Microsoft Office XP Pack for Tablet PC) permet aux applications Offices d'utiliser les fonctions du TabletPC. Et .NET ? Le TabletPC intègre la version standard du framework.

.NET Compact Framework : un framework sous contrainte

Comme vous le savez déjà, le .NET Compact Framework est un sous-ensemble du .NET Framework normal mais dédié aux terminaux mobiles de types PDA, smartphone. Le framework a été largement dégraissé pour faire environ 1 Mo. Contrairement au TabletPC, il n'est pas possible de faire

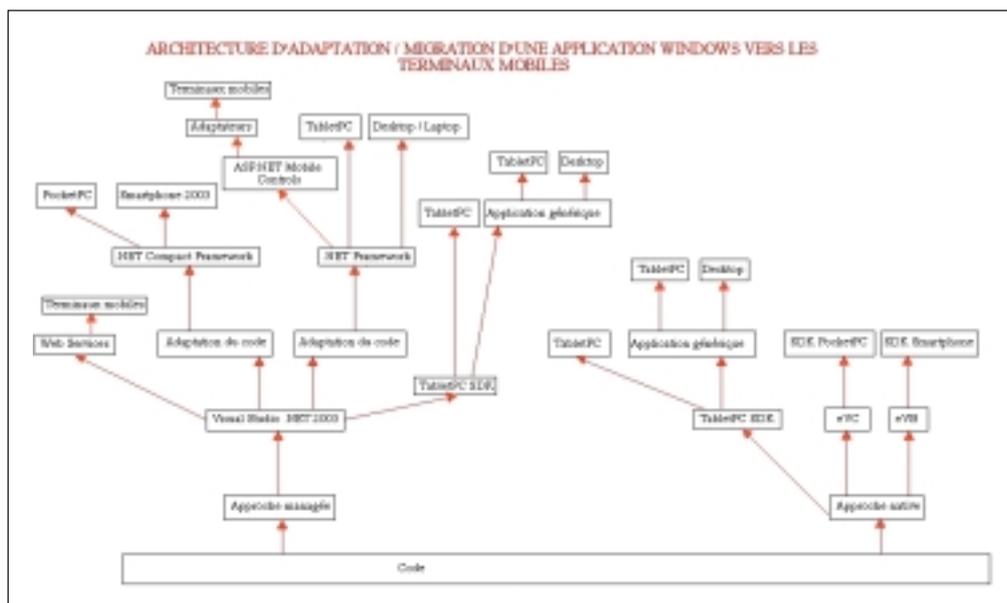
une application générique contenant le code .NET normal et le code pour mobiles. L'application serait bien trop lourde. On sera donc obligé de faire deux exécutables différents avec 2 codes différents. De plus, il faudra apprendre les spécifications du Compact Framework pour pouvoir épurer le code source. Pour le moment, il n'y a pas d'outils automatiques. Le Compact Framework poursuit deux objectifs : proposer des codes managés sur des terminaux mobiles et des Web Services. Le module Compact Framework est en standard dans Visual Studio .NET. Attention, pour pouvoir utiliser le Compact Framework sans un Smartphone, il faudra attendre la sortie de Smartphone 2003 !

.NET Compact Frameworks s'est vu couper un certain nombre de classes et de fonctions. Dans le sous-ensemble Drawing, il manque l'imaging et le printing. Dans XML, il n'y a pas de Xslt/Xpath et sérialisation. Dans le système, le côté runtime a été enlevé ainsi que le ServiceProcess et configuration. Dans la partie Data, il manque le Design et le System.Web contient uniquement les Services et la sécurité. ASP.NET a été totalement exclu du framework tout comme Direct X. Par contre, ADO.NET est bien présent. Et on peut utiliser les Web Services. En dehors de cela, le Compact Framework fonctionne comme le framework standard avec les mêmes notions de JIT, gestion d'erreurs, de garbage, etc.

Un ASP.NET pour mobiles !

ASP.NET ne fonctionne pas tel quel sur les terminaux mobiles. Microsoft a choisi de créer un ASP.NET spécialement dédié : ASP.NET Mobile Controls. Il peut fonctionner avec une multitude de terminaux et de plates-formes mobiles : WAP, PDA, Smartphone, Symbian, Palm, i-mode, etc. Il suffit d'avoir le bon Device Adapters. ASP.NET Mobile Controls est directement incorporé dans le .NET

DOSSIER



Framework (version 1.1). Son but est de pouvoir créer rapidement des applications Web dynamiques. Il contient des MobileWeb Forms Controls générant le code pour les différents terminaux, un Mobile Internet Designer pour concevoir l'interface, etc.

Conclusion : que l'on soit sur un desktop ou un mobile, désormais, on a la possibilité d'utiliser un

modèle de développement unique et unifié autour de .NET et de son framework. Malgré tout, il ne faut pas s'attendre à des miracles. Il faudra faire le sacrifice de certaines fonctions pour pouvoir porter ses applications sur des PDA ou smartphone. Pour ne pas s'embêter on peut aussi passer par des Web Services

FT

"Un modèle de développement unifié"

INTERVIEW AVEC ANTOINE DRIARD
(Chef de produits Outils de développement, Microsoft France)

Avec les importantes annonces et sorties se succédant depuis quelques mois chez Microsoft, Programmez ! a voulu en savoir un peu plus sur la stratégie et l'approche de l'éditeur envers les outils de développements et les systèmes dédiés à l'univers de l'informatique mobile.

Programmez : *Le TabletPC est sorti depuis quelques mois. Peut-on déjà tirer premier bilan et dévoiler un peu de son futur ?*

Antoine Driard : Aujourd'hui, il se vend mieux que prévu (ndlr : il s'est environ vendu 72 000 unités de TabletPC entre novembre et fin décembre 2002, chiffre IDC). On attendait entre 5 et 10 % des ventes des laptops. Il s'agit d'une première version. Nous livrons le système et un SDK complet. Il permet à des ISV d'intégrer leurs applications, d'intégrer l'encre. On peut interagir avec son application. Pour son évolution future, c'est encore un peu tôt.

Le fait d'intégrer des fonctions TabletPC génère-t-il des bugs ? De plus, comment bien adapter une application aux fonctions TabletPC comme les notions d'encre ?

AD : Une telle intégration ne génère pas spécifiquement de bugs. Il faut savoir où l'encre est utile dans une application. Le développeur aura tendance à vouloir tout mettre à l'encre. Il faut un équilibre entre le stylet et l'encre et l'interface classique. Il ne faut pas sous-estimer cette question. Qu'est-ce que l'on veut faire avec le SDK ? Il s'agit plutôt d'une question d'intégration d'interface.

Il y a deux grandes notions dont on a rapidement parlé : le Ink et le Pen. Sont-elles séparées ou liées ?

AD : Assez liées. Si le stylet est utilisé sans la notion d'encre, il est assez proche à l'usage d'une souris, même si là, il faut analyser la pression, le tracé, etc.

Peut-on créer une seule et même application, une sorte d'application générique, pouvant fonctionner sur un Windows XP normal et un TabletPC ?

AD : Oui. On peut créer un seul exécutable. Celui-ci détecte si on utilise un TabletPC ou non.



Abordons maintenant .NET et en particulier .NET Compact Framework. Comment pourriez-vous le définir ?

AD : .NET Compact Framework est un sous-ensemble de .NET Framework. On a fait des choix pour savoir ce que l'on garderait et ce que l'on ne garderait pas. On a énormément discuté et travaillé. Au final, le .NET Compact Framework fait environ 1 Mo.

En quoi peut-il servir à un développeur faisant des applications mobiles ?

AD : Aujourd'hui, un développeur, pour créer des applications mobiles, devait connaître des IDE, des langages et des bibliothèques spécifiques à chaque terminal ! Un développeur connaissant déjà .NET, doit juste apprendre ce qu'il y a ou pas dans le .NET Compact Framework. On unifie le modèle de développement, tout en restant dans l'idée du terminal intelligent et des clients riches.

Peut-on créer un seul et unique exécutable comme on peut le faire avec le SDK TabletPC ?

AD : Je ne pense pas que l'on puisse le faire. Cela prendra peut-être son sens dans le futur. Aujourd'hui, on attend des applications légères.

Prenons un cas concret. J'ai une application écrite en C#. Existe-t-il une fonction ou un outil me permettant de vérifier automatiquement le code fonctionnant ou non avec .NET Compact Framework ? Cela simplifierait beaucoup le travail du développeur.

AD : À ma connaissance, un tel outil n'existe pas.

Au-delà des aspects programmation, quelle est la finalité de ce sous-ensemble ?

AD : Avec le Compact Framework, on a le sentiment de pouvoir faire ce que l'on fait habituellement. On a gardé l'accès aux données, ADO.NET,

Ipracom : une idée bien française...

Cette jeune société française conçoit une plate-forme P2P dédiée à la communication entre des terminaux mobiles et les postes fixes. Les environnements Ipracom utilisent les réseaux GPRS et sans fil 802.11. Le but est de rendre mobiles les données des utilisateurs. La liaison se fait directement via une connexion P2P. On peut même avoir des bloc-notes partagés entre plusieurs utilisateurs nomades, très pratique pour échanger rapidement des données.

Blackberry : l'autre PDA

En dehors des PDA PocketPC, Palm, Linux ou autre, il existe le BlackBerry. La programmation des applications se réalise surtout en Java mais aussi en C++. Le constructeur propose en téléchargement son propre IDE J2ME, avec support MIDP. Pour le moment, l'IDE fonctionne uniquement sous Windows. Deux émulateurs sont disponibles selon le modèle.

les Web Services ou encore XML. Ce qui manque, ce sont les composants serveurs et des éléments graphiques. Il ne faut pas oublier que l'on a un petit écran. La grande majorité des scénarios peut se réaliser avec le Compact Framework.

À quel type de terminaux, le Compact Framework se destine-t-il ? Peut-on doré et déjà l'utiliser avec un Smartphone Windows ?

AD : Actuellement, c'est le PDA qui est visé. Tous les nouveaux PocketPC sont déjà équipés du Compact Framework. Pour les versions plus anciennes, il y a la possibilité de le télécharger. Pour ce qui est du Smartphone, la version est prévue. La problématique est un peu différente. Il faudra attendre la sortie de Smartphone 2003 pour voir un Compact Framework implémenté. Mais, on ne pourrait pas installer le framework sur un Smartphone 2002. Si je prends un Windows XP Embedded, on aura un .NET Framework " normal ", par contre sur un Windows CE.NET, ce sera du Compact Framework. Aujourd'hui, les applications possèdent du code managé et non managé. Le code managé se

généralisera au fur et à mesure dans les applications tierces et Microsoft.

Existera-t-il une mise à jour automatique de Compact Framework lorsque la " version standard " sera mise à jour ?

AD : Oui, les deux évolueront en même temps, mais pas forcément exactement à la même date. Il y aura une réflexion pour savoir s'il y a utilité ou non à ajouter dans le Compact Framework les nouvelles fonctions. Actuellement, Compact Framework ne possède pas l'implémentation de WS-Security, mais la question se posera très rapidement.

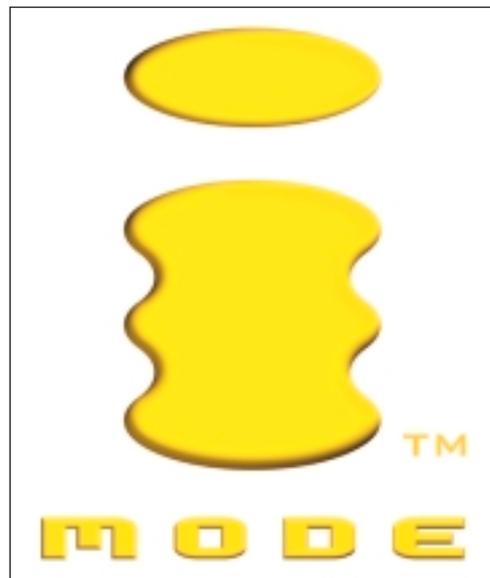
N'est-il pas gênant de ne pas pouvoir bénéficier de WS-Security quand on met en œuvre des Web Services avec Compact Framework ?

AD : Oui, c'est un peu gênant. Mais, aujourd'hui, le Web Service s'utilise plutôt dans le cadre d'un Intranet. La définition de WS-Security est bien figée maintenant. L'implémentation dans le .NET Framework est en bonne voie. ■

Propos recueillis par François Tonic

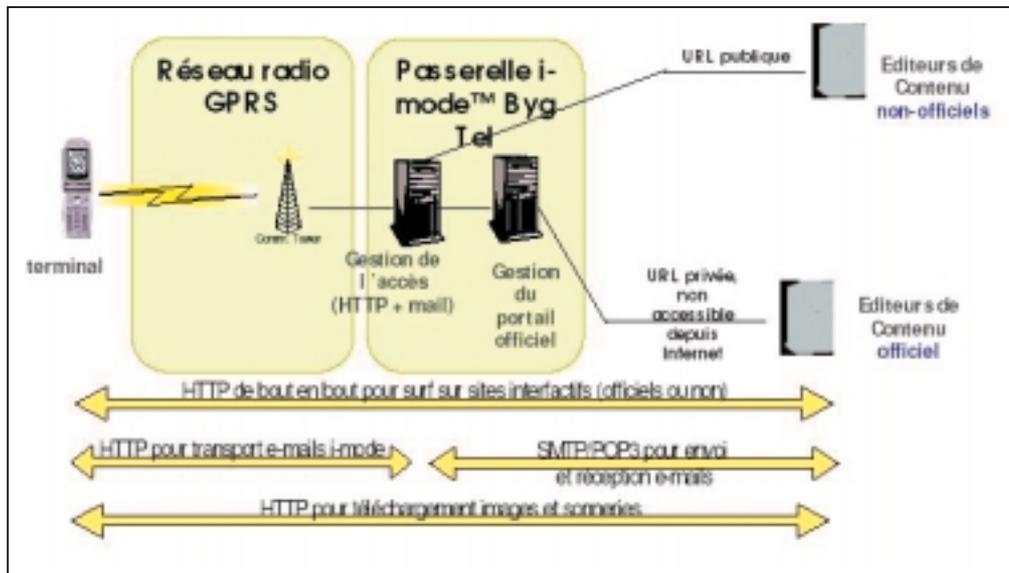
3 • L'AUTRE VISAGE DU SMARTPHONE : I-MODE

La bataille pour la technologie dans les téléphones portables est engagée et chaque acteur tente de s'imposer. Aujourd'hui, il y a trois grands prétendants : Microsoft, Symbian et i-mode. Palm a peu de chances et Linux reste largement en retrait (pour le moment).



Depuis plusieurs mois, i-mode déferle en France après s'être imposé au Japon. L'i-mode c'est avant tout une plate-forme technologique pour téléphone portable. Il fut introduit en début 1999 au Japon. Depuis, il y a plus de 36 millions d'abonnés. I-mode est un format propriétaire. En France, seul Bouygues Télécom l'exploite (avec support GPRS). Pour les transmissions, i-mode utilise les transactions par paquet. Le langage utilisé est le c-HTML, un dérivé du HTML. Contrairement au WAP avec son WML, le c-HTML est bien plus standard. L'adaptation d'un site HTML nécessite donc bien moins de temps. Si à l'heure actuelle, on est encore limité à 4 096 couleurs et à l'absence de frame, de JavaScript ou encore de feuilles de style, les futures évolutions des terminaux i-mode laissent entrevoir de belles avancées dans les prochains mois. Une page i-mode type pèse maximum 10 ko (image fixe ou animée comprise). Le poids est important, car on paie selon les ko ou mo transmis et non au temps passé. S'il existe des sites amateurs, il y a aussi des sites officiels référencés par l'opérateur. Ces sites peuvent posséder une partie payante. L'éditeur se rémunère dessus (l'opérateur versant le montant perçu).

L'i-mode a trois grands marchés : l'e-mail, les sites et les applications Java. Dans une entreprise, on peut imaginer un lien entre le mobile i-mode et la messagerie de l'entreprise qui peut transmettre un mail en cas de besoin. Le mail de l'i-mode est standard. Il est parfaitement envisageable d'adapter une partie d'un extranet en c-HTML. Ainsi, un salarié



pourrait accéder immédiatement au réseau d'entreprise. Pour les sites et services eBusiness, le i-mode représente un nouveau marché potentiel et surtout plus simple à mettre en place que le WAP. Des banques comme le CCF ont déjà ouvert des services i-mode. Concernant Java, les futurs terminaux français doivent l'accueillir. Cette présence fournira de nouveaux arguments à cette plateforme.

Il y a pas mal de critiques envers i-mode : pas de sécurité, nécessité d'une connexion continue, interface utilisateur pauvre. Je serais d'accord sur un point : la sécurité, même si i-mode implémente les

mécanismes de bases de sécurité (liées à http). Sur l'interface, l'apport de Java est déterminant. Côté outil, il y a le choix. Des IDE Web proposent la compatibilité c-HTML. On peut aussi utiliser des convertisseurs de type WebSphere Transcoding Publisher. Pour le déploiement des sites, les sites officiels sont directement référencés par l'opération dans les serveurs i-mode reliés au réseau GPRS. Pour les autres, ils ne sont pas listés et donc, il faut connaître l'URL, mais l'architecture reste unique. L'opération utilise alors un relais du réseau pour établir la connexion à Internet.

FT

Pour aller plus loin

Site officiel français de l'i-mode (avec section développeur) : <http://www.imode.fr>

Site corporate de NTT DoCoMo : <http://www.nttdocomo.com/>

FAQ indépendante sur l'i-mode : <http://www.eurotechnology.com/imode/faq-gen.html>

1er portail européen : <http://www.imocentral.com/i/index.htm>

Norme c-HTML : <http://www.w3.org/TR/1998/NOTE-compactHTML-19980209/>

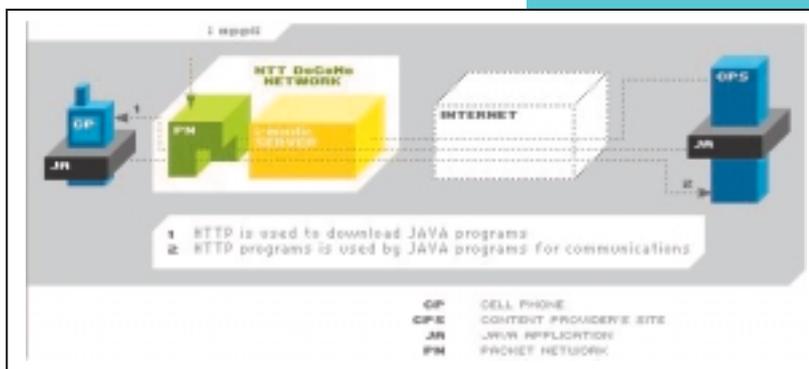
Études de marché sur les mobiles au Japon : <http://www.mobilemediajapan.com/>

4 • JAVA ET LA MOBILITÉ

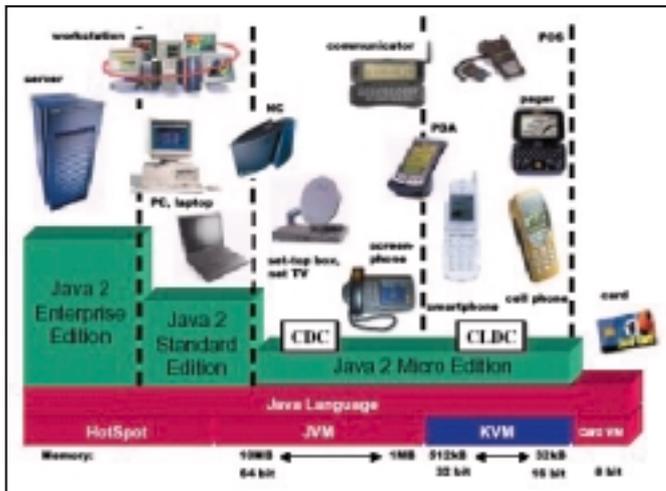


Java s'impose, via sa version Java 2 Micro Edition : J2ME, sur les nouveaux mobiles. En témoigne l'essor des solutions embarquées intégrant J2ME qui permet d'exécuter des applications sur les périphériques mobiles. Que ce soit Motorola, Nokia, Siemens, Sony, NEC, Matsushita/Panasonic, Mitsubishi, Fujitsu ou Samsung, aucun n'a raté l'occasion de sortir ses terminaux compatibles Java depuis le début de l'année 2002. Le marché paraît juteux puisque Nokia prévoit en effet de vendre plus de 100 millions de téléphones Java d'ici fin 2003, soit avec un peu de retard sur ses prévisions initiales. Certains pourraient dire " Enfin ! ". Après plusieurs années d'annonces prématurées, de reports pour cause de mauvaises performances de la machine virtuelle Java ou d'un marché qui se cherchait, ce mouvement d'adhésion général à J2ME s'explique par le fait que les applets mobiles démultiplient les applications et consomment dorénavant peu de ressources. Cela facilite la mise en forme graphique des informations et est utile pour des applications comme : les jeux interactifs, le multimédia ou la

géolocalisation. Les développeurs n'ont plus à réécrire les applications en C++ selon les familles de téléphones portables. Les éditeurs s'engouffrent donc dans la brèche. Par exemple, Borland encourage l'utilisation de JBuilder MobileSet, l'extension de son atelier Java pour le développement de " midlets ", le nouveau nom des applets mobiles. Sun préconise Forte et propose des plates-formes Palm et Symbian, ce dernier intégrant Java en natif depuis la



version 7. À en croire le Gartner Group, J2ME sera exploitée dans 70 % des terminaux mobiles d'ici à cinq ans. La bataille risque d'être rude avec Microsoft, dont l'objectif est de placer son OS dans 25 % des terminaux mobiles d'ici trois à cinq ans, soit 100 millions de téléphones. Déjà quelques PDA (iPAQ, Jornada) - dotés du système d'exploitation de Microsoft - suivent le mouvement et intègrent Java, encore de manière optionnelle. En définitive, la cohabitation sera peut-être inéluctable.



L'environnement Java sur mobiles

Et rien n'arrête la vague Java, pas même le retard des déploiements 3G chez Nokia en Finlande ou chez Telefonica en Espagne. La raison en est simple : les opérateurs parient sur les jeux, comme future killer, application du monde mobile. D'ailleurs, en Europe, les quelques 35 millions d'abonnés des opérateurs qui ont tenté l'aventure i-mode (Telefonica, e-Plus, KPN, Bouygues Telecom) utiliseront bientôt Java sans le savoir. Aujourd'hui, 30 millions de terminaux sont compatibles J2ME : selon le cabinet ARC Group, ils le seront tous d'ici à 2007.

En effet, la plate-forme Java est parfaite pour développer des applications qui fonctionnent ensuite sur les différents systèmes d'exploitation des mobiles. Malgré la domination de Sun sur le marché des machines virtuelles Java pour mobiles, d'autres éditeurs viennent le concurrencer, tels Insignia ou Esmertec. Ainsi, Palmsource a annoncé en juin 2002 la fourniture imminente d'une machine virtuelle Java Insignia dans son système d'exploitation Palm OS 5. Quant à l'éditeur suisse Esmertec, il assure avoir développé " une machine virtuelle dix fois plus rapide, à processeur équivalent, que celles disponibles sur le marché, en compilant les données au lieu de les interpréter ", selon son directeur technique. De bonne guerre, Sun a certifié cette machine concurrente à la sienne. Le PDG historique de Sun, Scott McNealy, sait bien qu'en généralisant Java sur les mobiles, sa société récoltera

une moisson de contrats d'infrastructure de la part des opérateurs. Et la société californienne en a bien besoin actuellement pour rebondir face aux menaces grandissantes sur ses marchés d'origine...

Techniquement, J2ME est une version du J2SE, qui contient le nécessaire pour développer des applications capables de fonctionner dans des environnements limités, tels que les assistants personnels (PDA), les téléphones portables ou les systèmes de navigation embarqués. Un terminal J2ME dispose d'une machine virtuelle KVM au-dessus de laquelle se trouve la configuration CLDC (Connected Limited Device Configuration). Elle définit les capacités minimales et les bibliothèques disponibles sur tous les terminaux, comme la connectivité, le modèle de sécurité et les entrées/sorties en plus des fonctions de la KVM. CLDC est donc le plus petit commun dénominateur de la technologie Java applicable à une grande variété de terminaux mobiles et garantit la portabilité et l'interopérabilité du code entre les différents types de terminaux mobiles CLDC. Les applications Java pour mobiles utilisent la machine virtuelle KVM du téléphone pour être exécutées, de la même façon qu'une Applet Java utilise la machine virtuelle (JVM) d'un navigateur web.

Les applications Java utilisent un profil qui leur est propre, qui se situe au-dessus de CLDC. Ce composant nommé MIDP, Mobile Information Device Profile, prend en charge les fonctionnalités de plus haut niveau, en définissant les bibliothèques qui sont utilisées par les développeurs pour écrire des applications pour un terminal spécifique ou pour un ensemble de terminaux. Cette architecture permet ainsi à l'interface utilisateur d'une application J2ME tournant sur un téléphone intelligent d'être différente de celle tournant sur un Palm Pilot.

La convergence JVM - OS dédiée - processeur : tout pour la performance !

La convergence tant attendue entre machine virtuelle spécifique, système d'exploitation dédié et processeur est d'ailleurs enfin sur les rails. " Pour les terminaux de téléphonie cellulaire, le problème n'est pas tant l'exécution des applications J2ME, que leur manque de sophistication ", estime le fondateur de Savage, spécialiste de systèmes d'exploitation pour la plate-forme J2SE. Dans le cas de J2ME, l'optimisation bute sur la limitation des capacités mémoire inhérentes à ce type de terminaux. Et depuis la naissance de Java, les initiatives se sont multipliées pour résoudre ce problème, et tenter de faire de Java le langage objet de référence des terminaux mobiles.

La première tentative a d'ailleurs été la mise au point de la KVM, " K " pour une mémoire limitée mesurable en kilo-octets. Par exemple, le traitement des classes d'erreurs et le nettoyage de la mémoire sont réduits à leur plus simple expression et ni RMI (Remote Method Invocation), ni JNI (Java

Native Interface) ne sont supportés. En contrepartie, la KVM se concentre sur l'optimisation du téléchargement des classes Java.

La première version de la KVM connaît son baptême du feu au Japon. Fin 2000, l'opérateur DoCoMo réalise le premier déploiement mondial de téléphones J2ME. Plusieurs "jeunes pousses" s'approprient alors la KVM pour améliorer l'implémentation de référence de Sun, ce qui passe la plupart du temps par un couplage de la KVM à un OS temps réel. Elles optimisent ainsi le développement d'applications basées sur le profil MIDP, troisième composante de J2ME. De son côté, DoCoMo bloque la taille des applications à 20 Ko pour que les terminaux allouent davantage de ressources à la qualité d'affichage. Mais cette optimisation purement logicielle ne profite pas encore d'un renfort matériel. Pourtant, dès 1996, Sun lance en fanfare une ligne de trois processeurs dédiés, mais sans succès, principalement à cause de leurs consommations excessives en énergie.

Aujourd'hui, l'intégration de Java à un processeur générique ouvre une voie prometteuse. Le pionnier ARM, dont les processeurs équipent déjà les terminaux mobiles, monte en puissance sur sa technologie Jazelle qui fait un véritable tabac auprès des fabricants de processeurs. Son atout : exécuter dans le même processeur un ARM 7 - et, bientôt, un ARM 9 -, le bytecode Java, et le code natif de l'OS. Sun vient justement de s'associer à ARM pour porter une nouvelle génération de machines virtuelles sur Jazelle et sur l'OS de Symbian. La boucle entre optimisation matérielle et logicielle est donc achevée.

Le point de vue des développeurs

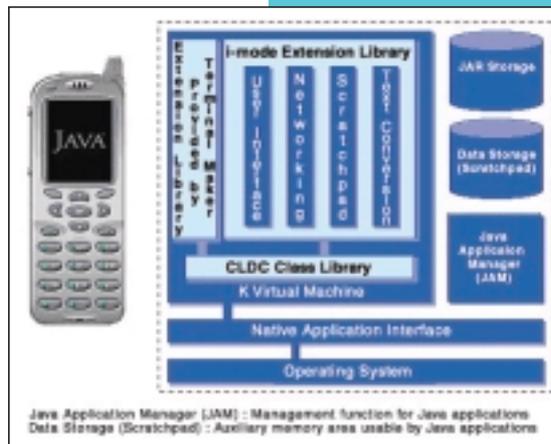
Même si une majorité de développeurs sur les plates-formes mobiles se rejoignent sur le fait de dire que Java est bien "the next big thing", la promesse originelle de Sun "Write once, Run anywhere" est encore loin d'être une réalité. Le langage Java a été conçu dans la première moitié des années 90 dans le but d'exécuter le même code source Java sur toute plate-forme munie d'une machine virtuelle. Cet impératif est assez bien respecté sur les PC et les serveurs mais il n'en va pas de même sur le marché non standardisé des terminaux mo-

biles. Et sans cette nécessaire homogénéisation, l'éditeur de contenus Java ne peut pas espérer amortir rapidement ses investissements en développement. Il suffit d'aller sur le site de Sun pour constater qu'il y'a près de 60 terminaux compatibles avec J2ME, mais ils ont quasiment tous des résolutions d'écran ou un nombre de couleurs différents.

Plusieurs éditeurs de jeux sur mobiles reconnaissent que porter un jeu Java d'un terminal à un autre peut augmenter de 20 à 25 % leurs coûts de développement. De ce fait, pour limiter la gabegie financière, ils préfèrent limiter leurs efforts aux terminaux les plus diffusés, sans exploiter les capacités étendues de mobiles plus coûteux possédant par exemple une plus grande résolution d'écran.

Maintenant, un signal fort pourrait faire réagir Sun en voyant une société comme Qualcomm et la dernière mise à jour de sa technologie Brew, pour Binary Runtime Environment for Wireless, qui permet d'exécuter des applications écrites par exemple en Macromedia Flash. Il est en effet de notoriété publique que le langage Flash, de par ses capacités vectorielles, s'adapte facilement aux différentes résolutions d'écrans et qu'il est très rapide.

Marc Cernes



Pour aller plus loin

J2SE (notamment pour le compilateur) : <http://java.sun.com/j2se/1.4/>

Documentation J2ME : <http://java.sun.com/j2me/docs/>

Configuration CLDC : <http://java.sun.com/products/cldc>

Profil MIDP : <http://java.sun.com/products/midp>

Emulateur Java for i-Mode : <http://www.zentek.com/i-JADE/en/index.html>

Emulateur i-Mode fonctionnant sous I.E : <http://www.x-9.com/mimic/default.asp>

La technologie Jazelle d'ARM : <http://www.arm.com/armtech/jazelle>

La technologie Brew de Qualcomm : <http://www.qualcomm.com/brew/>

NTT DoCoMo : <http://www.nttdocomo.com>

10 millions de téléphones avec Java en Standard

INTERVIEW DE ALEXIS MOUSSINE-POUCHKINE
(Consultant Java/SunONE, Sun France)

Depuis quelques mois, Java se fait bousculer sur le marché de l'informatique mobile. Il fallait en savoir plus et connaître l'autre mobile Java de Sun.

Programmez : De quelle manière Sun prend-il en compte les terminaux mobiles et comment se décline l'offre Java ?

Alexis Moussine-Pouchkine : Sur Java, il y a trois éditions, J2SE, J2EE et J2ME. J2ME est un peu à part des autres. Il doit fonctionner dans des contraintes

de ressources limitées. À cela, on peut ajouter trois autres familles : le CLDC, le CDC et éventuellement JavaCard / JavaRing. Les deux derniers sont intéressants pour la sécurité. Les approches sont différentes pour un PDA et un téléphone. Pour la téléphonie, il existe le MIDP (Mobile Information



Alexis Moussine-Pouchkine

Alexis Moussine-Pouchkine est titulaire d'un Master of Science de l'université de San José. Il est actuellement architecte autour des technologies Java et des produits SunONE, intervenant auprès des clients Sun et traducteur de plusieurs ouvrages techniques sur Java et XML.

Device Profile). Il est sorti en version 2.0 en janvier 2003. Il s'agit d'un API pour le téléphone. Pour Sun, le téléphone portable est l'outil le plus " démocratique ". Aujourd'hui, il y a plus de dix millions de téléphones équipés en standard de Java. Il y a plus de 80 modèles de téléphones incorporant Java. MIDP 2.0 est une évolution majeure : on peut utiliser https, il y a des API orientées jeux pour le son et le rich media. Il y a plus d'objets d'interface. Et puis, il y a eu aussi l'année dernière l'arrivée d'une nouvelle JVM : CLDC Hot Spot. Elle améliore les performances et le temps de lancement des applications. Avant cette nouvelle machine, on parlait de KVM. Dans MIDP 2.0, il manque des éléments. Il n'y a pas encore de Web Services. Si on veut utiliser des services Web, il faut passer par des implémentations tierces comme KSOAP. Les spécifications Web Services seront fournies cet été. Le dialogue avec les cartes à puce et la cryptographie ne sont pas encore disponibles. Ce sera utile pour la sécurité, mais il faudra patienter jusqu'à l'automne prochain. On travaille aussi sur les implémentations de Bluetooth et du Wifi. Là, le problème est avant tout la sécurité. La géo-localisation est un autre chantier important sur lequel on travaille. Mais doré et déjà, MIDP 2.0 est une solide base pour développer des applications.

On n'a pas l'impression que les outils dédiés à J2ME soient très nombreux. Sun a-t-il un IDE spécifique pour créer des applications mobiles avec J2ME ?

AMP : Aujourd'hui, il existe une dizaine d'outils dédiés à J2ME. Chez Sun, il existe SunONE Studio ME. Il s'agit d'un outil gratuit, téléchargeable sur notre site. Il est disponible sur Linux, Solaris et Windows

(ndlr : version OS X prévue). On peut aussi utiliser uniquement le Wireless Toolkit. SunONE Studio se base sur la plate-forme NetBeans, un framework Open Source. On peut développer des applications mobiles et les tester, via une dizaine d'émulateurs disponibles en standard. On peut aussi implémenter un émulateur Palm. Actuellement, SunOne Studio ME utilise MIDP 1.x. La version 5.0 de notre outil qui sortira en juin prochain intégrera en standard MIDP 2.0.

Lorsque l'on a un code existant, peut-on aisément effectuer une migration ?

AMP : Dans J2ME, on a retiré un certain nombre d'éléments : RMI, certaines classes graphiques, etc. Dans un code existant, il faut voir si on utilise des fonctions non présentes dans J2ME. Dans ce cas il faudra sans doute ré-architecturer le code. Sur la partie graphique, on aura à effectuer une conversion. Sur les TabletPC et PDA, on peut utiliser directement J2SE. Plusieurs JVM existent sur ce type de terminaux. Le code fonctionnera sans modification.

.NET Compact Framework sera-t-il un concurrent sérieux pour Java et surtout J2ME ? Comment Sun le perçoit-il ?

AMP : J'aurai deux types de réponses. Cela fait plusieurs années que Java est présent dans les téléphones (et les PDA). .NET Compact Framework est très peu implémenté. D'autre part, la portabilité du code est du domaine de Java. Java est fait pour être portable et très sécurisé. Il n'y a pas de virus Java. C'est très différent avec la technologie Microsoft. Smartphone (Windows) est peu présent. ■

Propos recueillis par François Tonic

5 • DÉVELOPPER SOUS / POUR PALM OS

Les environnements de développement pour Palm se distinguent en fonction du langage proposé (standard ou propriétaire), de l'équipement exploité (palm ou autre système hôte : pc / mac) et du type de programmes générés : interprété, compilé en pseudo code (nécessite un moteur d'exécution ou une machine virtuelle) ou nativement (autonome). Les principaux langages s'avèrent actuellement supportés : Assembleur, Basic, C, C++, Forth, Java, Lisp, Logo, Pascal, Smalltalk, etc. Les plus populaires bénéficient de multiples déclinaisons (commerciales ou non). Le classement de l'intégralité des produits disponibles - au-delà de son aspect rebutant - ne peut raisonnablement s'envisager : trop volumineux. La présentation proposée considère les motivations des développeurs et suggère une réponse

technique avisée (dotée d'un budget adapté) pour le hobbiste, l'amateur averti (ou technicien non spécialisé) et le professionnel aguerri aux plates-formes embarquées.

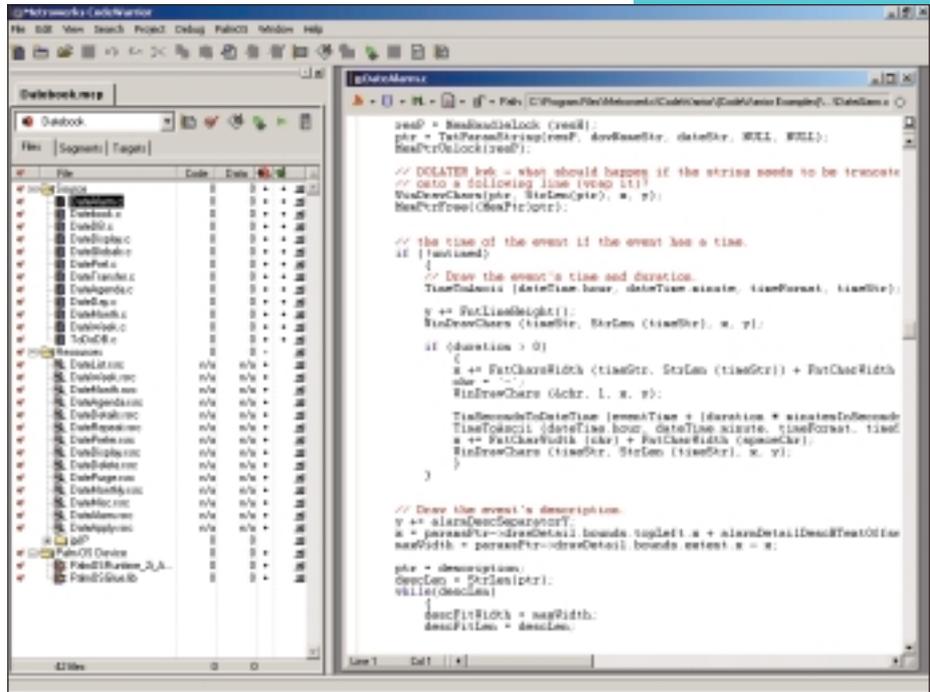
L'entrée en matière, accessible aux néophytes, s'effectue facilement à l'aide de cbasPad. Interpréteur Basic, il permet l'écriture et l'exécution de petits programmes directement sur tout Palm et son mode d'exécution immédiat permet de le substituer à toute calculatrice scientifique. L'auteur en décline une version "Pro" (~20\$), nommée HotPaw, dotée de nombreuses extensions : fonctions de tracés 2D et 3D, gestion du son de la couleur, etc. PP permet aux adeptes du Pascal de compiler (en 68k) n'importe quel code source standard basé au choix sur Palm OS (fenêtres, menus, Âc) ou VF-Sdos (émulateur de DOS). Il offre ainsi un mode console (40x26) et gère les entrées sorties standard (stdio) : idéal pour les habitués (ou nostalgiques) du couple MS-DOS / turbo pascal. La sépa-



cbasPad

DOSSIER

ration des associés de IndiVideo (éditeur de OnBoardC), amène son auteur (Roger Lawrence) à offrir généreusement à la communauté l'ensemble de ses sources. Le produit devient alors libre (en mars 2002). OnBoardC, littéralement "C embarqué", repose sur les API du Palm et produit des exécutables autonomes particulièrement performants, exemple : PP est réalisé en OnBoardC. L'amateur dispose d'un goût prononcé pour la programmation sur Palm et des compétences requises : C, C++ et / ou Java. Disposé à investir un minimum (de 20\$ à 85\$) pour assouvir sa passion, les produits OrbWorks ne peuvent que retenir son attention. PocketC (C) et OrbForms Designer (C++) proposent un véritable environnement de développement intégré (IDE) sous Windows. L'usage d'une syntaxe simplifiée et d'instructions puissantes facilite la réalisation d'applications de qualité et véloces (bien que dépendantes d'un moteur d'exécution). Dans le même esprit, Java s'avère



Metrowerks CodeWarrior



OnBoardC

accessible sous différentes formes : JBuilder Handheld (toute la puissance de J2ME / KVM : du "vrai" java), Jump (génère du 68k avec une bibliothèque spécifique Palm : ensemble de classes hors norme), VisualWaba / Waba / SuperWaba (java "palmisé" performant avec classes et machine virtuelle), etc.

Le spécialiste du développement Palm ne retient qu'un seul nom (il est d'ailleurs déjà informé) : Metrowerks CodeWarrior pour PalmOS Platform (à partir de 399\$). Il s'agit de l'environnement ultime (C / C++) doté de compilateurs performants (68k / ARM). Il supporte l'intégralité des SDK fournis par les constructeurs, expressément générés par ceux-ci à son intention : HandEra, HandSpring, Palm, Sony, Symbol, etc. La distribution inclut (à titre d'exemple) le code source des applications usuelles du Palm : Agenda, Bloc-notes, Carnet d'adresses et Liste de tâches. Quasiment tous les éditeurs de langages dédiés à Palm OS en permettent l'extension par l'intermédiaire de bibliothèques, modules, ressources réalisés en CodeWarrior et fournissent à cet effet un squelette de projet. A noter le support – on ne peut

plus réactif – de Ben Combee sur les groupes de discussions, dont codewarrior.palm et ceux de PalmSource (contôlés par Palm).

Des produits intermédiaires méritent également le coup d'œil (voir la liste des liens), en vrac : AppForge, CASL, Falch.net DeveloperStudio, NS Basic et Satellite Forms. Parmi l'ensemble des nominés, les deux produits retenus (qui doivent au minimum être évalués avant un décision d'achat se nomment : Metrowerks CodeWarrior (vainqueur toutes catégories) et OrbWorks OrbForms Designer (extensible à l'aide du premier).

Le Web intelligent...

Le Web Clipping (ou Palm Query Application) est une solution de communication mobile. L'architecture inclut des applications clientes (fichiers ".pqa") qui fonctionnent sur un assistant numérique portable à base de Palms OS et de serveurs proxy pour gérer la transaction entre l'application cliente et ses serveurs. L'application cliente téléchargée, une fois implantée, ne récupère plus du site fournisseur que les données utiles à son actualisation. Le code d'une page HTML consultée comporte à cet effet des liens vers les applications du palm à activer, exemple : `Memo ` lance le bloc-notes. Une seconde méthode permet d'effectuer un branchement ponctuel (ref="palmcall") vers une application opérant un traitement particulier : mise en forme de données, etc. L'application



OrbWorks PocketC (compilation et execution)



OrbWorks PocketC (source dans le bloc-notes)

Quelques liens intéressants

- AppForge (Visual Basic : commercial)
<http://www.appforge.com>
- CASL (propriétaire : commercial)
<http://www.cassoft.com>
- cbasPad (basic : gratuit, fin de vie)
<http://www.nicholson.com/rhn/palm.html>
- Falch.net (basé sur PRC-Tools : commercial)
<http://www.falch.net>
- JBuilder Handheld (java : commercial)
<http://www.borland.com/jbuilder>
- Jump (java : gratuit, mort)
<http://hewgill.com/pilot/jump>
- Metrowerks CodeWarrior (c et c++ : commercial)
<http://www.metrowerks.com>
- NS Basic (basic : commercial)
<http://www.nsbasic.com>
- OnBoardC (c : gratuit, communautaire)
<http://onboardc.sourceforge.net>
- OrbWorks (c et c++ : commercial : PocketC, OrbForms Designer)
<http://www.orbworks.com>
- PalmOS Pascal Compiler (pascal : gratuit, auteur dynamique)
<http://pages.total.net/~hkonstas/p.p.html>
- Satellite Forms
<http://www.pumatech.com>
- SuperWaba (java "palmisé" : gratuit, communautaire)
<http://www.supervwaba.com>
- VisualWaba (java "palmisé" : gratuit, en sommeil ?)
<http://www.dmic.fr>, puis cliquez "JAVA pour PDA" (url indigeste : lien lotus domino)
- Waba (java "palmisé" : gratuit, communautaire)
<http://www.wabasoft.com>



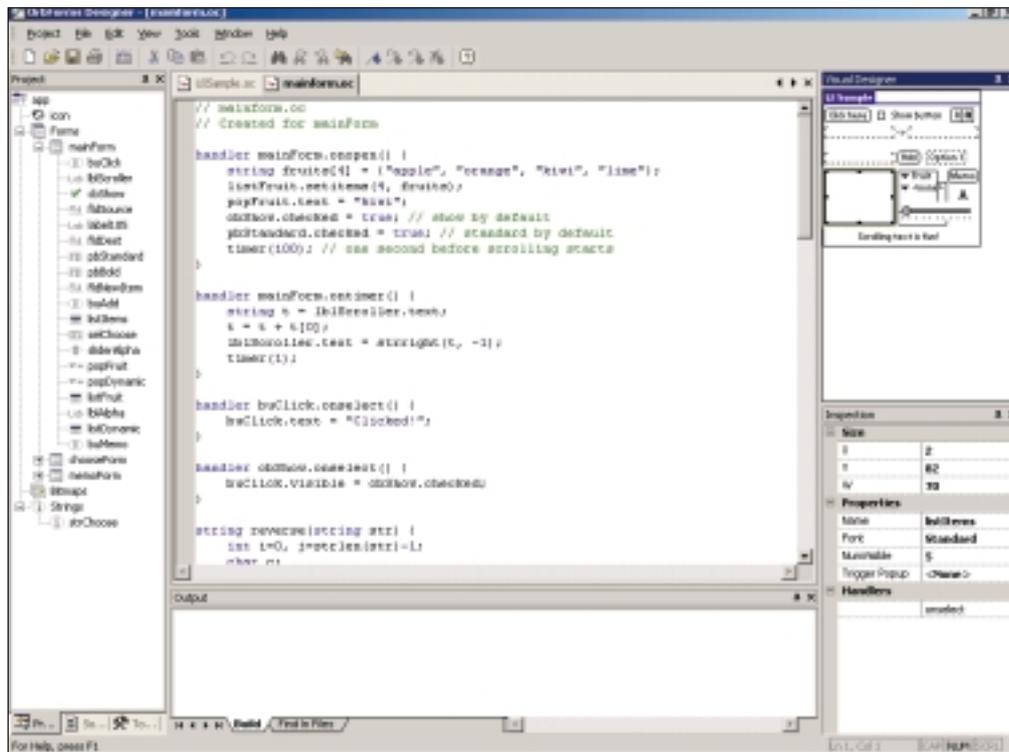
PalmOS Pascal Compiler (Shell)



PalmOS Pascal Compiler (Compilation)



PalmOS Pascal Compiler (VFSdos)



OrbWorks OrbForms Designer

PQA se comporte comme un mini site Web, implanté localement sur l'assistant. L'accès à ses données s'avère par conséquent instantané et gratuit (effectué hors ligne). L'économie se manifeste immédiatement sur l'ensemble des éléments statiques : images, pages d'aides, etc. L'url proposée illustre la mise en œuvre (selon TF1) : <http://www.tf1.fr/mobiles/pda/webclipping/>

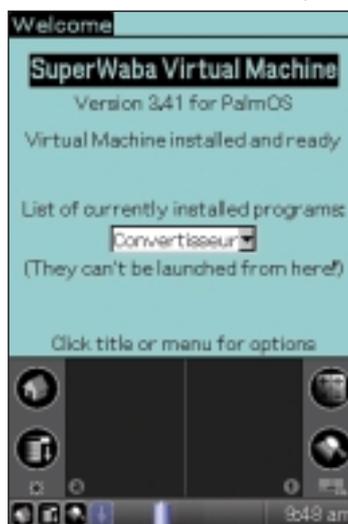
Palm OS 5.0

La dernière version de Palm OS provoque une rupture dans l'architecture matérielle des plateformes hôtes. Délaissant le DragonBall (à base de 68k) au bénéfice du processeur ARM, les équipements proposés rivalisent avec les Pocket PC en terme de puissance. Leurs caractéristiques multimédias s'avèrent inédites (pour des Palms) et offrent ainsi l'accès à un écran couleur d'une meilleure résolution (jusqu'à 320x480, 16 Bits), à une gestion sonore digne d'intérêt et la normalisation de l'usage des périphériques de stockage externes : memory stick, etc. La plateforme reste compatible avec l'existant, grâce à la présence d'un émulateur qui simule le fonctionnement du processeur DragonBall (33Mhz) et trans-

forme les appels systèmes en leurs équivalents (en mode ARM à 200Mhz). Les applications traditionnelles bénéficient ainsi d'une augmentation considérable de leur vitesse d'exécution. Elles peuvent effectuer des appels à des ressources en langage machine ARM, grâce à un système d'appels de sous routines dédiées et appréhendent des traitements en temps réel sur le son (MP3) et l'image (vidéo). Deux constructeurs proposent actuellement des assistants estampillés "Palm OS 5 Powered" : Palm (TUNGSTEN) et Sony (CLIEs dotés d'une Webcam).

L'attitude de Sony envers les développeurs constitue le seul point noir. Contrairement à Palm, le SDK proposé pour ses modèles (NX, NZ) interdit l'accès aux principales ressources multimédia : son et Webcam. Il est regrettable de constater que les dernières évolutions du SDK (4 à 5 et correctif de la version 5) retirent successivement les éléments nécessaires à l'usage de MSA (Memory Stick Audio), puis de SSL (Sony Sound Library). Par conséquent, le programmeur ne peut plus intégrer la lecture de fichier audio (MP3 et WAV) sur un Sony Clié sous Palm OS 5 (alors que cela reste possible avec les anciens modèles sous Palm OS 4).

Jean-Marc QUERE



SuperWaba

DOSSIER

STRATEGIE : "Il est crucial pour Microsoft, de convaincre les développeurs"



Mark SPAIN



Laurent Dugimont

Les applications feront le succès des mobiles, pense Microsoft. On cite le chiffre de 30.000 Smartphones vendus en France.

L'éditeur veut démontrer au développeur que ce marché nouveau est une opportunité.

Nous avons rencontré les têtes pensantes de l' "évangélisation" des développeurs : Mark Spain, directeur marketing-Partenaires à Redmond, et Laurent Dugimont, chef de produit pour la France.

Que faites vous pour inciter les développeurs à prendre le risque de se lancer sur ces technologies nouvelles, au lieu d'attendre qu'elles aient fait leurs preuves, et que le marché soit suffisant ?

Mark SPAIN : Il est crucial pour nous de convaincre les développeurs. Les clients attendent du contenu. Regardez le catalogue des applications, en ligne : il y en a plus de 10.500 !

Toutes nos initiatives visent à faciliter la démarche du développeur. Le programme M2M (Mobile To Market) va dans ce sens. Il n'y a aucune redevance à régler à Microsoft. Le seul coût est celui de la certification "designed for windows". Ce label d'un laboratoire donne droit à un logo qui vise à rassurer les opérateurs de télécoms. 200 applications bénéficient déjà de cette homologation. Il coûte 500\$ et il sera gratuit pour les 500 prochaines applications! Nous avons segmenté la communauté des développeurs en deux :

- ceux qui développent déjà pour les mobiles, avec

d'autres technologies. Ils comprennent les contraintes, notamment de code optimisé;

- les 7 millions de développeurs qui maîtrisent les outils VB mais ne connaissent pas les contraintes du mobile : contraintes de mémoire, rôle du réseau, interface utilisateur...

Nous avons conçu le Kit : cette valise qui comprend un SmartPhone et tous les outils de développement, pour un prix de 599\$!

Bill Gates a présenté ce Kit à la Conférence Développeurs de La Nouvelle Orléans, le 19 mars, avant Paris, Tokyo, Séoul et Sydney. Il a répondu à la demande des développeurs : rendre plus rapide et plus simple le développement !

La stratégie de Microsoft est elle désormais orientée d'avantage vers des applications spécifiques, sur mesure, plutôt que standard ?

Nous avons une culture double. Nous savons travailler avec les grands comptes. Mais les programmes conçus pour les mobiles sont généralement clairement élaborés pour être vendus en boutique.

Quelle serait la "Killer Application" pour le PocketPC ou le Smartphone... ?

On s'est réuni pour y réfléchir, et on a conclu qu'on était trop vieux autour de la table... ! Je ne veux pas dire pour autant que ce serait un jeu. ■

Propos recueillis par Jean Kaminsky

Agena 3000 : Traduction EDI

Agena 3000 propose une nouvelle version de son progiciel de traduction des Echanges de Données Informatisés (EDI) basé sur XML. Adoptée par des sociétés prestigieuses (Polo Group, Rosières, Mer Alliance, ...), elle s'appuie essentiellement sur l'atelier de génie logiciel WinDev7.5.

Lorsque Agena 3000 rencontre PC Soft...

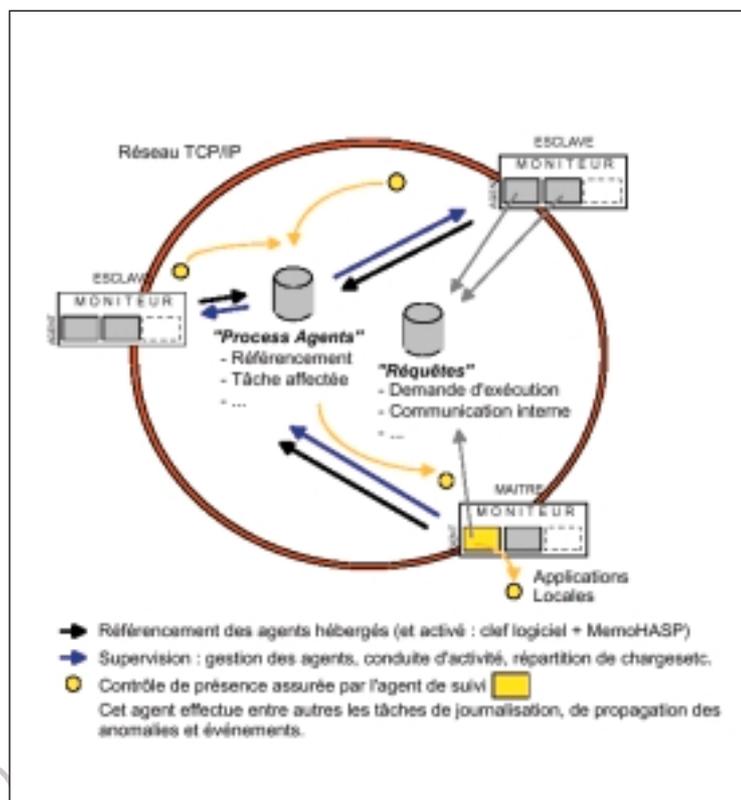


Historiquement, AGENA dispose de deux produits EDI, le premier sur mini (AS/400) et le second sur micro (PC / FoxPro). L'usage de consoles 5250 et de systèmes d'exploitation 16 bits ne correspondant plus aux goûts actuels, les produits s'avèrent respectivement en fin de vie (2000). L'intégration des comportements induits par l'évolution des normes (principalement basées sur XML) et l'usage massif d'architecture reposant sur IP (messagerie, Internet, ...),

s'illustrent par la volonté de réaliser un nouveau produit commun aux deux plateformes. Les deux priorités inhérentes à l'expression des spécifications résident donc en une dissociation du produit de la plateforme hôte (système d'exploitation, base de données) et d'accès (exploitation, administration). La démarche de veille technologique sous-jacente, met alors en évidence l'atelier de génie logiciel de PC Soft (WinDev7) qui retient l'attention de par ses caractéristiques. La facilité d'accès au produit, son adéquation (exhaustif) et son indépendance vis-à-vis des différentes bases de données et serveurs SQL permettent aux concepteurs de se concentrer d'avantage sur leur métier (EDI) et non plus sur l'outil. La démarche s'engage finalement sous la forme d'une Interface Homme Machi-



ne reposant sur un serveur Web (Tomcat / servlets Java). L'application s'avère ainsi accessible à partir de tout navigateur Web en local, ou par l'intermédiaire d'une connexion distante (voire Internet). Java permet d'exploiter le framework "maison" employé dans le



Agena 3000, fiche d'identité

Le rachat de la société Art Anjou (fabrication de meubles) par Gautier, privilégiant une informatique orientée BULL, provoque le licenciement des informaticiens maisons : Alain Guimberteau, Philippe Maudet, Pierre Sanier. Fidèles à IBM et particulièrement convaincus par les performances du nouvel IBM 38, ils saisissent l'opportunité pour s'engager dans la création d'une société de prestations de services en informatique. En l'absence de clients et fonds, seule la synergie de leurs compétences respectives (cursus commerciaux, techniques et scientifiques) en permet la concrétisation. "Agent IBM" dès le début (1980), la société se concentre sur les produits de gestion et l'usage des codifications liées à l'emploi des codes à barres (GENCOD). Les produits édités s'avèrent ainsi compatibles avec les normes employées par l'industrie et la grande distribution. La société emploie actuellement près de 100 personnes (filiales incluses) et assure des prestations à l'échelle nationale et internationale. Partenaire de GENCOD et travaillant étroitement avec ses différents groupes de travail, AGENA s'initie à l'EDI en substitution à l'échange des commandes sur disquettes lors du déploiement d'ALLEGRO.

native XML



PC Soft, éditeur de WinDev

Etudiant, Patrice Prut travaille très rapidement dans le domaine de l'informatique parallèlement à ses études. Il constate alors un manque manifeste d'organisation dans les différentes structures fréquentées. Son projet de dernière année (5ème) en école d'ingénieur consiste en la création d'une SSII dotée de méthodes et surtout d'outils adaptés. Il se concrétise en PC Soft à l'aide d'amis animés des mêmes ambitions. Dès le départ, l'entreprise se dote d'outils de développement à vocation interne pour répondre de façon optimale à ses différentes missions. Lors de démonstrations et recettes, les visiteurs et clients s'intéressent particulièrement au procédé "HighScreen" développé pour dissocier la réalisation des écrans de la programmation pure. Les principaux clients ne tardent pas à demander une licence du produit pour l'employer en interne, au bénéfice de leurs propres équipes de développement. PC Soft évolue ainsi progressivement du domaine de la prestation de service à celui d'éditeur d'outils de développement. L'énergie de Jean-Michel Majester (principal auteur) permet ainsi à HighScreen d'adopter rapidement les principales plates-formes de DOS (1986) à Windows (1993) via Unix et OS/2. En parallèle, la commercialisation des solutions HyperFile et HyperPrint (1989) débute et connaît un essor significatif auprès des développeurs Basic, C ou Pascal. L'arrivée de Windows provoque la refonte de l'ensemble en un atelier de génie logiciel (développez moins ; utilisez d'avantage) : WinDev.

cadre des autres produits (gestion commerciale GENACOD sous WebSphere, ...) et de profiter de ses évolutions. Les applications bénéficient ainsi d'une unicité d'accès (Web) et d'une ergonomie similaire (framework). Les traitements métiers consistent en l'ensemble des opérations requises à la conversion des données reçues (via messageries ou Réseau à Valeur Ajoutée) en informations exploitables localement et vice-versa. De par leur importance (volume, puissances de calcul et rectitude requise), ils sont intégralement confiés à WinDev sous la forme d'un "cluster". Un ensemble d'ordinateurs en réseau TCP/IP héberge l'application développée sous la forme d'une grappe d'esclaves gérés par un maître. Des agents, correspondants aux différents processus, s'exécutent en simultanément (multitâche).

Leur répartition s'effectue en temps réel, selon la disponibilité des équipements hôtes et l'adéquation de ceux-ci (dotés d'un périphérique particulier). Les agents reçoivent leurs tâches à exécuter lors de leur activation. En retour, ils postent des requêtes à l'attention du maître, afin de demander l'exécution de tâches particulières, ou d'indiquer un état d'avancement. Le système effectue son propre suivi. Il identifie les anomalies inhérentes aux traitements opérés ou liées à une panne, s'assure de la remontée de l'alerte aux intéressés (humains et applications : courriel,

requête HTTP, envoi SMS, ...) et le cas échéant prend en charge la mise en œuvre d'un mode de fonctionnement dégradé. L'un des tests de mise en service consiste d'ailleurs à éteindre (directement avec l'interrupteur électrique) l'un des ordinateurs (indifféremment maître ou esclave) en cours de traitement, afin de vérifier : la diffusion de l'alerte, le redémarrage et la répartition des travaux affectés à celui-ci sur d'autres postes et surtout la prise en compte de ce dernier lors de son redémarrage au sein du cluster, de façon totalement automatisée.

L'usage de WinDev a facilité la mise au point du noyau de l'application (aspect multitâche et cluster) grâce à la modélisation objet. De multiples traductions de document EDI peuvent ainsi s'effectuer en simultanément, être à plusieurs stades d'avancement : traitement des flux en continu. La faculté de WinDev à compiler du code (W-Langage) de façon dynamique, a autorisé la réalisation de scripts pour compléter les opérations de "mapping" (concordance de zone d'un document à un autre) et d'obtenir du code qui s'auto modifie, afin de s'adapter à tous les besoins (notamment en complément de l'usage de XSLT). Suite à 18 mois de développement (à 3 personnes), l'application fonctionne sur 30 sites (groupes agroalimentaires et fournisseurs de la grande distribution). ■

Jean Marc Quere

EDI, mode d'emploi

La mise en œuvre de spécifications imposées, ou une réflexion arrivée à maturité sur la rationalisation de leurs coûts, poussent les entreprises à adopter l'EDI. Les flux usuels concernés sont liés aux principaux actes commerciaux : commandes, avis d'expéditions, factures. Une telle liaison permet d'intégrer ces documents directement dans le système informatique de chaque entreprise. Les opérations de ressaisies - et les risques d'erreurs inhérents - sont ainsi définitivement éliminés. L'utilisation de messages normalisés permet également l'établissement de nouvelles relations commerciales avec un seul et même outil. L'EDI comporte deux subdivisions correspondant à des fonctions distinctes : messagerie et traduction. La messagerie regroupe tous les dispositifs chargés d'acheminer des données (fichiers) entre les différents correspondants. La traduction consiste en l'ensemble des opérations requises à la conversion des données reçues en informations exploitables localement et inversement pour l'émission. Ces opérations requièrent un logiciel spécifique, nommé "Traducteur EDI".

Bientôt, avec Office 2003

InfoPath 2003 : XML à la portée des utilisateurs avancés

Cet éditeur de formulaires XML interface aussi des sources de données relationnelles et XML, ainsi que des services web. Simple d'emploi, il se destine à la fois aux développeurs et aux utilisateurs avancés. Objectif : permettre à n'importe quel collaborateur de l'entreprise de manipuler des données XML.

Nouveau venu dans la gamme Office 2003, InfoPath comble le fossé qui séparait jusqu'à présent la technologie XML et les utilisateurs bureautiques.

L'objectif d'InfoPath est en effet d'apporter plus d'autonomie aux utilisateurs finaux en leur donnant la possibilité de partager et d'accéder aux ressources de l'entreprise, sans avoir recours aux services du département informatique.

InfoPath est à la fois un éditeur de formulaires XML et un outil de création d'interfaces utilisateurs. Mais contrairement aux autres outils de développement de Microsoft, InfoPath est aussi facile à utiliser que Word et Excel.

L'ergonomie des outils bureautiques

Comme InfoPath vise les non informaticiens, son ergonomie est très proche des outils bureautiques existants. Ses menus sont, par exemple, identiques à ceux de Word et d'Excel - Fichier, Edition, Affichage, etc. - et ses principes ergonomiques sont identiques aux outils de la suite Office 2003 : volet Office, clic droit pour obtenir les propriétés d'un objet, mise en forme reposant sur les mêmes menus, publication vers SharePoint Portal Server, etc.

La majorité des opérations s'effectue par simple glisser & déposer et par paramétrage. Bien qu'il soit possible d'ajouter des scripts évolués, aucune programmation n'est nécessaire si l'on s'en tient aux fonctionnalités de base. La majorité des prises de décision sont guidées par un assistant qui encadre l'utilisateur.

Bien entendu, il est possible de reprendre la main à tout moment afin de basculer dans un fonctionnement de type " mode expert " .

Couplage lâche entre présentation et données

Si l'on résume le champ d'action d'InfoPath, ce dernier permet de :

1. créer un formulaire à partir d'un schéma XML ou d'un document,
2. se connecter à une source de données relationnelles,
3. interfacier un service web

L'interfaçage d'un service web, ou d'une source de données relationnelles autorise potentiellement la sélection, l'affichage, l'ajout, la modification et la suppression de données. Bien que ces projets puissent sembler assez différents dans leur nature, InfoPath les aborde avec les mêmes principes de base.

La lecture physique et l'enregistrement des données s'effectuent au travers des " sources de données " que l'on peut résumer au concept de connecteurs.

La collecte et l'affichage des informations s'effectuent au travers de formulaires. Ces formulaires sont appelés des " vues " et correspondent à l'interface utilisateur qui collecte

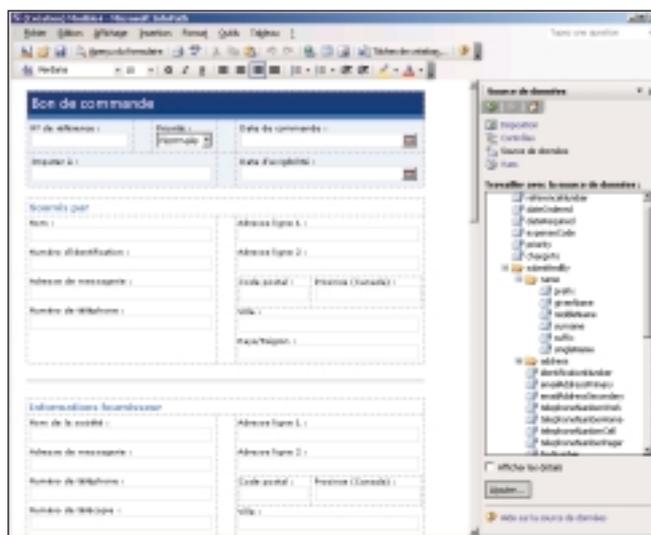
les données, contrôle les saisies, affiche les résultats et propose une version imprimable. La construction de cette interface utilisateur repose sur des contrôles traditionnels tels que bouton, liste déroulante, zone de texte modifiable, etc. et des contrôles moins traditionnels tels que les tableaux et sections extensibles. Une vue assemble des contrôles. Certains de ces contrôles peuvent être liés à une source de données. Une vue peut également publier les données recueillies vers une source de données.

Trois types de vues

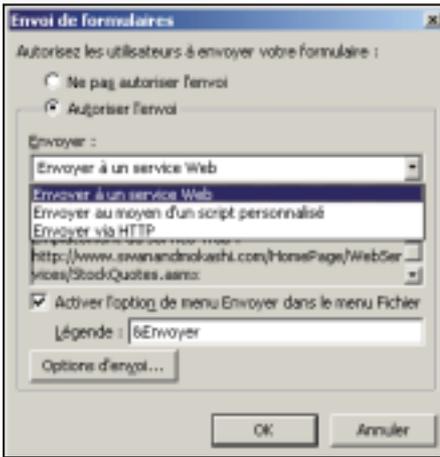
Trois types de vues permettent de construire un formulaire : les "requêtes", les "entrées de données", et les "masques d'impression". La vue "Requête" recueille les informations saisies par l'utilisateur. La saisie est à la fois contrôlée et assistée par InfoPath. Ce dernier s'appuie en effet sur le type de chaque champ SQL / élément XML pour fournir un premier niveau de contrôle de saisie automatique. L'utilisateur ne peut, par exemple, pas saisir des lettres dans un champ censé recueillir un numéro de téléphone. Ce premier niveau de validation peut être amélioré par deux mécanismes : des règles logiques et un système d'assistance à la saisie. Les règles logiques sont générées sans programmation, à partir d'un assistant, sur un principe d'association de conditions proche du filtre automatique

d'Excel. L'assistance à la saisie repose elle aussi sur un assistant qui propose des formats de saisie adaptés au type des champs SQL / éléments XML. Ainsi, le développeur peut, par exemple, choisir dans une liste un format de date proposé par défaut. L'utilisateur verra de son côté apparaître un petit assistant calendrier qui guidera sa saisie.

Contrairement à son nom ambigu, la vue "Entrée de données"



InfoPath permet de créer des formulaires complexes.



Les données du formulaires peuvent être envoyées vers un service web ou une URL (POST HTTP).

affiche les informations qui proviennent d'une "source de données". Mais elle permet aussi de saisir des informations qui seront ensuite envoyées vers une "source de données" par le formulaire.

Les vues "imprimables" correspondent à des états d'impression. On peut donc combiner une utilisation électronique et papier des formulaires créés avec InfoPath.

Sources de données

Les sources de données sont utilisées à la fois pour alimenter les vues et pour réceptionner et stocker les informations saisies. Elles sont créées à l'aide d'assistants très perfectionnés qui

Chaque saisie peut être contrôlée.

permettent de se connecter au contrat WSDL d'un service web, d'importer un schéma XML ou bien encore de se connecter à une base relationnelle.

La connexion aux bases de données relationnelles repose essentiellement sur la création d'une source de données. C'est en effet à ce niveau que l'on paramètre l'accès à la base – provider OLE DB ou ODBC - mais aussi que l'on manipule les données. On peut, par exemple, sélectionner les tables et les champs à inclure, effectuer des tris, créer des relations 1-n, ou modifier directement le code SQL à la main si l'on possède les compétences nécessaires. Encore en bêta lors de

notre test, InfoPath ne supporte pour le moment que les bases Access et SQL Server, mais nul doute que la version finale s'ouvrira à l'ensemble des bases et sources du marché. Côté SOAP, l'assistant services web récupère les méthodes utilisables et les paramètres associés attendus, puis génère automatiquement une vue " Requête " et une vue " Entrée de données " correspondantes. Aucune connaissance de SOAP n'est nécessaire, puisque l'étape la plus compliquée consiste à fournir l'URL du contrat WSDL.

Les sources de données – fichier XML, bases relationnelles et services web - sont également utilisées pour alimenter certains champs des formulaires comme les listes déroulantes, les sections et les tableaux extensibles. Ainsi, même si la vue " Requête " ne repose pas sur une source de données, elle peut néanmoins être enrichie.

Publication vers des services web et des POST HTTP

Le déploiement est extrêmement simple puisqu'un formulaire InfoPath prend la forme d'un fichier binaire autonome à l'extension *.XSN. Ce fichier contient tous les éléments nécessaires à sa bonne interprétation par le lecteur installé sur les postes clients : schéma XSD des données, couche de présentation incluant les contrôles et assistants de saisie, etc. On manipule un formulaire InfoPath comme n'importe quel document bureautique. Il suffit donc d'envoyer le formulaire par e-mail aux utilisateurs cible ou de le partager sur un portail d'entreprise pour le déployer. Les données saisies peuvent ensuite être

indifféremment stockées dans le formulaire, exportées au format XML brut dans un fichier de données accompagné de son schéma, envoyées vers un service web ou bien encore envoyées vers une URL (WebDav et POST HTTP traditionnel). InfoPath concilie donc la

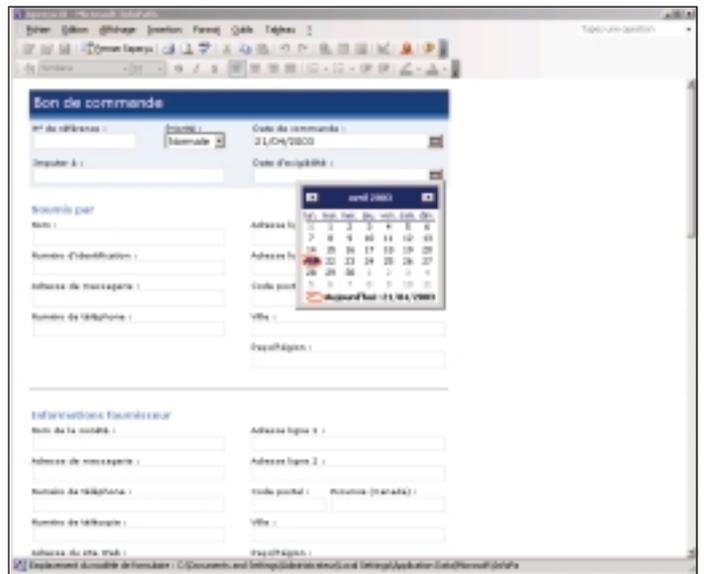
souplesse de manipulation des documents avec la puissance centralisatrice des applications traditionnelles. Un formulaire déployé par e-mail peut, par exemple, alimenter un service web.

De multiples applications

Microsoft propose par défaut un certain nombre de formulaires prédéveloppés tels que " Bon de commande ", " Informations fournisseurs ", " Rapport de vente ", " Note de frais ", etc. L'entreprise peut personnaliser ces modèles de formulaires pour construire des vues adaptées à son activité.

Mais, au vu de ces fonctionnalités, InfoPath ne se limite pas à un simple éditeur de formulaires électroniques. S'il excelle dans cette discipline, on peut aussi l'utiliser pour construire rapidement des interfaces utilisateurs simples, qui se connectent à des données et services web de l'entreprise.

On peut facilement imaginer que le service informatique l'utilise pour exposer des sources de données, afin que les utilisateurs finaux puissent s'y connecter facilement pour lire, ajouter, modifier et supprimer des données, sans avoir à passer par le service informatique.



L'utilisateur final est guidé dans sa saisie.

Utilisé à bon escient, InfoPath devrait décharger le service informatique de nombreux développements, tout en apportant aux utilisateurs finaux une rapidité de mise en œuvre encore inconnue.

David Thévenon

Vérifiez le niveau de sécurité de votre réseau avec un logiciel de simulation d'intrusions.

N'attendez pas qu'un intrus s'introduise dans votre système et auditez dès maintenant votre réseau ! SAINT (*Security Administrator's Integrated Network Tool*) est un outil d'audit réseau Unix parfaitement adapté pour vérifier la sécurité d'un réseau hybride de machines Windows et Linux. Il est le successeur de Satan (*Security Administrator Tool for Analyzing Networks*), outil précurseur dans ce domaine, mais dont la dernière version est dépassée (elle date de 1995).

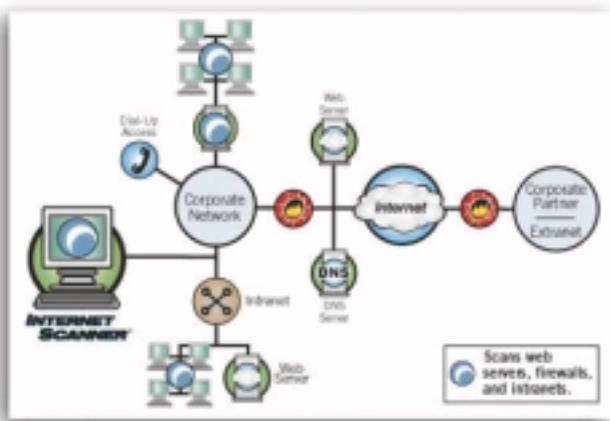
Pourquoi SAINT ? De nombreux vérificateurs de failles existent, comme Nessus sous Linux ou InterNet Scanner (IS) (<http://www.iss.net/>) sous Windows (ou encore TITAN [UNIX] (<http://www.fish.com/titan/>), SARA [UNIX] (<http://www-arc.com/sara/sara.html>), RETINA [Windows] <http://www.eeye.com/html/Products/Retina/>). Chacun de ceux-ci possède des qualités et défauts. Par exemple avec SAINT, l'absence de verrouillage sur des numéros IP permet de tout tester, ce qui n'est pas le cas avec IS. Par contre IS

teste plus de failles (dont en gros la moitié affecte Windows), mais est plus lent au scannage. SAINT et Nessus tournent sous Linux, tandis qu'IS tourne sous Windows. Nessus est d'une approche plus complexe pour le débutant, teste plus de failles que Saint et est capable de découvrir des services de manière dynamique. (figure 1)

Comme de nouvelles failles sont découvertes sans cesse, un logiciel d'audit réseau doit être mis à jour régulièrement (avant chaque audit). Dans le cas d'IS ou de Nessus, il suffit d'ajouter un module. Dans le cas de SAINT il est nécessaire de changer de version de logiciel (le recompiler, puis le réinstaller).

SAINT est un bon choix de départ pour tester la sécurité d'un petit réseau. En effet, il est très facile à installer, possède une interface Web soignée, et l'administrateur, même débutant, pourra balayer son réseau en quelques minutes.

Mise en garde : le sondage d'une machine extérieure à votre propre réseau est considéré comme une infraction aux yeux de la loi si vous n'avez pas préalablement été autorisé à effectuer cet audit par le propriétaire du système cible.

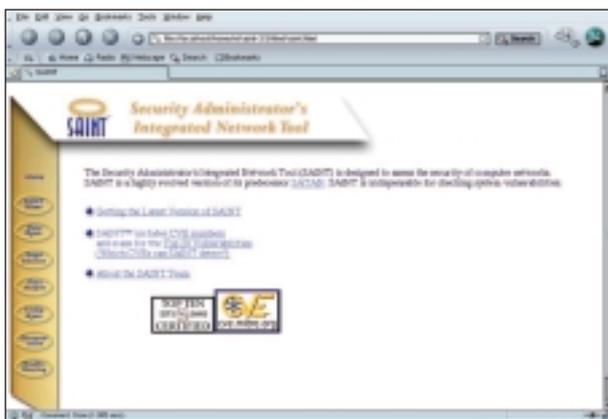


(figure 1)

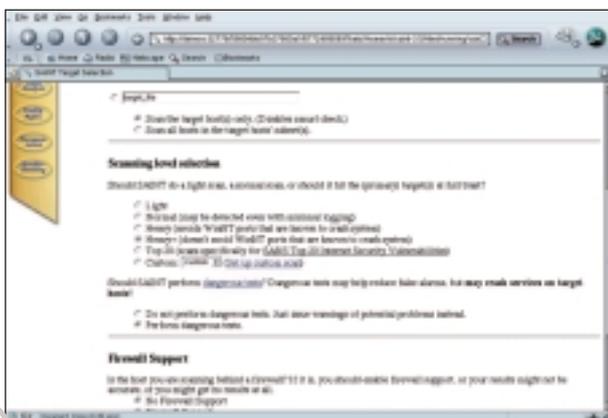
La phase d'installation

Avant de télécharger SAINT vous devez vous assurer de posséder une série de logiciels dont il dépend. Nous avons testé SAINT sous Linux REDHAT 8.0. L'ordinateur est équipé d'un processeur ATHLON cadencé à 1,8 Ghz, d'une mémoire vive de 256 Mo et d'un disque dur de 80 Go. Quoique cette configuration soit très courante de nos jours, SAINT peut aussi fonctionner avec un équipement plus modeste. Néanmoins, un serveur Linux non équipé d'une interface graphique ne conviendra pas. Vous avez besoin d'une version récente de PERL (>= 5.0) et des logiciels Flex et Bison. Pour vérifier les vulnérabilités de machines Windows, un client SAMBA sera installé. Le scanner de port NMAP devra aussi être opérationnel (<http://www.insecure.org/nmap>). L'ensemble de ces éléments est disponible d'office avec les packages livrés par REDHAT.

Un scanner de vulnérabilités, comme Nessus, permet aux administrateurs systèmes de déterminer les services TCP/UDP disponibles sur un serveur, même si ceux-ci n'utilisent pas les numéros de ports standard. L'une des règles fondamentales de la sécurité des serveurs est de désactiver tout service non utilisé par le système. En effet, tout service TCP/UDP actif offre aux hackers une porte d'entrée potentielle pour pénétrer votre système.



(figure 2)



(figure 3)

Vous pouvez récupérer une version d'évaluation de SAINT à l'adresse <http://www.wwdsi.com/saint/>. D'autres versions antérieures sont téléchargeables sur Internet comme la version 3.5 Open Source sur tu cows

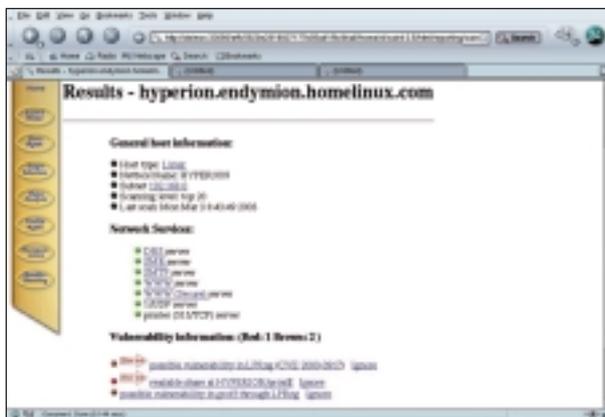
(<http://proxad.linux.tucows.com/internet/preview/51654.html>). Evidemment, plus le numéro de version est élevé, plus les vulnérabilités étudiées sont nombreuses. Cependant la version 3.5 convient déjà parfaitement si vous désirez initier le premier audit d'un réseau.

Il ne vous reste plus qu'à extraire les sources de l'archive, configurer et compiler SAINT (tar -xzf saint-3.5.tar.gz ; cd saint-3.5 ; ./configure ; make all ; make install).

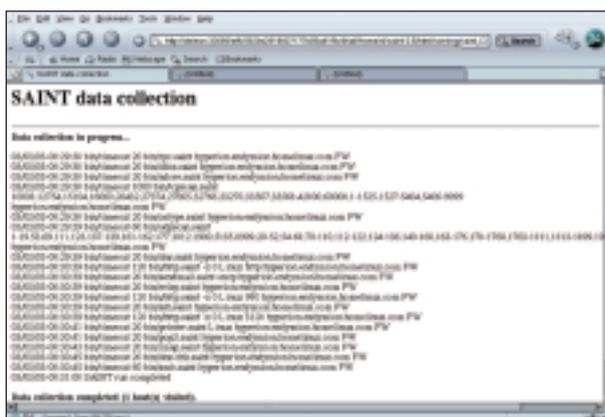
Reste la problématique du navigateur. Lors d'un premier "./configure", SAINT nous a déclaré ne pas trouver de navigateur Internet (...). Il utilise en effet un navigateur comme interface graphique. A ce propos, attention : c'est l'utilisateur "root" qui doit le lancer et vous ne devez pas surfer sur Internet avec cette instance du navigateur, car vous serez potentiellement vulnérable à une attaque. Avec REDHAT 8 vous avez par exemple MOZILLA qui est installé, mais pas Netscape. Nous en avons déduit que nous devons installer Netscape 7.02 (www.netscape.com), et relancé le "./configure" le même message s'affichait. Malgré ce message La compilation a fonctionné. En toute logique cette installation de Netscape 7.02 n'était pas nécessaire, car l'exécution de "./saint" a fonctionné au premier essai.

L'utilisation

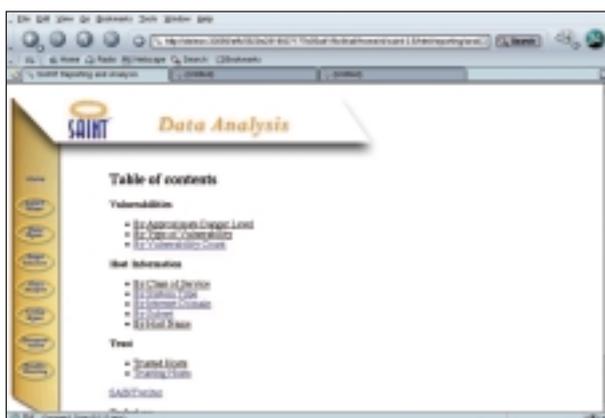
Vous devez vous connecter en tant que super-utilisateur et lancer "./saint" sous un terminal X. Après quelques secondes, Netscape s'exécute pour afficher la page d'accueil. Vous pouvez commencer immédiatement le balayage de vos machines en choisissant l'option "Target Selection" du menu, à gauche de



(figure 4)



(figure 5)



(figure 6)



(figure 8)

votre écran. Vous pouvez maintenant saisir le nom d'une machine cible ou encore son adresse IP. Si vous tapez un nom, assurez-vous en ouvrant une autre fenêtre de terminal, que vous pouvez l'atteindre par un ping. Sinon, si vous êtes certain de son existence, entrez de préférence son adresse IP. Vous pouvez aussi balayer l'ensemble de votre réseau en spécifiant plusieurs adresses IP séparées par un espace, ou bien spécifier un sous réseau entier.

Vous devez maintenant indiquer quel sera le niveau du balayage (de "léger" (light) à "heavy+"). Nous vous conseillons de travailler en dehors des heures de bureau des utilisateurs et d'indiquer "Heavy+" puis d'indiquer que vous autorisez SAINT à effectuer des tests qui pourraient éventuellement "geler" certains services de machines cibles étudiées. (figure 3)

Cette dernière option permettra de clarifier la liste des vulnérabilités en écartant en principe des avertissements sans gravité. (figure 4)

Appuyez sur le bouton "start the scan". SAINT affichera une première fois, un avertissement qui stipule que vous ne pouvez utiliser cette instance du navigateur pour surfer en dehors de votre Intranet. Cliquez sur l'icône "recharger" ce qui affichera l'état d'avancement du balayage en cours. (figure 5)

La visualisation des résultats

Avouons-le : jusqu'ici l'utilisation de SAINT est un vrai plaisir de convivialité, et il en sera de même pour l'analyse des résultats. Lorsque le balayage est terminé, sélectionnez "Data Analysis" qui offre une série de liens affichant les vulnérabilités que SAINT a trouvé. (figure 6)

Celles-ci sont, par exemple, affichables par degré de gravité ("By approximate Danger Levels") ou par type. Vous pouvez les classer par service, par hôte, ou par sous réseau. Si vous cliquez sur le premier lien ("par niveaux de dan-

ger") SAINT vous affiche une liste de nouveaux liens qui correspondent aux vulnérabilités. Si en face d'un lien vous voyez "TOP 10" ou "TOP 20", vous devez vous concentrer immédiatement sur celui-ci, car il s'agit d'une faille souvent exploitée qu'il sera nécessaire de combler d'urgence. (figure 7)

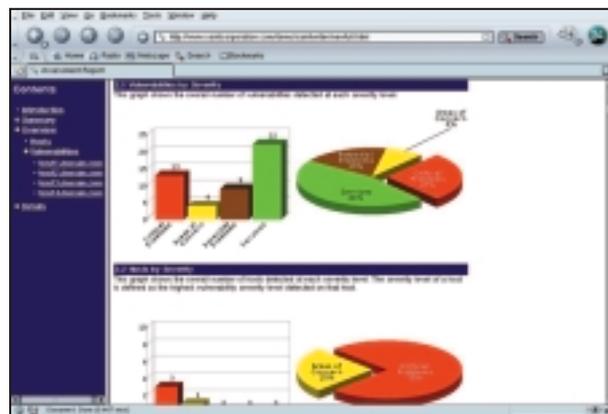
A chaque fois que vous cliquez sur le lien d'une vulnérabilité trouvée, SAINT vous donnera le nom commun de la vulnérabilité, le niveau du danger, les types et ver-

sions des systèmes d'exploitations et logiciels affectés, et enfin les solutions éventuelles. (figure 8)

Pour aller plus loin

Notez qu'un outil complément de mise en forme de rapports du nom de SAINTWRITER existe. Les graphiques générés de cette manière seront interprétables par un non technicien, ce qui est le but de la manœuvre. (figure 9)

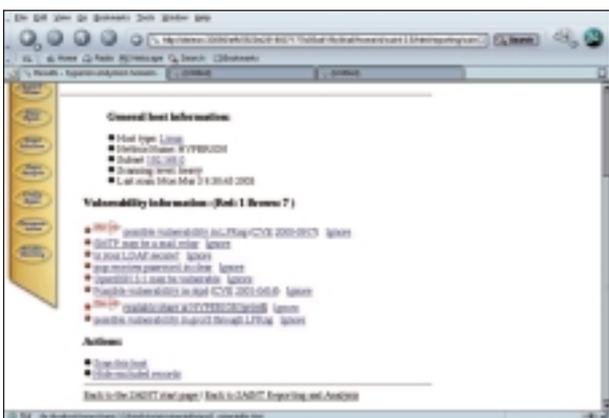
Si votre réseau est d'une taille importante nous vous conseillons de vous équiper d'un outil d'audit capable de se mettre à jour régulièrement par l'intermédiaire d'internet. La mise à niveau des failles détectées par SAINT passant obligatoirement par une recompilation du logiciel, cette situation n'est pas adaptée pour l'adminis-



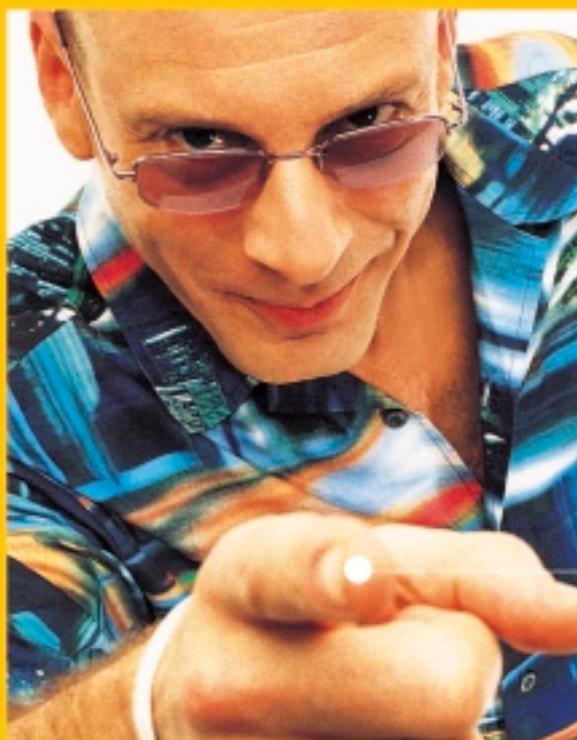
(figure 9)

trateur qui réalise très régulièrement des audits. Une sélection plus poussée des tests à effectuer serait également un plus : cette possibilité existe d'un seul clic pour Nessus, ce qui n'est pas possible aussi finement avec SAINT. Le revers de la médaille dans le cas de Nessus et une mise en œuvre (plus) complexe par opposition à SAINT, qui se révèle un excellent outil pour découvrir sans peine ce qu'est véritablement un "simulateur d'intrusions".

Xavier Leclercq



(figure 7)



Si quelqu'un vous disait qu'il s'éclate en utilisant les outils de gestion de bases de données de son entreprise... Vous en penseriez quoi ?

Ben voyons! Toutefois, vous n'en seriez pas aussi convaincu si vous possédiez des outils de gestion de bases de données! qui vous obéissent au doigt et à l'œil. Sans stress. Avec DBArtisan® par exemple, vous glissez avec une facilité déconcertante sur tous les obstacles que peut induire la gestion de graphiques fastidieux, du stockage, des performances ou de la sécurité.

L'ensemble des solutions Embarcadero, en constante expansion, apporte désormais ce même sentiment de simplicité, de rapidité, de puissance et d'intuitivité à un nombre croissant d'outils de développement, de gestion et d'intégration pour toute sorte d'environnement IT. Homogène ou hétérogène. Débutant ou expert. PME ou grand groupe. Composez le +44 1628 684421, ou téléchargez une version d'évaluation gratuite de nos solutions primées... et retrouvez le plaisir de la gestion de base de données !

<http://www.embarcadero.com/prog.asp>



**EMBARCADERO
EUROPE**

Embarcadero Europe Thames House 17 Marlow Road Maidenhead Berkshire SL6 7AA
Tel: +44(0)1628 684400 Fax: +44(0)1628 684401 sales@embarcadero.co.uk
www.embarcadero.co.uk

©2002 EMBARCADERO. Le logo et les noms de produits et services d'EMBARCADERO TECHNOLOGIES sont des marques commerciales déposées par EMBARCADERO TECHNOLOGIES LTD. Tous les autres marques ou noms de produits appartiennent à leurs propriétaires respectifs.



Faut-il devenir indépendant ?

Choisir sa mission, choisir son entreprise, diversifier les projets sur lesquels on travaille, pouvoir s'organiser, réduire ou augmenter son temps de travail en fonction de ses envies et de ses besoins, ne plus supporter un supérieur : le statut d'indépendant se choisit pour de nombreuses raisons. Ci après quelques pistes pour devenir indépendant... mais pas inactif.

Pourquoi la période n'est pas si mauvaise ?

Crise économique, chômage en hausse, SSII qui licencient. Au premier abord, la période n'est pas propice à la création d'entreprise. Plutôt que de se morfondre en inter contrat dans une SSII, il est possible de négocier son départ et tenter l'expérience de l'indépendance. Depuis 2001, Une mesure facilite et sécurise le passage du statut de salarié à celui d'indépendant : tout salarié qui démissionne pour créer son entreprise est couvert par l'assurance chômage en cas d'échec durant trois ans.



Les compétences les plus recherchées



Les compétences les plus offertes

Source www.freelance.com au 7/04/03

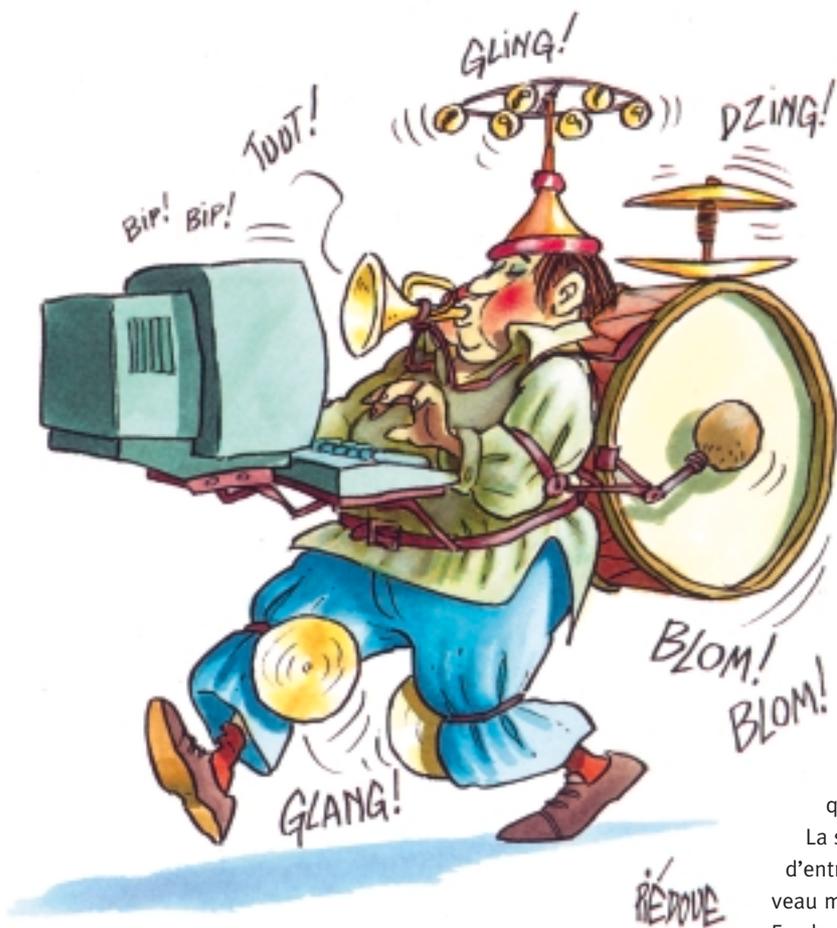
De plus, la nouvelle loi sur l'initiative économique, qui devrait être adoptée prochainement, va offrir de nouveaux avantages, et supprimer des inconvénients liés au statut de créateur d'entreprise, notamment la présomption d'indépendance.

Avant de se lancer...

Sylvain Vieujo, fondateur et dirigeant de freelance.com souligne qu'il est nécessaire de tester le marché. Il conseille, pour s'installer comme indépendant "d'avoir une vraie bonne expérience technique". Avoir travaillé en SSII et déjà si possible, sous forme de mission peut également être un atout. Autre conseil : être conscient de ses limites, ne pas s'afficher comme un expert dans tous les domaines.

Toutes les compétences ne se valent pas et ne sont pas également recherchées (voir tableaux 1 à 4) : bingo si vous êtes un pro des bases de données et de l'environnement Unix ou assimilés : respectivement : 19 et 20% des missions proposées sur ce site. Les langages sont moins demandés : 10% des offres de mission, dont 8% pour Java.

Si on rentre dans le détail des langages, ce ne sont ni les plus sexy ni les plus récents qui sont les plus demandés : l'offre de compétences semble trop importante sur C, C++, html, Cobol (oui, ça existe encore). Décalages très importants sur ASP (0,6% des offres de mission, 12,15% des compétences proposées), Linux (0,22% des missions, 10,53% des compétences), UML (4 fois plus



de propositions que de missions), VB (0,28% des missions, 13,77% des compétences proposées), Corba (0,28% des missions, 2,83% des compétences offertes) ou encore JSP (0,06% des missions, 0,4% des demandes). Si vous développez en html, sans vouloir vous décourager, passez votre chemin ou optez pour une formation complémentaire.

Le baromètre et les statistiques d'hitechpros font apparaître le client/serveur comme la compétence la plus demandée, et l'une des plus offertes par les indépendants. Également demandée : MVS et Unix, et tout ce qui a trait au décisionnel et bases de données (Oracle notamment)

Comment trouver des missions ?

Le plus difficile pour un indépendant en informatique est de trouver ses contrats et d'avoir les bons contacts pour cela. Surtout dans un contexte où les entreprises clientes ont tendance à réduire leur nombre de fournisseurs sous la pression de leurs directions achats. C'est l'occasion ou jamais de tester votre carnet d'adresses (Associations d'anciens élèves notamment). Selon l'étude précédemment citée, les indépendants trouvaient leur mission à 59% par hasard ou bouche à oreille, 55% par recommandation des clients et 17% par d'autres freelances. Cependant, des réseaux un peu plus formalisés se développent. Certains sites d'offres d'emploi ont des rubriques dédiées aux indépendants. Ci-après, quelques organismes, très différents dans leur approche, mais qui sont des pistes.

HitechPros est à l'origine une place de marché des inter-contracts en informatique, qui met en relation SSII et directions informatiques, nous avons eu l'occasion de vous en parler à plusieurs reprises dans Programmez !. Le site hitechpros.com a lancé depuis deux ans plusieurs services dédiés aux indépendants en informatique. Moyennant un abonnement (180 Euros HT pour un mois, 360 euros pour 3 mois, 480 euros pour un an), vous pouvez diffuser votre offre de services aux 2200 SSII et 1900 DSI inscrites sur le site et ce, anonymement. Elles vous contactent par mail, vous choisissez de rappeler celles qui vous intéressent. Vous recevez tous les jours un mail avec les missions proposées, environ 40 par jour, indique Emmanuel Heuze, en charge des freelances sur Hitechpros. La plupart des offres sont issues de SSII qui vous sous-traitent tout ou partie d'une mission. Avantages : aucune commission n'est prélevée par le site, les seuls frais sont ceux de l'abonnement. Inconvénient : mieux vaut savoir se vendre. Environ 1300 indépendants en informatique sont inscrits sur ce site (voir tableau).

HiTechPros fait également bénéficier gratuitement ses membres, de formations informatique à prix négociés (jusqu'à 60% de réduction) et de conseils juridiques et fiscaux. La société devient aussi client des Freelances abonnés. Chacun d'entre eux peut facturer 60 € HT au site lorsqu'il coopte un nouveau membre.

Freelance.com est une SSII virtuelle: des commerciaux, indépendants eux aussi cherchent des missions pour les indépendants chez des grands comptes. Les offres sont diffusées auprès des freelances inscrits, qui témoignent le cas échéant de leur intérêt. Le commercial sélectionne trois personnes correspondant aux critères du client, qui reste bien sûr décisionnaire. Il est possible que celui-ci négocie les tarifs, le commercial les renégocie le cas échéant avec les freelances. C'est la société qui facture le client et

Indépendants : portrait robot

Une étude, faite en 2001 par Ifop et Freelance.com montrait que les indépendants étaient des gens plutôt expérimentés. Selon cette enquête le profil type de l'indépendant était un homme (à 70%), diplômé (40% avaient un Bac+5, 35% entre Bac+2 et Bac+4), plutôt expérimenté (51% avaient entre 35 et 49 ans, 30% plus de 50 ans). Près de 80% d'entre eux étaient précédemment salariés. Le statut d'indépendant paraissait stable: 49% des personnes interrogées étaient indépendants depuis plus de 5 ans et 82% voulaient le rester dans les deux à venir. En 2001, 86% se déclaraient satisfaits de leur situation, le niveau de satisfaction étant toutefois plus important chez ceux qui venaient de se lancer.

Aujourd'hui sur Freelance.com, l'âge moyen des inscrits est de 37 ans. Seulement 4% ont moins de 25 ans, 17% entre 25 et 30 ans, 22% entre 30 et 35 ans, autant entre 35 et 40 ans, 16% entre 40 et 45 ans, et 10% entre 45 et 50 ans. Le prix moyen de la journée, 565 euros environ, témoigne de l'expérience de ces indépendants.



Catégories techniques	Répartition
Consulting-Expertise	25%
Client-serveur	14%
Nouvelles technologies	13%
Systèmes, réseaux, sécurité	12%
Mainframe	11%
CRM, Décisionnel, Datamining	7%
Moyens systèmes (minis)	7%
ERP	5%
Recette, support utilisateurs	2%
Info. indus., électronique, R&D	2%
Télécoms	2%

Les compétences des 1 300 indépendants inscrits sur hitechpros.com

vous facturez Freelance.com. La marge est d'environ 20%, répartie entre le commercial et la structure. Avantage : pas de problème de recouvrement de facture. Vous n'avez pas à négocier avec le client et vous pouvez vous concentrer sur votre mission. Inconvénient : le service a un prix. Freelance.com compte 30 000 inscrits à son site, mais tous ne sont pas encore indépendants. 300 à 400 personnes sont en mission actuellement, via cette structure. Autre société spécialisée, plus récente : Manaïpi, qui s'est fixée comme objectif de "fédérer les meilleures compétences IP du marché". Elle se positionne comme la "centrale" de sous-traitance spécialisée dans les infrastructures IP. La marge prélevée est de l'ordre de 20%.

Quel statut ?

Comme indépendant, la structure juridique peut être une EURL (Entreprise unipersonnelle à responsabilité limitée), une SARL, voire une Société Anonyme, mais c'est plus rare. Comme toute société, les formalités sont assez nombreuses (URSSAF, TVA, Complémentaire retraite, assurance maladie), même si elles devraient être simplifiées dans le futur, compte tenu de la nouvelle loi. Il faut tenir compte dans ses tarifs, des charges relativement lourdes. Et en cas d'échec, pas d'assurance chômage pour l'ex-indépendant.

Des structures existent pour éviter ces formalités et rester salarié : les sociétés de portage. Vous trouvez votre mission et ce sont elles qui se chargent de facturer le client et de vous reverser un salaire. Elles permettent également une domiciliation et offrent souvent des services divers (secrétariat, location d'une salle pour une réunion, etc.), pour un coût d'environ 12% du prix de la mission, qui peut apparaître assez modique au premier abord mais attention, elles ne vous reversent qu'un salaire charges salariales et patronales déduites : ce qui fait que par exemple pour 1000 euros de facturation, elles ne vous reverseront en moyenne que 500 euros (120 euros pour la société de portage et 380 euros de charges patronales et salariales). A utiliser avec modération donc, ou lors d'une installation après un travail salarié et si vous voulez conserver vos droits à indemnisation Assedic en cas d'échec.

Pour en savoir plus

www.freelance.com : explication du mode de fonctionnement, témoignages, références. Possibilité de s'inscrire sur une mailing list, pour recevoir des offres correspondant à votre profil.

http://index.freelance.com pour les compétences les plus recherchées, un baromètre actualisé en permanence.

www.hitechpros.com

http://indep.hitechpros.com. des témoignages de freelances satisfaits.

www.manaïpi.com : très orienté sur les spécialistes IP, mais possibilité de réaliser une simulation financière pour comparer statut de freelance et de salarié, un site clair et des pistes intéressantes

www.freelance-europe.com/fr. Le site n'est pas toujours bien actualisé, mais on y trouve notamment l'intégralité du texte de la nouvelle loi sur l'initiative économique (rubrique actualité). cette structure édite un annuaire professionnel des freelances et organise des manifestations sur leur statut.

www.canam.fr : le site de la caisse nationale d'assurance maladie des professions indépendantes. Calcul des cotisations, etc.

www.munci.org : ce mouvement pour une union nationale des consultants en informatique vise à fédérer les informaticiens, en vue de créer un ordre de la profession.

Devrait être réédité en juin le "Carnet pratique du Solo" de Michel Paysant. Editions groupe L'Expansion en partenariat avec Freelance en Europe. Il existe une Association des prestations de services informatique et pour les démarches liées à la création de votre entreprise : l'APCE (Association pour la création d'entreprise).

Une presse spécialisée existe, et vous donnera plein de conseils pour les démarches relatives à l'installation : Défis, l'Entreprise (une rubrique solo sur le site **www.lentreprise.com**).

La sérénité

Thierry Chombeau, 43 ans, père de trois enfants, s'est installé comme indépendant depuis deux ans. Après avoir travaillé onze ans dans une entreprise, où il est passé de programmeur à chef de projet, il a intégré une SSII durant 18 mois avant de décider de devenir indépendant. Thierry explique avoir eu l'impression de perdre contact avec l'évolution des technologies en restant confiné dans un même système d'information. Impression qui rend logique son intégration dans une SSII, où il s'est vite aperçu qu'il n'aurait pas non plus la possibilité de changer d'horizon technologique. Il a donc décidé de devenir indépendant et s'est inscrit sur Freelance.com. Une première mission lui a été proposée chez Auchan, et il a depuis lors enchaîné les missions de longue durée chez ce même client, comme chef de projet, en environnement As/400. Il estime avoir eu des missions intéressantes et diversifiées au sein d'une grande entreprise, et apprécie de pouvoir s'organiser comme il le souhaite.

Selon Thierry, le fait d'avoir eu recours à freelance.com pour trouver ses missions lui apporte une certaine sérénité, du fait de ne pas avoir à se partager entre son travail et des démarches de prospection. A échéance d'une mission, le commercial avec qui il travaille habituellement se charge de négocier durée et tarif du nouveau contrat, en relation avec Thierry, ce qu'il estime justifié, dans la mesure où il les rémunère depuis deux ans. A l'avenir, il envisage de se former aux technologies réseau, l'As/400 étant un peu sur le déclin.



ANALYSE ET CONCEPTION

LES DIAGRAMMES UML

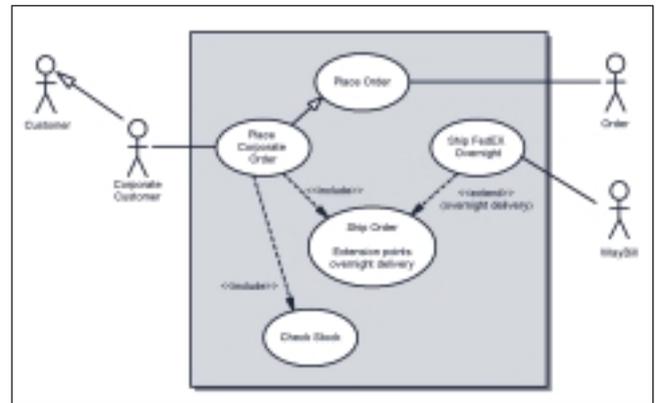
Méthode

(suite)

Dans le précédent numéro, nous avons étudié 5 diagrammes UML : les diagrammes de cas d'utilisation, de séquence, d'activités, de classes et de composants. Ces diagrammes aident à comprendre les exigences du système, à en concevoir sa structure et son comportement attendu et à transcrire l'ensemble en un code. Nous étudions aujourd'hui quatre autres diagrammes.

DIAGRAMME D'OBJETS

Tel le diagramme de classes, le diagramme d'objets est un diagramme de structure statique UML ; il définit les éléments physiques du système à un instant T et ne tient pas compte de l'activité dynamique du système. Il établit une correspondance 1:1 avec le code, mais à la différence du diagramme de classes, nous utilisons des instances spécifiques d'un classifieur et non la définition de ce dernier. La meilleure description d'un diagramme d'objets est de le présenter comme un diagramme d'instances de classes.



> Figure 1:

Le diagramme d'objets sert essentiellement à l'analyse. Certaines contraintes entre classes ne peuvent pas être représentées dans un diagramme de classes. Nous utiliserons donc le diagramme d'objets pour les documenter. Il nous permet par ailleurs de définir des scénarios spécifiques de type " que se passerait-il si ? " s'intéressant à des exemples spécifiques des instances de classes gérées, afin de clarifier l'interaction entre ces éléments.

Pour ceux qui ne sont pas familiarisés avec la modélisation orientée objet, les classifieurs sont la définition abstraite de la structure d'un objet. Ils nous indiquent quel type de données est géré (les attributs/membres représentent des éléments de données) et quelles sont les capacités du classifieur (les opérations/méthodes représentent le comportement de l'objet).

Les instances sont un exemple spécifique d'un classifieur. Une classe appelée "Customer" est définie pour avoir un attribut "Name".

L'instance "Jane Doe" de la classe "Customer" est le client dont le nom est exactement "Jane Doe". Les instances ont généralement plus de sens que les classifieurs car ces derniers représentent uniquement un niveau d'abstraction. Rassembler plusieurs instances, ou exemples, d'un classifieur peut vous aider à en comprendre l'objectif et l'utilisation.

Le diagramme d'objets est une forme spécifique du diagramme de classes, donnant des exemples d'instances de classes et révélant

UML

NIVEAU : INTERMÉDIAIRE

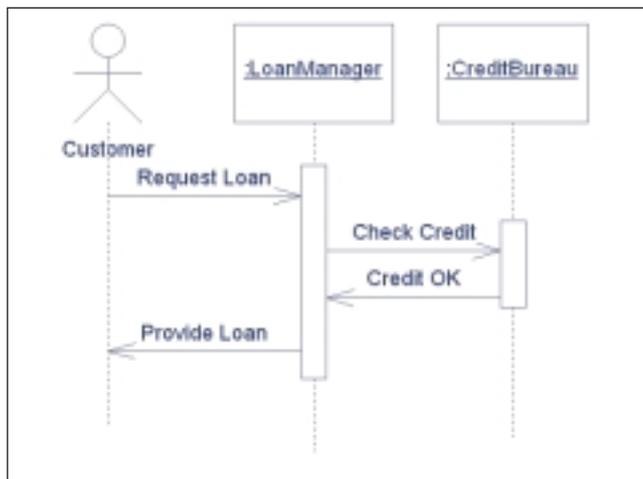
les principales valeurs et relations. Ainsi, le client dont l'identifiant est exactement 52271, et dont le nom est exactement " John Doe " est une instance de la classe " CustomerBean ". Cette instance client est liée à trois instances de commande (trois commandes) dont les numéros de commande sont 122047, 122103, et 122399. (figure 1)

DIAGRAMME DE COLLABORATION

Le diagramme de collaboration est cousin du diagramme de séquence. En fait, un diagramme de séquence et un diagramme de collaboration peuvent effectivement être utilisés à la place l'un de l'autre et être facilement convertis l'un en l'autre. La différence entre ces deux types de diagramme réside dans la manière dont l'utilisateur les lit et les comprend.

Les diagrammes de séquence sont très hiérarchisés et suivent l'axe temporel. Les diagrammes de collaboration sont plus axés sur la structure de l'objet. L'ordre des messages est représenté en numérotant ces derniers dans le diagramme. De cette manière, la synchronisation est conservée, même si l'organisation du diagramme n'est pas temporelle.

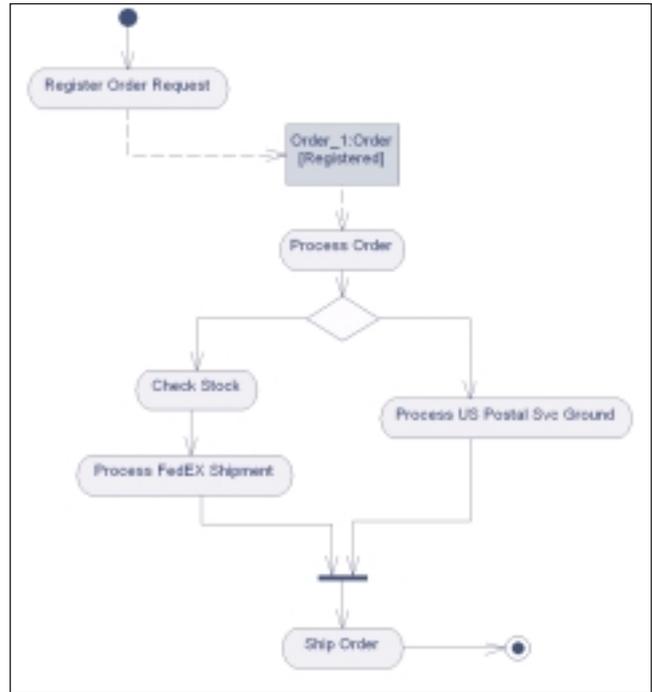
Un diagramme de collaboration représente la dynamique d'un système, son comportement au fil du temps, en termes d'interactions



> Figure 2:

entre les éléments, ou objets, du système. Il complète les diagrammes de classes et d'objets en représentant la structure statique du système, mais ce sont des " messages " d'interaction qui sont échangés entre les objets du système et non des relations basées sur la structure.

La construction d'un diagramme de collaboration peut également donner l'occasion de tester le modèle statique au niveau conceptuel. Des instances de classe sont définies dans le diagramme de classes. Leur interaction définit un scénario d'utilisation spécifique et les communications internes qui ont lieu entre ces éléments. Il



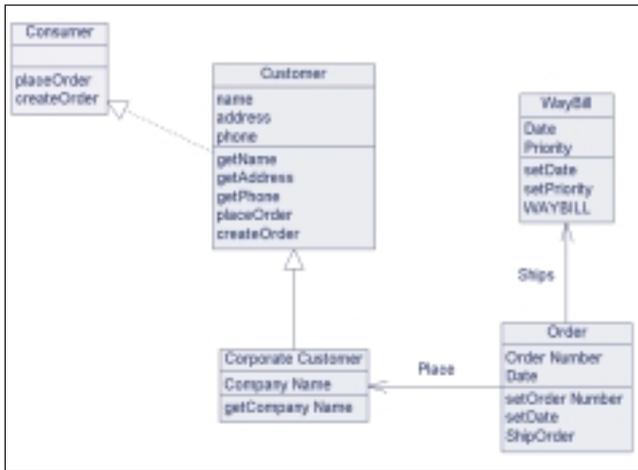
> Figure 3:

est également possible d'utiliser des acteurs pour représenter les influenceurs externes ainsi que les utilisateurs externes du système (comme un diagramme de cas d'utilisation).

A titre d'exemple, nous allons utiliser un système de saisie de commandes utilisé par des clients (Customers) et des commerciaux (Sales Representatives). Le client interagit avec le système en créant une nouvelle commande. L'objet commande dialogue avec l'objet ventes, et ce dialogue est représenté par des messages de la liaison – dans notre cas, il n'y en a que deux, une requête de commande de la part de la classe Orders, et une confirmation de la commande de la part de la classe Sales. Le nombre de messages que vous pouvez avoir sur une liaison est illimité. Notre dialogue commence donc par une requête de commande puis se poursuit avec la confirmation de cette commande. (figure 2)

APPLICATION

Les diagrammes de collaboration présentent une importance toute particulière pour les concepteurs car ils clarifient les rôles des objets. Vous pouvez construire un diagramme de collaboration avant un diagramme de séquence (si vous prévoyez de construire les deux), mais ces diagrammes sont habituellement générés après le diagramme de classes, afin d'illustrer les interactions des objets dérivés des classes. Vous pouvez utiliser un ou plusieurs diagrammes de collaboration pour représenter un cas d'utilisation ou pour décomposer un comportement complexe en sous-divisions logiques.



> Figure 4:

DIAGRAMME D'ETATS

Le diagramme d'états (également appelé "machine d'états"), décrit le comportement d'une classe ou d'un composant particulier en suivant son évolution au fil du temps. Ce diagramme montre comment un objet passe d'un état à un autre et quelles opérations sont applicables à chaque état. Ainsi, une commande qui vient d'être créée est dans son état initial. Après que le client ait confirmé que la commande est correcte, un état confirmé est saisi. A l'expédition de la commande, l'état doit passer de "confirmé" à "expédié".(figure 3)

A chaque fois qu'une classe présente différentes options disponibles (différents comportements valides) en fonction de l'étape du cycle de vie, vous utiliserez un diagramme d'états pour modéliser ces règles et conditions. Chaque étape du cycle de vie est un état pour l'objet, et chaque déclencheur de changement d'état représente une transition d'un état à un autre. Vous pouvez passer à autant d'états différents que vous le souhaitez, ou vous pouvez entrer un état à partir de nombreuses transitions différentes (à partir de nombreux autres états).

DIAGRAMME DE SOUS-ETATS

Pour garder à un diagramme d'états sa simplicité et sa lisibilité, vous pouvez trouver qu'un ou plusieurs états définis impliquent en fait un comportement beaucoup plus complexe, qui pourrait lui-même être défini dans un diagramme d'états. Au lieu de rendre le diagramme principal encore plus complexe et de lui ajouter ces détails supplémentaires, vous pouvez décomposer cet état en un sous-état, c'est-à-dire générer un deuxième diagramme d'états qui définira le comportement interne plus complexe de l'état parent. (figure 4)

DIAGRAMME DE DEPLOIEMENT

Le diagramme de déploiement permet de déterminer l'emplacement futur de tous les éléments du code, dans le serveur/poste de travail/base de données. Des noeuds sont rattachés aux logiciels/hardware qui exécuteront certaines parties de la logique applicative. Ces noeuds interagissent pour montrer comment les différents ordinateurs et systèmes développés interagissent et

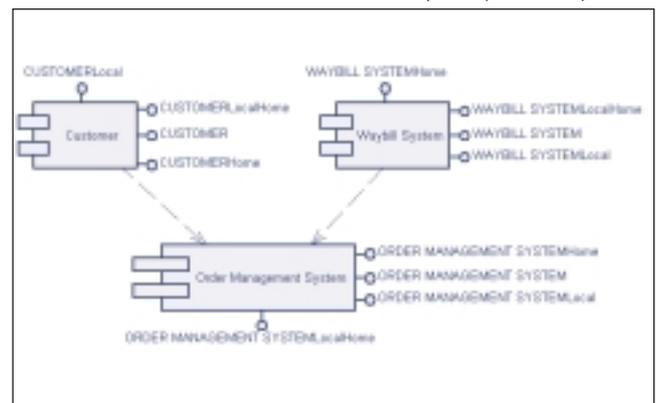
s'intègrent les uns aux autres. Les nœuds contiennent des instances de composants qui seront déployées dans les serveurs de base de données, d'applications ou Web.

Les diagrammes de déploiement sont utilisés pour le déploiement réel des composants sur les serveurs. En définissant l'emplacement où les composants sont censés s'exécuter, il vous est facile de mettre en correspondance, de déployer et de gérer la logique applicative distribuée entre les applications côté client, les composants des applications côté serveur ou la logique serveur côté base de données. Il s'agit d'un modèle 1:1 de l'architecture physique gérée.

Par exemple, vous avez décidé de mettre en oeuvre deux Enterprise Java Beans, et de les exécuter sur un serveur d'applications. Le diagramme suivant montre un noeud unique, et deux composants, chacun correspondant à un EJB, au sein de ce noeud. Le EmployeeBean dépend du CustomerBean au sein du même serveur d'applications. (figure 5)

CONCLUSION

Vous étant familiarisé à la modélisation UML de base avec les diagrammes de cas d'utilisation, de séquence, d'activités, de classes et de composants, il vous faut des outils supplémentaires pour définir les détails de certains éléments spécifiques du système.



> Figure 5:

Vous pouvez avoir envie d'exprimer la structure d'un objet en utilisant des exemples exacts, dans le diagramme d'objets. Vous souhaitez en savoir plus sur le comportement d'une classe présentant des états multiples et complexes, grâce à un diagramme d'états. Vous avez besoin de voir les interactions entre les composants de votre système, non en termes de temps mais en termes de structure, en utilisant un diagramme de collaboration. Enfin, il vous faut mieux comprendre comment votre architecture distribuée peut être exploitée au mieux en montrant l'emplacement de tous les composants de votre système dans le hardware ou les serveurs physiques de votre environnement, grâce au diagramme de déploiement.

UML vous offre quatre diagrammes supplémentaires qui complètent l'analyse technique, la conception et la mise en oeuvre de votre logique applicative. Ces 9 diagrammes, associés à vos modèles Merise de données classiques et à une nouvelle modélisation des processus métiers, offrent la gamme complète de tous les éléments qui pilotent le développement de votre logiciel, depuis les spécifications de haut niveau, jusqu'aux spécifications techniques et aux spécifications de données et à la mise en oeuvre physique.

David DICHMAN

Directeur Produit, Power AMC, SYBASE Inc

UML

NIVEAU : INTERMÉDIAIRE

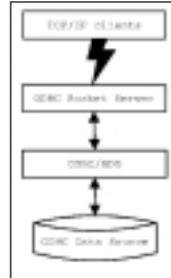
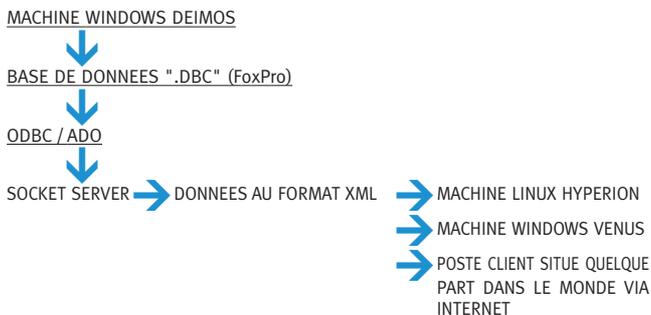
Jetons un pont ODBC entre nos données

Comment interroger une base de données localisée sur un serveur Windows alors que votre client se trouve sur un poste Windows ou Linux distant ?

Ne pensez pas que la migration des données est un problème facile à résoudre. Si par exemple vous disposez à la fois d'un moteur MySQL sous Windows et sous Linux, vous ne pourrez pas pour autant copier directement les fichiers de données de la première base sur l'autre machine. Pourquoi ? Simple-ment parce que le système de gestion des fichiers est différent : par exemple NTFS sous XP et ext3 sous Linux. En outre se pose-rait la question de la synchronisation des bases (ce qui entre pa-renthèse est réalisable, via un mécanisme de gestion des fichiers journaux binaires, à partir de la version 3.23.20 de MySQL).

Le scénario

Posons le décor. Vous disposez d'une base de données Visual Fox-Pro sur une machine Windows du nom de DEIMOS. Vous voulez interroger celle-ci depuis une autre machine Windows (VENUS) et depuis un serveur Linux (HYPERION). En fait, peu importe la source de données, du moment qu'elle est accessible par le mécanisme ODBC. Il pourrait donc s'agir d'une base de données ACCESS ou encore SQL SERVER. D'autre part, sur le serveur Windows VENUS tourne le serveur Web HTTP APACHE. D'une machine dis-tante sur Internet vous désirez également interroger votre base de données.



La solution idéale est d'employer un pont ODBC. La machine Windows VENUS pourra interroger la base de données située sur DEIMOS, à l'aide d'un client PYTHON (ou PERL ou C++). Ce même code sera utilisé par un client Python sous Linux (machine HYPERION). Le poste distant situé quelque part dans le monde, interrogera le serveur http de VENUS. L'interpréteur PHP installé sur le poste du client interrogera à son tour la base de données, en passant également par le pont ODBC.

Le principe de ce pont est simple : le client interroge un service réseau qui envoie les résultats des requêtes ODBC sous la forme de données XML.

Pour réaliser un tel pont ODBC, des solutions commerciales existent, comme celle proposée par EASYSOFT (<http://www.easysoft.com>). Souvent, on vous laisse même la possibilité de télécharger une version limitée.

Cependant, une solution libre, sans limitation existe. Elle se nomme "ODBC SOCKET SERVER" et c'est celle-ci que nous vous proposons de découvrir.

ODBC SOCKET SERVER

La solution que nous allons vous décrire a été testée en situation réelle. Pour pouvoir tester le pont ODBC sur votre installation vous devez disposer d'au moins deux machines reliées en réseau. Chaque machine doit posséder une pile TCP/IP fonctionnelle et vous pouvez atteindre l'une et l'autre par le biais de la commande "ping".

Il n'y aura pas de problème de résolution de noms, à condition de spécifier l'adresse IP et non le nom de la machine dans vos codes sources.

ODBC SOCKET SERVER est un projet sous licence GPL (<http://odbc-sock.sourceforge.net/>). Vous pouvez récupérer l'archive ZIP en cliquant sur le lien "The ODBC Socket Server installation". Le projet a été initialement développé pour tourner sous Windows 98, NT ou Windows 2000. Attention : les binaires sont différents pour Windows 98 ou NT/2000, car les services ne sont pas gérés de la même manière.

Nous avons testé ODBC SOCKET SERVER en l'installant sur la machine DEIMOS, qui est un poste Windows XP. Le poste client Windows VENUS tourne sous Windows 2000. Le poste client HYPERION est une machine Linux REDHAT avec noyau 2.4.18-5.

Les étapes de l'installation

1) Décompressez l'archive ZIP. Copiez le fichier ODBCsocketServer.exe sous l'arborescence "C:/Windows/system32". Exécutez ODBCsocketServer.exe /service sous une boîte DOS. Le service est maintenant installé. En lançant "outils d'administration / services" vous trouverez dans la liste des services l'entrée "ODBC

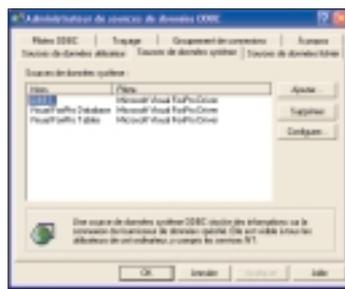
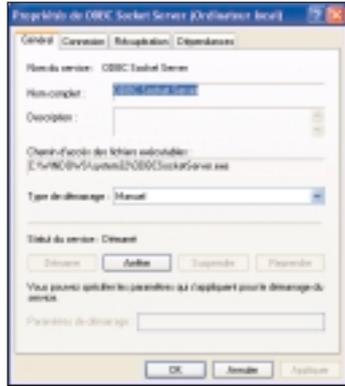
ODBC

NIVEAU : DÉBUTANT
ET INTERMÉDIAIRE

SOCKET SERVER". Démarrez-le. Vous pouvez indiquer "type de démarrage : automatique" pour que celui-ci s'exécute au prochain démarrage de la machine. Sinon laissez-le sur "manuel".

2) Ne copiez pas le fichier ATL.DLL dans le répertoire "C:/Windows/system32". Exécutez le fichier "Registry.reg".

3) Lancez "outils d'administration / administration de sources de données ODBC". Et ajoutez une "source de données système" (onglet "source de données systèmes"). Nous avons choisi le nom "GIBES" qui pointe sur une base de données VisualFoxPro "bays.dbc". Attention : la base de données (ou bien la table dbf, ou encore la table access) sera située sur la machine DEIMOS (machine sur laquelle tourne le service ODBC SOCKET SERVER).



Le test sur la machine locale

L'archive ZIP contient un client "COM". Vous devez commencer par double cliquer sur la bibliothèque "ODBCClient.dll" afin de l'enregistrer dans la base de données des registres (exécution de la commande regsvr32.exe). Ou bien exécutez directement la commande `regsvr32.exe /votrechemin/ODBCClient.dll` en ouvrant une boîte DOS.

Exécutez maintenant le fichier `socktest.exe`. Vous devez indiquer l'adresse IP de votre machine locale (ici 192.168.0.24). Comme déjà expliqué nous ne vous conseillons pas de mettre à ce niveau un nom de machine, ce qui vous évitera une erreur de résolution de nom (si vos machines sont mal configurées). Le port par défaut est ici "9628". La chaîne ODBC "connection String" est simplement "DSN=GIBES". Le "SELECT" est "select * from iso3166". Évidemment vous devez en principe indiquer une requête valide pour votre base de données. En cliquant sur le bouton de commande "ExecSQL", le client COM interroge le pont ODBC qui à son tour renvoie les données au format XML.

Vous devez obtenir quelque chose de ce style :

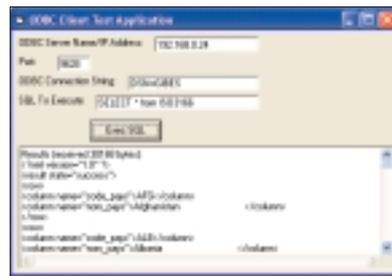
```
Results (received 30168 bytes):
<?xml version="1.0" ?>
<result state="success">
```

```
<row>
<column name="code_pays">AFG</column>
<column name="nom_pays">Afghanistan
</column>
</row>
...
```

Si vous obtenez une erreur `<result state="failure">`, et que la description n'est pas assez explicite (exemple : "Source de données introuvable"), vous pouvez ouvrir "l'observateur d'événements". En effet la base de données des registres contient l'entrée HKEY_LOCAL_MACHINE/SOFTWARE/ODBCSOCKETSERVER/DEBUGLEVEL qui en principe est à "1" par défaut (déboguage maximal).

Le test depuis la machine distante Windows 2000

Vous pouvez à nouveau commencer par exécuter `socktest.exe` (tout en ayant enregistré la bibliothèque "ODBCClient.dll"). En théorie cela doit fonctionner du premier coup. Si ce n'est pas le cas, vous avez un problème au niveau de votre réseau. Une source d'erreur possible est de constater que le port par défaut du service de la machine distante (ici DEIMOS) n'est pas joignable. Ce sera le cas, si un pare-feu empêche le dialogue. Vérifiez donc que le port 9628 est bien ouvert.



Le test depuis la machine distante Linux

Sous Linux vous devez récupérer le client Python. L'interpréteur Python doit évidemment être présent (en l'occurrence la version Python 1.5.2 mais nous vous conseillons une version plus récente). Les fichiers concernés sont "odbc.py" ("test.py" dans l'archive) et "odbcsocksvr.py".

Editez le fichier `odbc.py` :

```
#!/usr/local/bin/python
import odbcsocksvr

host = '192.168.0.24'
conn_string = 'DSN=GIBES;'
sql = 'select * from iso3166'

svr = odbcsocksvr.ODBCSocketServer(host,conn_string)
results = svr.ExecSQL(sql)
print results
```



L'exécution fonctionnera à condition, répétons-le, que le port 9628 de la machine distante soit accessible. Sinon vous obtiendrez :

```
?xml version="1.0"?>
<result status="failure">
<error>Could not connect to host</error>
</result>
Traceback (most recent call last):
File "odbc.py", line 9, in ?
```

ODBC

NIVEAU : DÉBUTANT ET INTERMÉDIAIRE

ASP.NET : L'AUTHENTIFICATION

par MICROSOFT PASSPORT

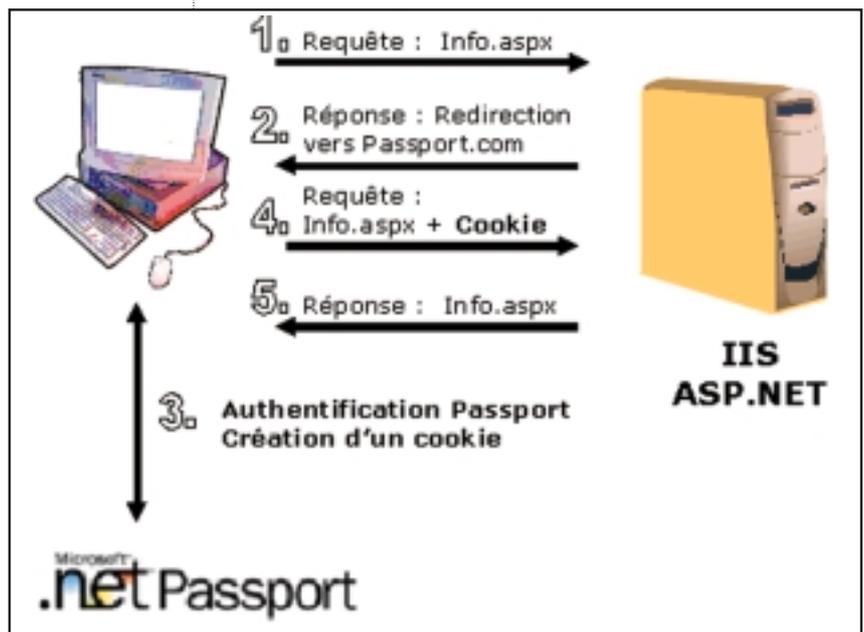
Après avoir traité l'authentification par formulaire, nous allons aborder l'authentification par Microsoft Passport. Microsoft Passport a été conçu à l'origine, comme le système d'authentification de la messagerie Hotmail. Il existe depuis 1999 plus de 210 millions d'utilisateurs détenant un compte Passport. Microsoft Passport permet à un site Web de gérer ses utilisateurs et ses profils de manière centralisée, sur un serveur hébergé par Microsoft, et offre aux internautes un système de signature unique sur le Web.

Plus besoin de multiplier les comptes et les mots de passe pour chaque site et de s'authentifier systématiquement. L'utilisateur s'authentifie une seule fois et peut directement accéder aux sites partenaires Passport.

Nous allons aborder le fonctionnement de .NET Passport Single Sign In. Puis nous expliquerons comment paramétrer .NET Passport sur notre environnement de développement, enfin nous implémenterons cette technologie sous ASP . Net.

Regardons plus précisément le fonctionnement de .NET Passport Single Sign In.

Le mécanisme d'authentification Microsoft Passport



Fonctionnement de l'authentification Passport

Voici les différentes étapes d'une authentification d'un utilisateur par Microsoft Passport :

1. Un client soumet une requête http pour accéder à une page ASP sécurisée. ASP .Net vérifie sur la machine de l'utilisateur que celui-ci détient un cookie spécifique à Passport.
2. Si le cookie n'est pas trouvé, l'utilisateur est redirigé vers le site Passport.com pour qu'il s'authentifie.
3. Une fois la signature de l'utilisateur saisie et celui-ci reconnu par Passport, le service Passport authentifie le client, et lui retourne un cookie auquel est associé un ticket d'authentification.
4. Le client est ensuite redirigé vers la page Web initialement demandée mais cette fois avec le ticket d'authentification.
5. La page est retournée au client.

L'authentification Passport se sert donc de la **redirection http** pour renvoyer l'utilisateur vers le site Microsoft Passport ou vers la page Web demandée. Les informations d'authentification sont transmises aux sites partenaires, via un **cookie crypté en triple DES** avec une clé à 168 bits, différente pour chaque site Web partenaire. D'autres cookies sont utilisés pour stocker le profil (nom, adresse, numéro de téléphone ...) et la liste des sites partenaires visités sur lesquels l'utilisateur peut retourner sans se réauthentifier. Tous ces cookies sont supprimés lors de la déconnexion de l'utilisateur.

Installation et configuration de l'environnement de développement

Il est nécessaire de réaliser les étapes suivantes pour mettre en place un environnement de pré-production utilisant Microsoft Passport.

■ Création d'un compte Passport

Vous devez d'abord disposer d'un compte Passport qui peut être créé à l'adresse suivante :

<http://www.passport.net/Consumer/default.asp>

Le service vous propose de saisir un formulaire d'inscription à Microsoft Passport et de fournir une adresse de messagerie et un mot de passe.

■ Installation du SDK Microsoft Passport

Le kit de développement Microsoft Passport est téléchargeable à l'adresse suivante :

<http://msdn.microsoft.com/downloads/default.asp?URL=/downloads/sample.asp?url=/msdn-files/027/001/885/msdncompositedoc.xml>

La version actuellement disponible est la version 2.5 qui nécessite Windows 2000 Server, Windows XP Professional ou Windows Server 2003 pour fonctionner.

Une fois le SDK installé, vous disposez de l'outil d'administration " Passport Manager Administration " vous permettant de configurer votre serveur de production afin qu'il soit reconnu par Microsoft Passport. Le SDK va également installer coté serveur le composant " Passport Manager " qui propose une API. Celle-ci permet de mettre en œuvre le mode Single Sign In et de récupérer des informations sur l'utilisateur authentifié.

■ Création d'un site Passport

Vous devez ensuite créer un site Passport à l'adresse suivante :

<https://www.netservicesmanager.com>

Nous allons créer une application de test implémentant le mode SSI.

Sélectionner " Create a new application " de type ".NET Passport Single Sign-In " puis renseignez les caractéristiques suivantes de votre application.

Liste des champs obligatoires à la création d'un site Web Passport :

Champ	Description	Exemple de valeur
Web Site Title	Nom du site	Programmez
Domain Name	Le nom de domaine de plus haut niveau du site	programmez.com
Default Return URL	Quand vous vous signez auprès de .Net Passport la requête s'accompagne d'une URL de retour. Si cette URL de retour n'existe pas alors l'URL définie dans " Default Return URL " est appelée.	http://www.programmez.com/Return.aspx
Privacy Policy URL	Ce lien doit pointer vers un document légal qui informe l'utilisateur sur la protection des informations personnelles au sein de .NET Passport.	http://www.passport.com/Consumer/PrivacyPolicy.asp
Co-brand Image URL	Image apparaissant au dessus du login Passport.NET lors de la connexion à votre site. L'image est de 468X60 pixels.	http://www.programmez.com/images/PassportCoBrand.gif
Co-brand Image2 URL	Image apparaissant sur la page de login Passport.NET lors de la connexion à votre site. L'image est de 102X80 pixels.	http://www.programmez.com /images/PassportCoBrand2.gif
Co-brand Instruction Text	Texte apparaissant sur la page de Login .NET Passport	Veuillez vous signer pour accéder à notre site.
Expire Cookie URL	C'est l'URL chargée de détruire tous les cookies .NET passport lorsqu'un utilisateur se déconnecte.	http://www.programmez.com /killcookie.aspx

Une fois le site créé, vous obtenez un site ID qui vous servira par la suite pour paramétrer votre environnement de développement avec Passport Administration Utility.

■ Enregistrement de la clé d'application

Vous devez maintenant installer votre clé d'application. Tout d'abord téléchargez la clé en cliquant sur l'option " Download a key " de la page " Manage My applications ". Vous obtenez un fichier du style PartnerXXXXXX_1.exe.

Il faut alors installer la clé à l'aide de la commande : PartnerXXXXXX_1.exe /addkey.

Puis la définir comme valeur courante : PartnerXXXXXX_1.exe /makecurrent /t o.

Votre clé d'application est désormais installée.

■ Configuration de l'application locale

Lancez l'utilitaire " Passport Administration Utility " installé avec le SDK Microsoft Passport. Puis configurez le en remplaçant le SiteID qui vaut actuellement 1, par la valeur que vous avez obtenue lors de l'enregistrement de votre application sur le site Passport.

Nous cloturons ainsi la configuration de notre environnement et pouvons passer à la partie implémentation du mode Passport Single Sign In sous ASP .Net.

Implémentation de .Net Passport SSI

■ Configuration du fichier Web. Config

Avant de pouvoir utiliser les fonctionnalités offertes par Microsoft Passport, il est nécessaire de configurer votre application Web pour réaliser une authentification Passport. Pour cela, il faut modifier l'élément " authentication " de la manière suivante :

```
<authentication mode="Passport">
</authentication >
```

■ L'objet PassportIdentity

La classe **PassportIdentity** de l'espace de noms System.Web.Security expose l'ensemble des méthodes et attributs qui seront utilisés dans le code pour authentifier l'utilisateur.

Les attributs **MemberIDHigh** et **MemberIDLow** sont deux attributs de 32 bits qui correspondent à un identifiant passport unique (**Passport Unique ID**) pour chaque utilisateur. Ces attributs sont également représentés sous la forme d'une chaîne de 16 caractères à travers la propriété **HexPUID**. Ce PUID étant unique, il peut être utilisé et stocké dans une base de données clients.

Nous allons utiliser plusieurs méthodes de l'objet PassportIdentity, pour développer notre page d'authentification, commençons par en décrire les principales.

GetIsAuthenticated :

Cette méthode détermine si l'utilisateur courant est authentifié ou non.

```
public bool GetIsAuthenticated(
    int iTimeWindow, // Durée effective de l'authentification
    bool bForceLogin, // Force les utilisateurs à se signer durant iTimeWindow
    bool bUseSecureAuth // Déclare le niveau de sécurité (false : 0, true : 10)
)
```

GetFromNetWorkServer :

Cette méthode détermine si une connexion provient d'un serveur Passport .Net ou non.

```
public bool GetFromNetWorkServer ()
```

LogoTag2 :

Cette méthode permet de générer un fragment de code HTML qui affiche un bouton de connexion (si l'utilisateur n'est pas déjà signé) ou de déconnexion (si l'utilisateur est signé).

```
public string LogoTag2(
    string strReturnUrl, // URL de retour une fois authentifié ou déconnecté
    int iTimeWindow, // Durée effective de l'authentification
    bool fForceLogin, // Détermine l'utilisation du paramètre iTimeWindow
    string strCobrandedArgs, // Spécifie les champs à ajouter à l'URL de retour
    int iLangID, // Définit le langage utilisé sur le serveur Passport (FR = 1036)
    bool fSecure, // Déclare si la méthode est appelée par une page HTTPS
    string strNameSpace, // Domaine de redirection pour créer un compte
    // Passport pour un utilisateur qui n'en possède pas.
    int iKPP, // Concerne le règlement COPPA valable aux états-unis
    bool bUseSecureAuth // Déclare le niveau de sécurité (false : 0, true : 10)
)
```



Les boutons affichés par la méthode LogoTag2.

Maintenant que nous connaissons ces méthodes, utilisons les dans notre page chargée de vérifier l'accès à notre application Web.

```
<%@ Language=C# %>
<%
    PassportIdentity oMgr;
    oMgr = (PassportIdentity)Context.User.Identity;

    string thisURL, logoutURL;
    bool bAuthentifie;

    // L'URL de la page courante
    thisURL = "http://" + Request.ServerVariables.Get("SERVER_NAME") + Request.ServerVariables.Get("SCRIPT_NAME");

    // L'URL de la page de déconnexion
    logoutURL = "http://" + Request.ServerVariables.Get("SERVER_NAME") + "/logoutuser.htm";

    // Nettoyage de la QueryString
    if (oMgr.GetFromNetworkServer)
    {
        Response.Redirect(thisURL);
    }

    // Si l'utilisateur est authentifié depuis moins d'une heure
    if (oMgr.GetIsAuthenticated(3600,false,false))
    {

        //Récupération du PUID
        string nickname, memberidhigh, memberidlow;
        memberidhigh = oMgr.GetProfileObject("MemberIDHigh").ToString();
```

ASP

NIVEAU : DÉBUTANT

```

memberidlow = oMgr.GetProfileObject("MemberIDLow").ToString();

// Recherche de l'utilisateur dans une base des clients
// La méthode UtilisateurEnBase doit être développée par le site
if (UtilisateurEnBase(memberidhigh,memberidlow))
{
    bAuthentifie = true;
}
else
{
    // La méthode AjouterPUIDEnBase doit être
    développée par le site
    AjouterPUIDEnBase(memberidhigh,memberidlow);
    bAuthentifie = true;
}
}
else bAuthentifie = false;

if (bAuthentifie)
{
    // L'utilisateur est signé et le bouton déconnexion est affiché
    Response.Write(oMgr.LogoTag2(logoutURL,3600,false,null,
-1,false,null,-1,false));
    Response.Write("<HR>");
    Response.Write("Vous êtes un utilisateur .NET Passport !");
}
else
{
    // L'utilisateur n'est pas signé, un bouton de connexion est affiché
    Response.Write(oMgr.LogoTag2(thisURL,3600,false,null,
-1,false,null,-1,false));
    Response.Write("<HR>");
    Response.Write("Vous n'êtes pas un utilisateur .NET Passport");
}

%>

```

Suppression des cookies d'authentification

Lorsqu'un utilisateur clique sur le bouton de déconnexion d'un site .NET passport, un script est exécuté et celui-ci récupère la liste des sites visités par l'utilisateur (Visited Sites cookie). Ensuite, ce script efface tous les cookies du domaine *.passport.com, puis appelle pour chaque site visité le script chargé d'effacer les cookies d'authentification. Enfin, une fois tous les scripts exécutés, une page est retournée à l'utilisateur pour l'informer du bon déroulement de la déconnexion.

Les cookies sont utilisés pour stocker de manière persistante les informations d'authentification et le profil de l'utilisateur habilité.

Chaque développeur d'un site .Net Passport doit produire un script de suppression des cookies, qui est d'ailleurs indiqué dans le champ **Expire Cookie URL**, lors de la création du site Passport.

Voici un exemple de script de suppression des cookies.

```

<%@ Page Language="C#"%>
<%@ Import Namespace="System.Web.Security"%>

<%
Response.ContentType="image/gif"
Response.Expires = -1;
Response.AddHeader("P3P", "CP=""TST""");

// Suppression des cookies utilisés par .Net Passport
Response.Cookies("MSPProf").Value = "";
Response.Cookies("MSPProf").Expires = DateTime.MinValue;

Response.Cookies("MSPAuth").Value = "";
Response.Cookies("MSPAuth").Expires = DateTime.MinValue;

Response.Cookies("MSPConsent").Value = "";
Response.Cookies("MSPConsent").Expires = DateTime.MinValue;

Response.Cookies("MSPSecAuth").Value = "";
Response.Cookies("MSPSecAuth").Expires = DateTime.MinValue;

// Suppression des cookies de l'application
Response.Cookies("MonCookie").Value = "";
Response.Cookies("MonCookie").Expires = DateTime.MinValue;

// Affichage du bouton de confirmation du bon déroulement de la déconnexion
System.Web.Security.PassportIdentity.SignOut("images/signout_good.gif");

%>

```

Conclusion

Comme nous l'avons vu, l'implémentation de Microsoft Passport une fois l'environnement paramétré, se réalise en quelques lignes. L'avantage principal d'une telle solution est la possibilité de se signer sur plusieurs sites sans devoir s'authentifier systématiquement. L'inconvénient est la gestion des habilitations par un tiers, ce qui peut poser des problèmes de confidentialité, de sécurité pour certaines entreprises.

Nous achevons ainsi notre série d'articles sur les différents modes d'authentification d'une application ASP .Net en espérant que vous aurez désormais les éléments pour choisir la solution d'authentification adaptée à vos besoins. ■

Christian Peyrusse

DEMYSTIFIER J2EE AVEC JBoss

Kit de survie du programmeur de JSP (2ème partie)

Pour ceux qui ne connaissent pas la programmation des JSP, je vais tenter dans les lignes qui suivent de leur fournir les premières notions indispensables, pour écrire des pages un tant soit peu fonctionnelles.

Une page JSP est un texte, à l'instar d'un fichier HTML. D'ailleurs tout fichier HTML peut devenir un fichier JSP valide. Un texte JSP permet de mixer HTML et code Java. Le code Java doit être compris entre les balises "`<% "` et "`%> "` ou "`<%= "` et "`%> "`.

La première paire de balises permet d'insérer une ou plusieurs déclarations ou instructions Java. Ces instructions sont exécutées sans incidence directe sur le code HTML généré. Voici un exemple qui affiche 10 fois HELLO :

```
<% for (int i=1; i<10; i++) { %>
HELLO<br>
<% } %>
```

La seconde paire de balises ne peut contenir qu'une seule expression (et non une instruction). Le résultat de cette expression – qui ne peut être void – est inséré dans le code HTML. Remarquez que cette expression ne se termine PAS par un "`;`". Voici un exemple qui affiche la date courante dans la page HTML.

```
<%= java.util.Calendar.getInstance().getTime() %>
```

Dans le code Java ainsi inséré, vous pouvez accéder à l'objet requête http, reçu grâce à la variable prédéfinie "`request`". L'intérêt essentiel de cette variable est qu'elle permet de retrouver le contenu des paramètres HTTP (et donc le contenu des champs de saisie HTML) associés à la requête. La méthode proposée à cet effet est : "`String getParameter(String nomDeParametre);`".

L'autre variable prédéfinie et indispensable est "`session`". Elle désigne l'objet Session qui est associé à chaque connexion d'un navigateur. C'est dans cet objet que vous pouvez stocker les données propres à l'utilisateur. La méthode pour stocker un objet est :

```
void setAttribute(String nomDAtribut, Object attribut);
```

Celle qui permet de retrouver un objet stocké dans la session est :

```
Object getAttribute(String nomDAtribut);
```

D'une requête à l'autre, même si l'utilisateur change de JSP ou passe la main à une servlet, les informations contenues dans l'ob-

jet session demeurent. Ce sont bien les seules : celles qui ont été définies dans la JSP sont systématiquement perdues à la fin de chaque génération de page.

Il faut noter aussi que `getAttribute` et `setAttribute` n'acceptent ou ne rendent que des attributs sous forme d'objets. Il n'est pas possible de stocker directement des types simples comme `int`, `boolean`, `float`, etc. Il faut passer par l'intermédiaire des `DataTypeWrapper` (`Integer`, `Boolean`, `Float`), c'est à dire la version Objet des types simples. Les Strings étant déjà des objets, elles ne posent pas ce type de contrainte.

Voilà. Ce devrait être suffisant pour effectuer vos premiers pas. Je vous encourage, après vous être familiarisé avec les principes que je viens d'exposer, de vous plonger dans une documentation sur les JSP. Il en existe de qualité. C'est un vaste sujet contenant beaucoup de ressources intéressantes. Afin de ne pas vous lasser trop rapidement sur ce sujet, voilà un mini exemple qui fait un premier tour des possibilités offertes par cette technologie :

```
<html>
<head>
<title>Premiers pas</title>
</head>
<body>
<%
String donnee = request.getParameter("info");
Integer compteur = (Integer)session.getAttribute("compt");
if (compteur==null) compteur = new Integer(1);
%>
Bonjour tout le monde !
<form>
Saisir pour la <%=compteur%> e fois : <input name="info" value="<%=
donnee %>"><br>
Le contenu de ce champ disparaît toujours : <input name="autre"><br>
<input type="submit">
</form>
<% session.setAttribute("compt", new Integer(compteur.intValue()+1)); %>
</body>
</html>
```

Vous pouvez constater en faisant fonctionner cette page, que la valeur saisie du champ "`info`" est conservée – grâce à l'objet requête – et que le compteur de soumissions de pages s'incrémente au fil des requêtes – grâce à l'objet session.

Créer une application J2EE complète

Le composant J2ee le plus approprié, pour définir un service accessible à distance est le "`session bean`". C'est la forme la plus simple des composants de la famille des Enterprise Bean, les fameux EJB, dont on raconte ici et là l'effrayante légende, le soir au coin des serveurs, pour terroriser les petits développeurs... Il est très probable que dans quelques lignes, vous rirez de la crainte qu'ils inspirent, car d'un point de vue strictement J2ee, écrire et déployer un EJB est une tâche aisée.

Avant de procéder au développement de notre service, il faut définir au sein de Jboss une application qui va le recueillir. A l'instar de

J2EE

NIVEAU : DÉBUTANT

ce que nous avons fait pour l'application " myapp.war " nous allons créer un répertoire " myapp.jar ". L'extension " jar " signifie que les composants contenus dans notre nouvelle application sont des EJB. Dans notre application finale les composants web vont faire appel aux composants EJB. Donc " myapp.war " a besoin de " myapp.jar ". Dans un tel cas de figure, il faut réunir ces deux applications au sein d'une application englobante dite " ear ". Il existe une raison technique sous-jacente. Elle est liée à la manière dont fonctionnent les chargeurs de classes d'une JVM. En bref, nous allons créer un répertoire " myapp.ear " et nous allons déplacer " myapp.war " dans ce répertoire. Toujours au sein de " myapp.ear ", créons " myapp.jar ". Pour signifier à Jboss que l'application " myapp.ear " est constituée de deux modules, il faut définir un fichier nommé " application.xml " qui sera placé dans un répertoire nommé " META-INF " (pour méta information). Ce répertoire " META-INF " est lui même placé dans le répertoire "myapp.ear". Le contenu de " application.xml " est le suivant:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE application PUBLIC "-//Sun Microsystems, Inc.//DTD J2EE Application
1.2//EN" "http://java.sun.com/j2ee/dtds/application_1_2.dtd">
<application>
  <display-name>MyApp</display-name>
  <module>
    <ejb>myapp.jar</ejb>
  </module>
  <module>
    <web>
      <web-uri>myapp.war</web-uri>
      <context-root>myapp</context-root>
    </web>
  </module>
</application>
```

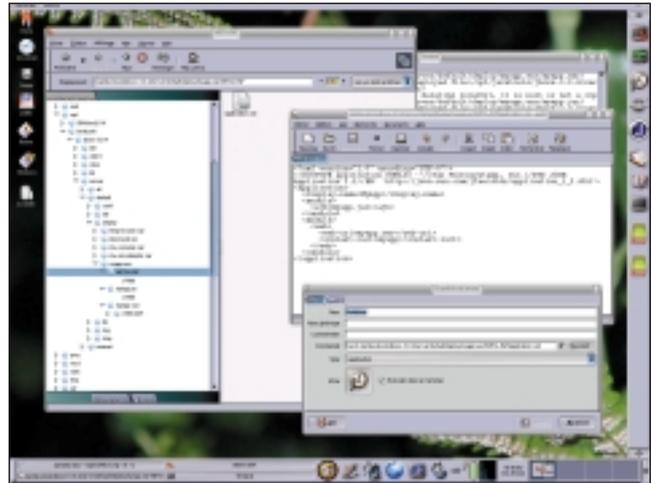
Constatons que la définition des deux modules est explicite. Nous remarquons que dans la déclaration de l'application intégrée "war" nous avons précisé – explicitement cette fois – le nom de l'application, tel qu'il doit apparaître dans l'URL de notre navigateur. C'est le rôle de l'élément " <context-root> " .

Premier déploiement

Sauvegardons le contenu de " application.xml ". Attendons quelques secondes en surveillant la fenêtre terminal sur laquelle s'exécute JBoss. Tout à coup celle-ci s'anime. Quelques lignes défilent puis l'affichage s'arrête sur une ligne qui ressemble à celle-ci:

```
13:58:07,560 INFO [MainDeployer] Deployed package: file:/opt/jboss-3.0.4/server/default/deploy/myapp.ear
```

Que s'est-il passé ? Jboss surveille en permanence les applications qu'il a déployées. En particulier il garde toujours un oeil sur le fichier de configuration principal de ces applications. Ce fichier, dans le cas des applications " ear ", est " application.xml ". Dès qu'il apparaît et ensuite chaque fois qu'il est modifié, Jboss, spontanément, redéploie l'application.



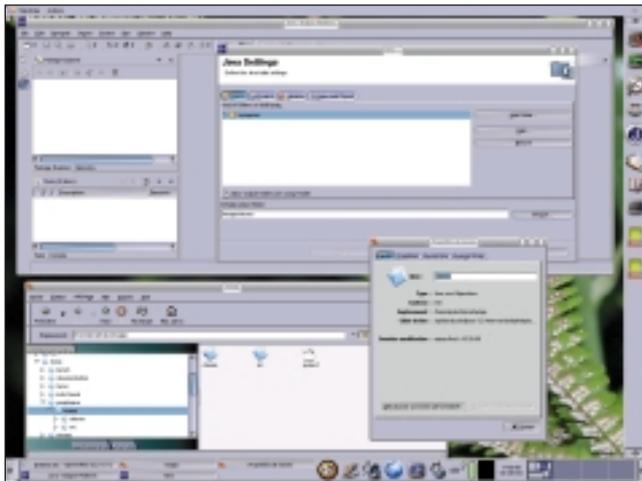
Cette capture d'écran montre l'organisation d'une application J2EE composée d'un module WEB (.war) et d'un module EJB (.jar). Ces deux modules sont déclarés dans le fichier application.xml. Cette capture montre aussi la définition du lanceur Linux, qui, en modifiant la date de mise à jour de ce fichier, provoque un redéploiement à chaud de l'application complète.

Ce redéploiement sait-il prendre en compte les mises à jour apportées aux classes Java (EJB et classes dépendantes) associées à l'application? La réponse est oui. En effet, une machine virtuelle java est capable de charger et décharger dynamiquement des classes.

Donc, chaque fois que l'on veut que nos mises à jour soient prises en compte, il suffit de faire croire à Jboss que notre fichier " application.xml " a été modifié. Sous Linux, c'est très simple. Il existe une commande qui fait ça très bien. Elle s'appelle " touch ". Je me suis personnellement fait un " lanceur d'applications " KDE qui exécute cette commande sur le fichier " application.xml ". Un clic de souris et l'application est redéployée ! Avec le clic nécessaire sous Eclipse pour sauvegarder et compiler les fichiers modifiés, ça fait deux clics. Et quatre secondes plus tard, mon application est prête pour un nouveau test. Sans IDE à 3 000 Euros ! Sans script Ant ! Sans passer des heures à configurer serveur d'applications et IDE ! Quant je pense aux projets que j'ai connus où entre deux essais il fallait taper plusieurs commandes et attendre jusqu'à 10 minutes ! C'est à ce genre de détail que l'on mesure le chemin parcouru ! Ici deux clics. Quatre secondes et c'est redéployé. Deux clics. Quatre secondes. On ne s'en lasse pas...

Sous windows, une solution pratique consiste à prendre le fichier " application.xml " sous l'IDE, d'effectuer une saisie factice (entrer un espace par exemple) afin que l'IDE accepte ensuite une nouvelle sauvegarde, et le tour est joué.

Passons maintenant à notre application intégrée EJB. L'arborescence des composants EJB (packages, classes) doit être placée directement sous le répertoire racine, donc sous " myapp.jar ". Il faut en plus créer, ici aussi, un répertoire " META-INF " qui va contenir le fichier de déploiement des EJB qui se nomme " ejb-jar.xml ". Son contenu est décrit dans les colonnes qui suivent.



Cette capture d'écran montre comment on peut connecter un projet Eclipse avec une application installée dans JBoss. Avec une telle installation, toute mise à jour effectuée à partir d'Eclipse est immédiatement reportée sur JBoss.

Intégration d'un environnement de développement

Nous avons créé notre premier composant Jzee à l'aide d'un éditeur de texte. Cette manière de faire est vraiment trop rudimentaire dès que l'on ambitionne de développer autre chose que des JSP. Il nous faut un outillage plus convivial. Nous allons voir maintenant, comment utiliser efficacement un environnement de développement Java pour construire une application Jzee.

Si vous pensiez que la programmation Jzee nécessite de disposer d'IDE onéreux disposant de fonctions très spécifiques aux EJB, j'ai pour vous une bonne nouvelle: vous allez pouvoir économiser quelques centaines, voire quelques milliers d'Euros. Les outils que je vous propose sont Jbuilder 8 personal edition ou Eclipse 2. Ces deux produits sont gratuits et facilement accessibles par le Web (www.borland.com pour Jbuilder et www.eclipse.org pour Eclipse). Ces deux environnements fonctionnent à merveille sur Linux et sur Windows. Je passe rapidement sur l'installation de ces deux produits, qui sort du cadre de cet article : Jbuilder propose un expert d'installation simple et convivial. Eclipse propose la même démarche que Jboss : on extrait le produit d'un fichier zip et ça marche, à condition de disposer au préalable d'un JDK récent, dont le répertoire " bin " est référencé dans la variable d'environnement PATH (sur Windows comme sur Linux). Intégrer Jboss et l'IDE consiste d'abord à paramétrer l'IDE pour que le résultat des compilations atterrisse dans la structure d'application de Jboss. Nous verrons que cela est facilement réalisable avec Eclipse et

Créé en 1991, Homsys Group accompagne les entreprises dans leurs innovations et la valorisation de leurs informations à des fins de productivité et de compétitivité. Homsys Group est spécialisé autour de trois grands pôles : la Business Intelligence, les Architectures Systèmes et Réseaux et les technologies Objet et Internet. Homsys Group, qui fédère aujourd'hui 250 consultants, est implanté à Bordeaux, Grenoble, Lyon, Marseille, Paris, Rennes et Toulouse. <http://www.homsysgroup.com>

JZEE

NIVEAU : DÉBUTANT

Jbuilder, sans que nous soyons contraints d'écrire des scripts Ant, make ou autre.

Eclipse et Jbuilder 8 définissent la notion de " projets ". Plusieurs projets peuvent coexister dans ces deux IDE. Les projets sont hautement paramétrables. Il est en particulier possible d'indiquer indépendamment (avec cependant une contrainte gênante pour Eclipse), un répertoire d'hébergement des sources et un répertoire d'hébergement des classes compilées. Le répertoire des sources peut être quelconque. Celui des classes sera dirigé dans la structure de Jboss vers le répertoire idoine : " server/default/deploy /myapp.ear/myapp.jar " .

Ouvrons l'IDE. Créons le nouveau projet. Les lecteurs qui ont opté pour Jbuilder peuvent créer le répertoire du projet où bon leur semble. A l'emplacement par défaut proposé par Jbuilder8 par exemple. Il ne leur reste plus qu'à indiquer le chemin des classes, donné plus haut, dans le paramètre du projet indiquant le répertoire de sortie (" Output path " sur l'expert de création de projet ou dans les propriétés du projet une fois que celui ci a été créé). Pour les utilisateurs d'Eclipse, c'est plus compliqué, car chemin des sources et chemin des classes ne sont pas vraiment indépendants (ils doivent posséder une racine commune !). Pour ceux qui ont installé Eclipse sur Linux, le problème peut être facilement contourné. Le projet peut être crée n'importe où (exemple "/home/projetsjava/myapp "). Il faut ensuite:

- indiquer à Eclipse que les sources se trouvent dans un répertoire " src " (idéalement placé sous le répertoire racine du projet). Pour cela, ajouter " myapp/src " à la liste " Source folders on build path " et répondre " yes " lorsqu'Eclipse vous propose de retirer la racine du projet de la liste des répertoires sources.
- créer un lien UNIX, appelé " classes ", placé à coté de " src " et qui désigne le répertoire " myapp.jar " .
- indiquer à Eclipse que le répertoire des classes est le lien que nous venons de créer. Pour cela, remplacez le contenu du champ " Default output folder " par " myapp/classes " .

Pour ceux qui utilisent conjointement Eclipse et Windows, la solution la plus simple consiste à définir le répertoire racine de notre application dans Jboss ("myapp.jar") comme racine pour les sources et pour les classes. C'est d'ailleurs le comportement par défaut d'Eclipse. Cette solution n'est pas très propre, mais à part utiliser un script de compilation Ant, je ne connais rien de plus adapté sur Windows.

Notre environnement de développement est maintenant presque prêt. Il ne lui reste plus qu'à découvrir les bibliothèques de Jzee. Pour cela il faut :

- Sur Jbuilder 8 : Editer le projet. Sélectionner l'onglet " Required libraries ". Appuyer sur le bouton " Add ". Une fenêtre modale de sélection de bibliothèques s'ouvre. Appuyer sur le bouton " New... ". Un expert de définition de bibliothèque apparaît. Entrez "jzee jboss" comme nom de la nouvelle bibliothèque. Appuyer sur le bouton "Add". Une fenêtre de sélection de fichiers/dossiers apparaît. A l'aide de cette fenêtre, sélectionner le fichier "server/default/lib/jboss-jzee.jar" de votre installation JBoss. Appuyez ensuite sur tous les boutons "Ok" que vous rencontrerez, afin de fermer toutes les fenêtres modales.
- Sur Eclipse : Editer le projet. Sélectionner la section "Java build path" puis l'onglet "libraries". Appuyer sur le bouton "Add External Jars". Une fenêtre de sélection de fichier apparaît. Sélectionner le fichier "server/default/lib/jboss-jzee.jar". Appuyer sur "valider" et ensuite sur "Ok" pour refermer la fenêtre de définition du projet. ■

Henri Darmet, consultant/formateur, Objet Direct

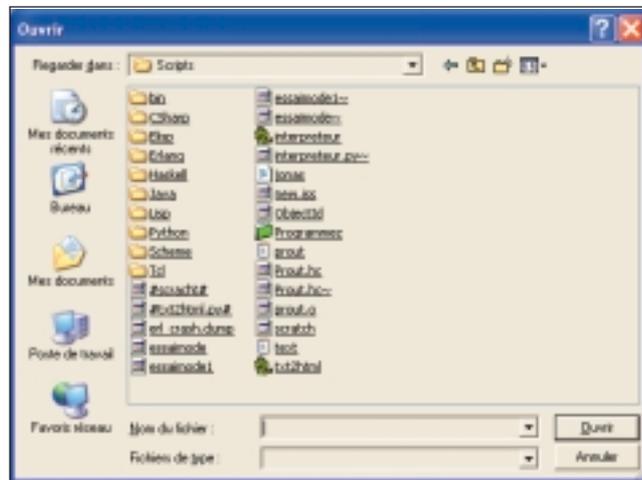
Les boîtes de dialogues prédéfinies de C#

Manipuler les boîtes de dialogues est facile avec C#, nous le savons déjà. Pour plus de facilité encore, C# propose des boîtes de dialogues prédéfinies. Tour d'horizon.

Le mois dernier, nous avons appris comment écrire le code de gestion d'une boîte de dialogue. Cela consistait principalement à instancier un objet Form de l'espace de noms System.Windows.Form, puis à faire apparaître la fenêtre proprement dite à l'écran avec la méthode ShowDialog. Cette méthode retourne, à l'issue des opérations, un code permettant de savoir si l'utilisateur a validé ses choix ou non.

La plupart des applications Windows ont comme point commun de proposer des boîtes de dialogues pour guider l'utilisateur dans des tâches récurrentes: ouvrir ou sauvegarder un fichier, sélectionner une couleur ou une police de caractères. Pour cela C# propose des boîtes de dialogues prédéfinies. Nous allons nous intéresser aujourd'hui aux boîtes de dialogues de fichiers, de couleurs et de polices. Le cas de l'impression et de la pré-visualisation avant impression sera étudié ultérieurement.

Les boîtes de dialogues prédéfinies obéissent à la règle que nous avons rappelée plus haut. Toutefois, les boîtes de dialogues prédéfinies de C# ne sont pas construites à partir de zéro, par dérivation de la classe Form, comme cela se fait pour une boîte de dialogue personnalisée. Au contraire, les classes C# qui nous intéressent aujourd'hui sont des encapsulations des boîtes de dialogues standard de Windows. Ces classes dérivent de l'ancêtre commun 'CommonDialog' de l'espace de noms System.Windows.Forms. Connaître ce point n'est certes pas indispensable pour manipuler une boîte de dialogue prédéfinie, mais cela permet d'en bien comprendre les fonctionnements particuliers, qui sont ceux des contrôles natifs de Windows. Prenons un exemple: la boîte de dialogue d'ouverture de fichier native conserve dans la base de registre, sous une sous-clé associée à l'extension, le nom du dernier fichier ouvert, avec son chemin complet. La classe C#, simple enveloppe autour du contrôle natif, fait tout naturellement de même.



> Figure 1: La classe OpenFileDialog en action

Ouverture de fichiers

Cette opération s'effectue avec la classe OpenFileDialog (figure 1). Cette classe présente bon nombre de propriétés dont nous récapitulons les plus intéressantes dans le tableau ci-dessous (toutes les propriétés sont en lectures écritures, sauf mention contraire).

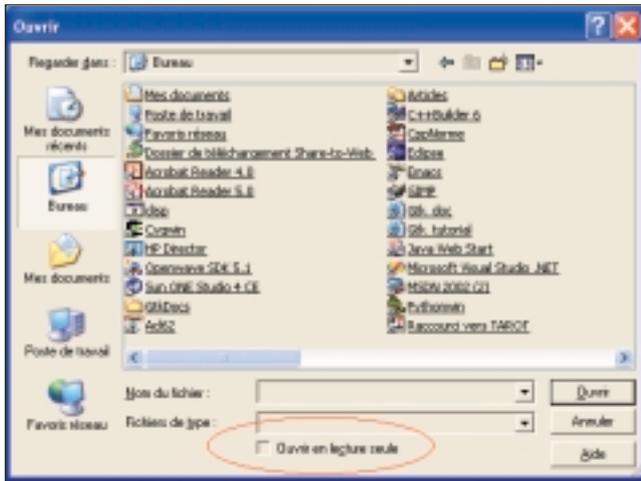
Nom	Description
Multiselect	Lorsque cette propriété vaut false (valeur par défaut), l'utilisateur ne peut sélectionner qu'un seul nom de fichier. Lorsque cette propriété est placée à true, l'utilisateur peut faire une sélection multiple.
FileName	Lorsque Multiselect est à false et que l'utilisateur a cliqué sur le bouton OK, cette propriété contient le nom du fichier sélectionné. Le chemin est inclus dans le nom de fichier.
FileNames	Cette propriété est en lecture seule. Lorsque Multiselect est à true et que l'utilisateur a cliqué sur le bouton OK, cette propriété contient un tableau de chaînes (string), chaque chaîne contenant le nom d'un fichier accompagné de son chemin. Travailler avec un tel tableau est une excellente occasion de découvrir la puissante instruction foreach de C#. Nous renvoyons le lecteur au listing d'exemple ci-contre (encadré 1)
InitialDirectory	Comme son nom l'indique, cette propriété permet de spécifier le répertoire à l'ouverture de la boîte de dialogue. Sinon c'est la valeur de la clé de registre qui est utilisée. Clé dans laquelle le dernier répertoire validé est conservé.
CheckPathExists	Si cette propriété est à false, l'utilisateur peut créer un répertoire qui n'existe pas encore. A true c'est impossible.
CheckFileExists	Même chose pour le nom de fichier.
ShowReadOnly	Mettre cette propriété à true faire apparaître une case à cocher dans la boîte de dialogue (figure 2). Cette case à cocher permet à l'utilisateur d'indiquer à l'application que le fichier devra être ouvert en lecture seule.
ReadOnlyChecked	Cette propriété permet de tester l'état de la case à cocher mentionnée plus haut. Si la case est cochée, l'usage veut d'ouvrir le fichier avec la méthode OpenFile de la boîte de dialogue elle-même. Ce faisant, le fichier est ouvert en lecture seule.
Filter	Permet d'indiquer les filtres d'extensions de fichier. Voir plus loin.
FilterIndex	Index de l'extension du nom de fichier. Voir plus loin.
ShowHelp	Affiche un bouton d'aide dans la boîte de dialogue.

C#

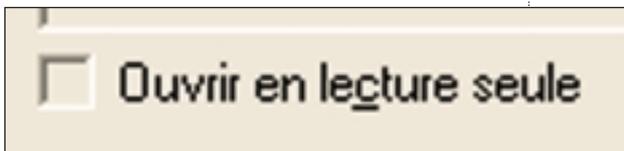
NIVEAU : DÉBUTANT

Filterer les fichiers.

Voyons maintenant en détail, comment spécifier un filtre d'extensions de fichier. Cette fonctionnalité est puissante. Il est possible d'indiquer plusieurs types de fichiers, chacun avec leurs exten-

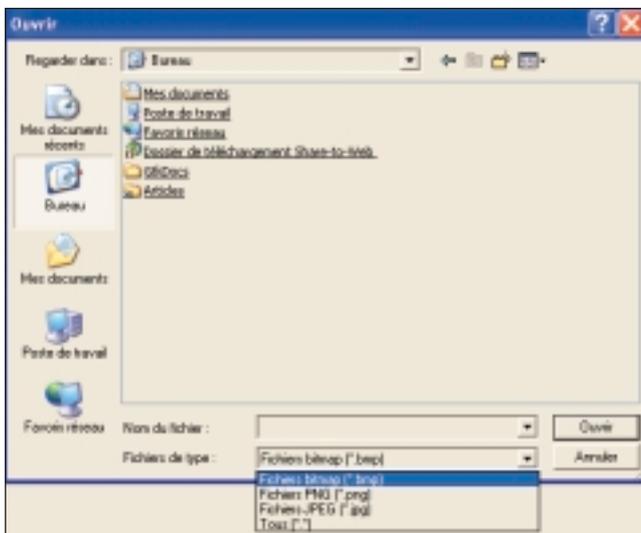


> **Figure 2:**
L'utilisateur peut demander une ouverture de fichier en lecture seule.



sions propres ou un type de fichier associé à plusieurs extensions. Il est encore possible de combiner ces possibilités. A l'écran les filtres apparaissent dans une boîte liste défilante à l'intérieur de la boîte de dialogue (figure 3).

Un filtre consiste en une chaîne de caractères composée de sous-



> **Figure 3:** Une boîte liste défilante énumère les filtres actifs.

chaînes séparées par le signe |. Une sous-chaîne est elle-même constituée de deux parties :

- Un texte à l'intention de l'utilisateur. Ce texte peut comporter à titre indicatif l'extension filtrée. La présence ou l'absence de cette extension est ignorée par le système.
- La ou les extensions filtrées. S'il y a plusieurs extensions, celles-ci doivent être séparées par un point-virgule.

Ces deux parties de la sous-chaîne doivent elles-mêmes être séparées par un signe | ce qui n'améliore pas la clarté des choses... Prenons donc quelques exemples pour nous fixer les idées :

```
"Fichier bitmap | *.bmp"
```

L'utilisateur est informé que l'on filtre des fichiers bitmap, sans être informé de ce qu'est réellement l'extension.

```
"Fichier bitmap (*.bmp) | *.bmp"
```

Même chose en informant l'utilisateur de la valeur de l'extension.

```
"Fichiers JPEG (*.jpg) | *.jpg;*.jpeg;*.jif"
```

On raconte à l'utilisateur que l'on filtre les fichiers JPEG d'extension .jpg, mais en réalité on valide les extensions jpg, jpeg et jif simultanément.

```
"Fichier bitmap (*.bmp) | *.bmp | Fichiers JPEG  
(*.jpg) | *.jpg;*.jpeg;*.jif"
```

Combinaison des deux cas précédents. Notez la présence de trois caractères | dans la chaîne.

Pour en terminer avec cela, signalons que lorsqu'un fichier est sélectionné dans un contexte de filtre, la propriété `FilterIndex` est mise à jour. La valeur est tout simplement le rang de la ligne dans la boîte liste défilante, sachant que la numérotation commence à la valeur 1.

Sauvegarder un fichier

C'est avec une logique implacable que cette opération s'effectue au moyen de la classe `SaveFileDialog`. Le comportement de cette boîte de dialogue a beaucoup en commun avec celle vue précédemment. Et son allure à l'écran ressemble à s'y méprendre à cette dernière (figure 4). Deux propriétés communes à `OpenFileDialog` induisent toutefois un comportement différent. Il s'agit des propriétés `CheckPathExists` et `CheckFileExists`. Si ces propriétés sont à `true` et que l'utilisateur spécifie un nom de répertoire ou de fichier qui n'existe pas, un message d'alerte est affiché (!) et la boîte de dialogue reste ouverte. L'intérêt de ceci ne saute pas immédiatement aux yeux, puisque lorsqu'on spécifie un nom de fichier, c'est normalement, pour aboutir à sa création... La classe `SaveFileDialog` comporte en outre trois propriétés intéressantes. Ces trois propriétés sont en lecture/écriture :

encadré 1

```
// Demonstration d'utilisation
// de OpenFileDialog (sur le Cd-Rom)

using System;
using System.Drawing;
using System.Windows.Forms;

class Demo : Form {

    public static void Main() {
        Application.Run(new Demo());
    }

    public Demo () {
        Text = "Démo de boîte de dialogue";

        Button bouton1 = new Button();
        bouton1.Parent = this;
        bouton1.Location = new Point(10, 10);
        bouton1.Text = "Simple";
        bouton1.Click += new EventHandler(OnSimpleClick);

        Button bouton2 = new Button();
        bouton2.Parent = this;
        bouton2.Location = new Point(10, 40);
        bouton2.Text = "Multiple";
        bouton2.Click += new EventHandler(OnMultipleClick);

        Button bouton3 = new Button();
        bouton3.Parent = this;
        bouton3.Location = new Point(10, 70);
        bouton3.Width = 100;
        bouton3.Text = "InitialDirectory";
        bouton3.Click += new EventHandler(OnInitialClick);

        Button bouton4 = new Button();
        bouton4.Parent = this;
        bouton4.Location = new Point(10, 100);
        bouton4.Text = "Filtre";
        bouton4.Click += new EventHandler(OnFilterClick);

        Button bouton5 = new Button();
        bouton5.Parent = this;
        bouton5.Location = new Point(10, 130);
        bouton5.Width = 100;
        bouton5.Text = "Help, ReadOnly";
        bouton5.Click += new EventHandler(OnHelpReadOnlyClick);
    }
}
```

```
void OnSimpleClick(object obj, EventArgs ea) {
    OpenFileDialog ofd = new OpenFileDialog();
    if(ofd.ShowDialog() == DialogResult.OK)
        Console.WriteLine(ofd.FileName);
}

void OnMultipleClick(object obj, EventArgs ea) {
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.Multiselect = true;
    if(ofd.ShowDialog() == DialogResult.OK)
        foreach(string s in ofd.FileNames)
            Console.WriteLine(s);
}

void OnInitialClick(object obj, EventArgs ea) {
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.InitialDirectory = "c:\\\\";
    if(ofd.ShowDialog() == DialogResult.OK) {
        Console.WriteLine(ofd.FileName);
    }
}

void OnFilterClick(object obj, EventArgs ea) {
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.Filter = "Fichiers bitmap (*.bmp) | *.bmp | " +
        "Fichiers PNG (*.png) | *.png | " +
        "Fichiers JPEG (*.jpg) | *.jpg;*.jpeg;*.jif | " +
        "Tous (*.*) | *.*";

    if(ofd.ShowDialog() == DialogResult.OK) {
        Console.WriteLine(ofd.FileName);
        Console.WriteLine(ofd.FilterIndex);
    }
}

void OnHelpReadOnlyClick(object obj, EventArgs ea) {
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.ShowHelp = true;
    ofd.ShowReadOnly = true;
    if(ofd.ShowDialog() == DialogResult.OK) {
        Console.WriteLine(ofd.FileName);
        ofd.OpenFile(); // etc, etc
    }
}
}
```

Frédéric Mazué *frederic.mazue@programmez.com*

SeaSide : Des Composants pour le Web

(Suite)

Le mois dernier nous vous avons montré la puissance de SeaSide pour construire des applications web. Nous vous avons fait voir comment le modèle à base de continuations de SeaSide permet d'implémenter des applications web comme n'importe quelle autre application, c'est-à-dire, sans avoir à modéliser le flot de contrôle de l'application de manière dédiée au modèle d'interaction du web. Ce pouvoir d'expression nouveau nous a permis de transformer très simplement un jeu en une application web. Ce mois-ci, nous présenterons le modèle de composants offert par SeaSide et aborderons plus précisément son architecture interne.

Rappel : comment obtenir et démarrer SeaSide

Obtenez Squeak 3.4 (www.squeak.org), le serveur web Comanche 5.0 <http://fce.cc.gatech.edu/~bolot/kom/kom-5-0.02> May1824.cs.gz et la dernière version de SeaSide (2.21) disponible à <http://beta4.com/seaside2/>. Chargez Comanche et SeaSide dans Squeak (open File List...). Pour démarrer Comanche configuré pour SeaSide, exécutez l'expression `WAKom startOn: 9090`. SeaSide doit maintenant écouter des requêtes sur le port 9090. Notez que si vous sauvez votre image, le service sera automatiquement lancé. Pour vérifier que SeaSide fonctionne bien ouvrez la `http://localhost:9090/seaside/counter` à votre browser. Vous devez arriver sur un exemple très simple : un compteur, comme montré par la figure 1. Cliquez sur les "+" et "-" (ce dernier doit produire une erreur comme nous l'expliquons un peu plus loin).



> Figure 1: SeaSide fonctionne et montre un compteur

Concepts de Bases : Sessions et Composants

Regardez l'URL qui apparaît dans le browser : il est composé d'un nombre et d'une séquence de lettres et chiffres séparés par un point-virgule, comme par exemple `2;78c9069f-3c61-11d7-a1d2-000393b2aa24`. La seconde partie, qui n'a pas changé durant votre utilisation, est un identifiant unique qui représente la session en cours. Une session est un concept central en SeaSide. Contrairement à la plupart des frameworks web, qui traitent les sessions principalement comme des entités, pour conserver l'état de l'application et s'occupent de gérer individuellement les cycles de requêtes et réponses, SeaSide traite une session à la manière d'un processus (thread): une session débute à un point donné et l'application est exécutée linéairement depuis ce point, affichant les pages web et attendant les données de l'utilisateur. Le lien "New Session" en bas de la page permet de réaliser une nouvelle session, i.e., crée un nouveau flot et initialise l'application.

Le nombre avant le point-virgule augmente régulièrement alors que l'application s'exécute. Ce nombre représente le nombre de requêtes par session et permet à SeaSide de connaître la progression de l'utilisateur. Notez que tout l'état de l'application est gardé sur le serveur. De manière similaire tout lien ou champ de formulaire possède un identifiant unique, que le serveur sait interpréter. Cette pratique a le désavantage de rendre les favoris illisibles, mais permet une très grande flexibilité.

Une application en SeaSide n'est pas composée de pages ou d'actions mais de *composants*, i.e., des objets qui modèlent l'état et la logique d'une partie de l'interface utilisateur (la page web). Ces composants peuvent être imbriqués les uns dans les autres et réutilisés entre applications. SeaSide propose un ensemble de composants prédéfinis. Quand une session démarre, une instance

Squeak

NIVEAU : INTERMÉDIAIRE

d'un composant est créée et celui-ci entre dans une *boucle de réponse* : il s'affiche et attend des entrées. Les entrées (sélection d'un lien et soumission d'un formulaire) déclenchent la méthode correspondante du composant. Quand cette méthode se termine, le composant s'affiche de nouveau. Les méthodes peuvent simplement changer l'état de l'application, du composant courant ou elles peuvent créer d'autres composants, ou encore transférer le contrôle de l'application à un autre composant qui entre alors dans sa propre *boucle de réponse*.

Caractéristiques d'un composant : état, action et affichage

Un composant a trois responsabilités: le maintien de l'état interne de l'interface (lorsque l'on se place dans une architecture découplant les objets domaines de l'application de leur représentation), la réaction aux actions de l'utilisateur et l'affichage. Regardons comment la classe WACounter fonctionne.

Etat. Alors que la plupart du temps, l'état d'une application est généralement stocké dans les objets du domaine ou une base de données, l'interface utilisateur possède aussi son propre état comme la valeur courante d'un champ. De telles informations sont stockées dans les variables d'instances du composant.

Pour notre compteur, le seul état à considérer est le compteur lui-même, comme vous pouvez le voir en lisant la définition de la classe WACounter, soit dans Squeak ou directement en cliquant sur le lien "Browse" (figure 3). Quand une instance est créée au début de la session, elle initialise la variable count à 0 (méthode initialize). Les liens "++" et "--" changent simplement la valeur de cette variable. Vous pouvez voir l'état du composant en cliquant sur le lien "Inspect", qui va ouvrir un inspecteur adapté au browser web, comme le montre la Figure 2.



> Figure 2: Inspecteur sur un compteur.

En plus de la variable count dont la valeur est utilisée dans le titre, un compteur hérite certaines variables de la class WACounter. Cliquez sur le lien "OK" pour revenir à l'application.

Action. Les actions sont représentées par les méthodes du composant. Dans le cas du compteur, deux actions sont possibles pour changer sa valeur. Le composant possède deux méthodes correspondantes (increment and decrement). Cliquer sur le lien "++" invoque la méthode increment dont le code est le suivant et qui ne fait rien de magique. Quand cette méthode se termine, la boucle de réponse se poursuit, le composant s'affiche en attendant la prochaine interaction.

```
WACounter>>increment
count := count + 1
```

```
WACounter>>decrement
self error: 'foo'.
```

```
count := count - 1
```

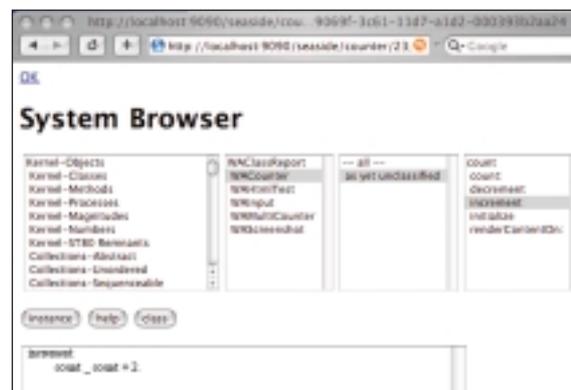
La version 2.21 de SeaSide montre comment les erreurs sont contrôlées et affiche la pile d'exécution dans un browser, en levant une erreur (méthode error: dans la méthode decrement). Afin de pouvoir interagir avec le compteur, redéfinissez cette méthode comme suit.

```
WACounter>>decrement
count := count - 1
```

Changez le code de cette méthode afin de pouvoir décrémenter le compteur en utilisant le lien "Browse". Pour compiler le code, cliquez sur le bouton "Accept" en bas de l'éditeur. Cliquez sur "Ok" pour revenir dans l'application.

Affichage. Pour afficher un composant, la méthode renderContentOn: est invoquée, avec comme paramètre une instance de WAHtmlRenderer. Chaque message ajoute du texte ou des éléments de documents comme, le montre la méthode renderContentOn: du compteur.

```
WACounter>>renderContentOn: html
html heading: count.
html anchorWithAction: [self increment] text: '++'.
html space.
html anchorWithAction: [self decrement] text: '--'.
```



> Figure 3: Pour coder, ou lire le code directement dans un navigateur Web.

Trois différents messages sont utilisés, chacun produisant des éléments HTML différents. heading: produit une section heading, space ajoute un caractère. anchorWithAction:text: produit les liens "++" et "--". En SeaSide, les liens n'ont pas de destinations, ils ont des callbacks: à chaque fois que l'on crée un lien ou bouton, on y associe un block. Lorsque ce lien ou bouton est activé, le block est évalué. Dans notre cas, les méthodes increment ou decrement sont invoquées.

La Boucle de Réponse

Regardons maintenant comment les composants sont connectés. Chaque composant effectue une *boucle de réponse* : s'affiche, attend des données, les traite et se réaffiche. Dans notre cas, une instance de WACounter est créée, la variable count est initialisée à zéro. SeaSide appelle la méthode renderContentOn: du compteur qui produit le document HTML montré dans la figure 1 avec

tous les liens, document qui est finalement affiché dans le browser. Lorsque l'utilisateur clique sur un des liens, le callback associé est exécuté et appelle les méthodes increment ou decrement qui changent l'état de l'objet. Une fois la méthode exécutée le compteur est de nouveau affiché, avec son nouvel état.

Interaction entre Composants

Pour transférer le flot de contrôle à un autre composant, la class WAComponent offre une méthode spéciale nommée call:. Cette méthode passe en argument le composant qui reçoit le contrôle et commence immédiatement la boucle de réponse du composant, qui s'affiche alors à l'utilisateur. Mais essayons plutôt cela : modifiez la méthode decrement, afin qu'un message soit affiché lorsque l'utilisateur essaye d'obtenir une valeur négative du compteur.

```
WACounter>>decrement
count = 0
ifFalse: [count := count - 1]
ifTrue: [self call:
        (WADialog new
         message: 'Pas de negatif!')]
```

Lorsque la variable d'instance count est nulle, une instance de WADialog est créée et passée comme argument à un appel à la méthode call:. Vous devez obtenir un message. En fait, afficher un dialogue est si courant, que SeaSide offre la méthode inform: que l'on utilise comme suit.

```
WACounter>>decrement
count = 0
ifFalse: [count := count - 1]
ifTrue: [self inform: 'Pas de negatif!']
```

La méthode inform: est légèrement différente. Cette fois, le dialogue propose un bouton "OK" qui vous ramène au compteur quand vous le pressez. La méthode inform: est définie comme suit:

```
WAComponent>>inform: aString
self call: (WADialog new
           message: aString;
           on: 'OK' answer: [true])
```

Comment tout cela fonctionne-t-il? L'instance de WADialog utilise la méthode answer qui rend le flot de contrôle au composant ayant invoqué la méthode call: originale. La méthode answer force la terminaison de l'invocation originale de call:. Le programme continue alors à partir de ce point. Dans notre exemple, l'appel au dialogue est la dernière instruction de la méthode decrement, donc cette méthode se termine et la boucle de réponse du compteur se poursuit.

Regardons exactement ce qui se passe.

(1) WACounter>>decrement appelle WAComponent>>inform:.
(2) WAComponent>>inform: crée une nouvelle instance de WADialog et la passe en argument de WAComponent>>call:.
Appelons ce point, le point d'appel. (3) Le WADialog commence sa

boucle de réponse et s'affiche dans le browser. (4) Quand l'utilisateur presse le bouton "OK", le dialogue invoque la méthode WAComponent>>answer. En fait, la méthode WAComponent>>answer ne termine jamais, car il saute directement au point d'appel, juste après l'appel de la méthode call:.
(5) WAComponent>>call: retourne à WAComponent>>inform:.
(6) WAComponent>>inform: retourne à WACounter>>decrement.
(7) WACounter>>decrement retourne à la boucle de réponse du compteur qui est alors affiché.

La méthode call: est en plus très puissante, car si un composant appelé spécifie un argument, lors de l'appel de la méthode answer:, cet argument sera retourné par la méthode call:. Ainsi invoqué, un composant peut rendre un résultat, ce qui est bien plus simple que mettre et enlever des composants d'une pile. Il devient alors très facile d'implanter la méthode confirm: qui affiche un texte et rend vrai ou faux suivant le choix de l'utilisateur. Modifiez la méthode decrement as comme suit:

```
WACounter>>decrement
count = 0
ifFalse: [count := count - 1]
ifTrue:
    [(self confirm: 'Allons dans les negatifs?')
     ifTrue: [self inform: 'Ok, allons-y'.
              count := -100]].
```

Maintenant si vous cliquez sur le compteur, vous allez obtenir trois pages différentes : le dialogue, le message 'Ok, allons-y' et finalement, le compteur lui-même.

Notez également que le bouton retour du browser est bien géré. Cet exemple montre la structure typique des applications développées avec SeaSide: au lieu d'avoir une série de pages, chacune connaissant exactement les précédentes et les suivantes, avec SeaSide chaque page contient de l'information présentée à l'utilisateur et ne doit pas se soucier du contexte dans lequel elle est utilisée, ce qui a l'incommensurable avantage de permettre de construire des pages réutilisables.

Vous ne l'avez peut être pas réalisé, mais ce n'est pas la première fois que vous avez rencontré le couple call:/answer. A chaque fois que vous avez cliqué sur "Inspect" ou "Browse", vous avez simplement invoqué les instances des composants WAInspector ou WABrowser, cliquant 'OK' utilise la méthode answer pour retourner à l'application principale. En fait, l'application principale est une instance de WAToolFrame, qui affiche la barre d'outils dans laquelle le compteur est inséré. SeaSide permet en effet d'imbriquer des composants les uns dans les autres, comme le montre l'exemple <http://localhost:9090/seaside/multi> mais ceci est une autre histoire ! Notez que SeaSide permet de gérer de manière plus contrôlée la sémantique du bouton back. Ceci permet de s'assurer que les objets restent dans des états cohérents par rapport à la page dans laquelle ils ont été modifiés et évite donc d'avoir des pages web désynchronisées quand on utilise les boutons de navigation, comme cela est encore le cas sur un grand nombre de sites. ■

Références

Squeak: <http://www.squeak.org/>

SeaSide: <http://beta4.com/seaside2/>

ESUG: <http://www.esug.org/>

Wiki Francophone: <http://www.iut3.unicaen.fr:8000/fsug/>

Livres Gratuits :

<http://www.iam.unibe.ch/~ducasse/WebPages/FreeBooks.html>

ESUG CD : <http://www.ira.uka.de/~marcus/Esugcd.html>

Squeak CD : <http://www.ira.uka.de/~marcus/SqueakCD.html>

Squeak

NIVEAU : INTERMÉDIAIRE



DirectX : quoi de 9 ?

Direct. C'est le mot qui symbolise le mieux DirectX depuis sa création. Direct pour " ligne directe " ou mieux pour " accès direct". Mais peu importe la signification historique de ce mot. Le parcours de DirectX depuis sa création en 1995 est devenu aussi prévisible qu'un algorithme déterministe. En effet, le plan de carrière de ce dernier est remarquable mais sans surprise, tout comme la dernière version du SDK (Software Development Kit) qui est opérationnelle depuis la fin de l'année 2002.

Un seuil de maturité technologique

Huit années se sont écoulées depuis sa création. DirectX semble atteindre un seuil de maturité technologique aujourd'hui. Totalement intégré dans l'univers des jeux vidéo, du multimédia et également de l'Internet pour Windows (via le composant DirectPlay), DirectX reste plus que jamais la référence absolue pour le développement d'application performante, éblouissante, fascinante... mais également complexe. Car c'est là le point faible de DirectX : il effraie les non initiés. Ses détracteurs premiers dénonçaient, à juste titre, sa complexité d'utilisation et d'intégration. Ils considéraient DirectX comme une pâle copie d'OpenGL, tant l'architecture et le design de ce dernier semblaient plus souples, voire plus agréables, que " l'immonde " usine à gaz de son principal concurrent. Mais il faut reconnaître que Microsoft a fait de gros efforts pour mieux diffuser DirectX parmi la communauté des développeurs de jeux vidéo. Si la version 1.0 de GameSDK (le nom historique de DirectX) fut une vraie catastrophe, la version 3.0 qui apparut quelques années plus tard commençait à présenter quelques caracté-

ristiques intéressantes. La fin de l'année 2002 à peine achevée, DirectX version 9.0 pointe fièrement son nez après une longue période d'hibernation. Fort de son bagage technologique impressionnant, surtout dans le domaine de la 3D, DirectX répond plus que jamais aux attentes des développeurs multimédia sur Windows ou Xbox. Mais la sortie de la version 9.0 peut laisser certains perplexes, pour ne pas dire déçus. Autant la transition entre la version 7.2 et la version 8.0 fut une véritable mini révolution pour les programmeurs, avec l'apparition du nouveau composant DirectX Graphics (permettant de gérer à la fois l'affichage 2D et 3D), autant la version 9.0 peut laisser quelques grincheux sceptiques. En effet, il peut être assez surprenant d'attendre aussi longtemps une nouvelle version présentant aussi peu de nouveauté. Selon certains développeurs, DirectX 9.0

aurait très bien pu d'ailleurs se nommer DirectX 8.3. Mais DirectX n'étant pas un produit qui a un impact commercial direct, il serait indélicat de le reprocher à Microsoft. Car DirectX est en premier lieu, et avant toute chose, un outil pour aider les développeurs à créer des jeux vidéo performants sous un système d'exploitation Windows.

DirectX 9.0, vers une nouvelle dimension ?

Ouvrons les entrailles de la bête pour confirmer (ou infirmer) notre première impression. DirectX est-il une véritable nouvelle version ? Une vraie petite révolution ou bien une simple évolution ?

Pour en avoir le cœur net, donnons la parole à l'équipe de développement du SDK, par l'intermédiaire de l'excellente et imposante documentation du dernier SDK (voir la figure n°1).

S'il y a une constante appréciable et récurrente dans les différentes versions du SDK, c'est bien la qualité indiscutable de sa documentation. Bien que le ton de cette dernière soit toujours aussi neutre, à l'instar d'un manuel utilisateur, il faut admettre que la documentation de DirectX demeure à la fois complète, limpide et ergonomique. En tout cas, l'une des meilleures conçues. Dans un style toujours très sobre et sans fioritures, cette documentation permet un accès direct aux informations recherchées, avec une interface de navigation simple mais efficace, thématique ou avec moteur de recherche. La documentation présente également les nouveautés de la version 9 en commençant par l'introduction de nouveaux outils.

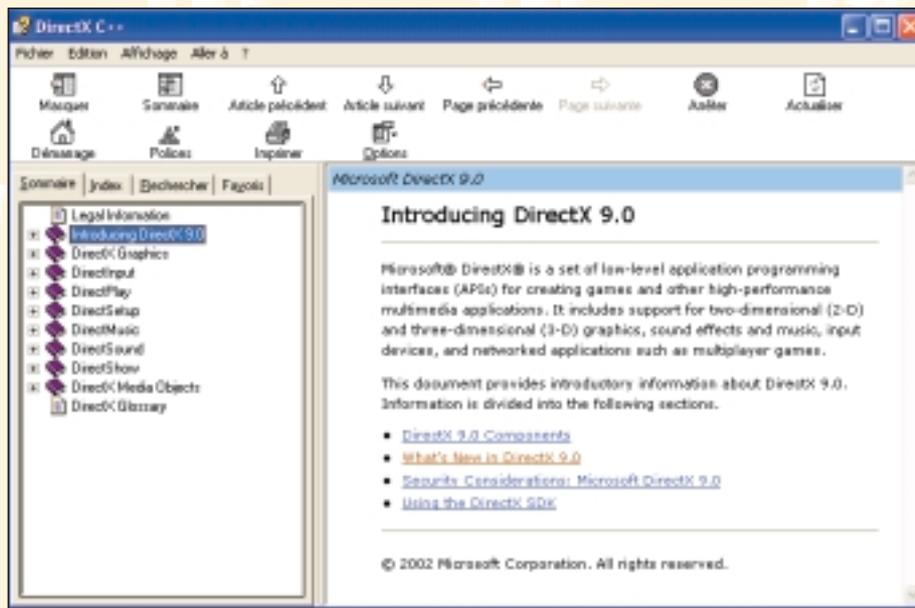


Figure n°1 : la documentation du SDK DirectX 9.0

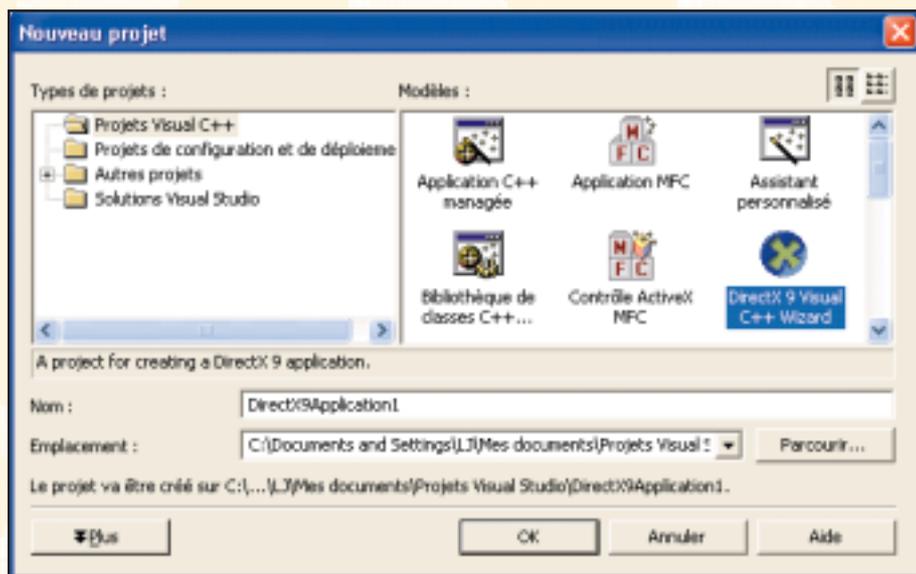


Figure n°2 : un DirectX Application Wizard enfin intégré avec Visual C++

Un peu de magie peut être parfois utile

Deux outils ont vu le jour avec cette nouvelle version. Le premier est un wizard qui porte le nom original d'AppWizard. Cet outil permet de générer automatiquement un petit projet C++ dit template, qui fournit les fonctionnalités de base pour créer rapidement une application DirectX. Rien de révolutionnaire en soi, notons que cet outil aurait dû apparaître plus tôt. Il est surprenant en effet de voir cet utilitaire émerger seulement avec la version 8.1 puis avec la version 9.0 pour Visual C++... quoiqu'il en soit, les développeurs DirectX chevronnés éviteront d'utiliser cet outil, alors que les néophytes l'accueilleront probablement avec soulagement. (figure n°2)

Un chasseur de bug

Le second outil disponible avec DirectX 9.0 est l'outil Error Lookup Tool. Cet utilitaire permet de retourner un message d'erreur explicite et textuel à partir de codes erreurs hexadécimaux. Autrement dit, il sert de traducteur lors du développement et permet de décrire avec un peu plus de précision le bug à traquer. (figure n°3)

Ces deux outils sont intéressants, surtout l'Error Lookup. Cependant, ils ne sont pas nouveaux, contrairement à ce qu'annonce la documentation du SDK. Les utilisateurs de la version 8.1 de DirectX s'en servaient déjà depuis quelques mois.

Tout pour le graphisme

C'est devenu une habitude, DirectX Graphics, qui réunit à la fois les composantes de gestion graphique 2D et 3D de DirectX, bénéficie une fois de plus des principales améliorations du SDK. Les autres composants resteront encore les parents pauvres de cette nouvelle version.

Les shaders passent en force

On ne présente plus les shaders aujourd'hui. Ces petits programmes, généralement codés en Assembleur, sont devenus des éléments essentiels de la programmation d'effets spé-

ciaux graphiques. La version 8.0 de DirectX a vu apparaître ses premiers shaders sous la forme de deux catégories : les pixels shaders et les vertex shaders. La version 9.0 amplifie ce phénomène en introduisant une version 2.0 (bientôt suivie par une 3.0) pour les shaders qui ne seront plus représentés sous la forme d'instances mais plutôt sous la forme d'objets COM avec un jeu d'instructions plus important, passant de 256 pour la version 2.0 à plus de 512 pour la future version. (figure n°4)

Le support HLSL

Ecrire des shaders en Assembleur est pour beaucoup une tâche à la fois complexe et fastidieuse. Bien conscient de ce problème, Microsoft propose dans la version 9.0 un nouveau langage évolué, proche du langage C d'un point de vue syntaxique. Il permet ainsi de faire abstraction du code Assembleur et de concevoir des programmes shaders plus aisément. Ce langage porte le nom barbare, mais parlant, de langage HLSL (High-Level Shader Language ou langage shader haut niveau en français). Notons également que les principaux constructeurs de cartes vidéo proposent également un langage évolué pour la programmation de shaders. Le plus évolué d'entre eux est incontestablement le Cg (C for graphics) de nVidia. (figure n°5)

Plus qu'une simple alternative pour les allergiques au code Assembleur, le HLSL permet le développement de shaders sous forme de fonctions avec de véritables conditions, type de données standard, directives preprocessur, etc.

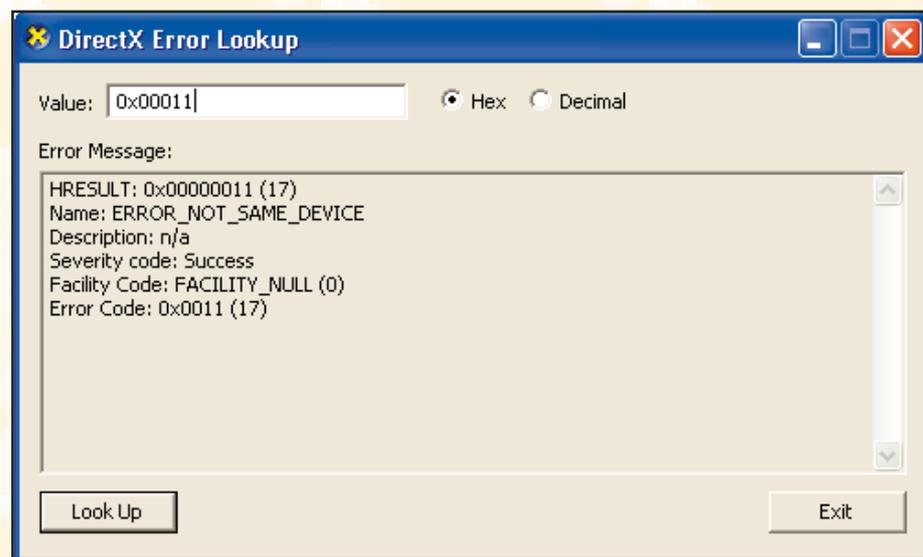


Figure n°3 : l'Error Lookup Tool s'avère particulièrement efficace pendant les phases de debugging

La version actuelle du HLSL ne supporte pour l'instant que les conditions de type if()...else. Mais la version future proposera les conditions classiques de type for() et do...while(). Bien entendu, les conditions de type switch...case ne sont pas au programme, car ces dernières consomment beaucoup trop d'instructions Assembleur. En revanche, le Cg de Nvidia proposerait, dans son évolution prochaine, des conditions de type switch...case.

Bienvenue au langage C#

Dernier arrivé parmi la famille MS, le langage C# bénéficie étonnamment d'une première compatibilité avec DirectX. Information étonnante, vu la jeunesse du langage C# (en comparaison, VB a dû attendre la version 8.0 de DirectX pour être enfin compatible). Mais cette compatibilité montre bien la volonté de MS d'inscrire DirectX comme une composante évolutive de sa politique .NET.

Cependant, que les supporters du C# ne s'emballent pas trop vite, les exemples et possibilités de compatibilité C# sont bien loin d'atteindre celles disponibles en C++. Gageons que MS proposera une compatibilité



Figure n°4 : les effets d'eau font partie des applications les plus spectaculaires des shaders.

plus poussée vers C# pour la prochaine version 9.1 du SDK DirectX.

Quoi de 9 à l'horizon ?

DirectX reste incontestablement le SDK de référence pour la programmation multimédia sous Windows ou Xbox. Depuis la version 6.0,

intéressantes, mais DirectX 9.0 n'est pas la révolution attendue. Il est en effet regrettable de constater que les autres composants de DirectX, comme DirectPlay, ne bénéficient pas de la même cadence de développement de la part de Microsoft. Sans être véritablement abandonnés, ces derniers ne subissent pas de grosses évolutions et c'est dommage, car certains d'entre eux mériteraient un sérieux remaniement interne.

DirectX poursuit son ascension fulgurante et la version 9.0 confirme la bonne forme de DirectX sur les différentes plates-formes de Microsoft. Surtout depuis la récente défection de Microsoft du consortium OpenGL, qui a toujours autant de mal à sortir sa fameuse version 2.0 tant attendue. Cependant, la version 9.0 n'est peut être pas un modèle d'innovation. Certes, les évolutions proposées, notamment au niveau de la gestion approfondie des shaders, sont

Quoi qu'il en soit, il semblerait qu'un seuil technologique commence à être atteint, surtout au niveau de la 3D. Les évolutions futures de DirectX seront probablement moins impressionnantes que ne l'était le passage de la version 5.2 à la version 6.0 ou mieux, de la 7.2 à la version 8.0. Nous attendons donc la version 9.1 (qui devrait être disponible dans quelques semaines) pour être fixés sur l'évolution future du SDK. Mais il est préférable de ne pas s'attendre à de trop grosses évolutions.

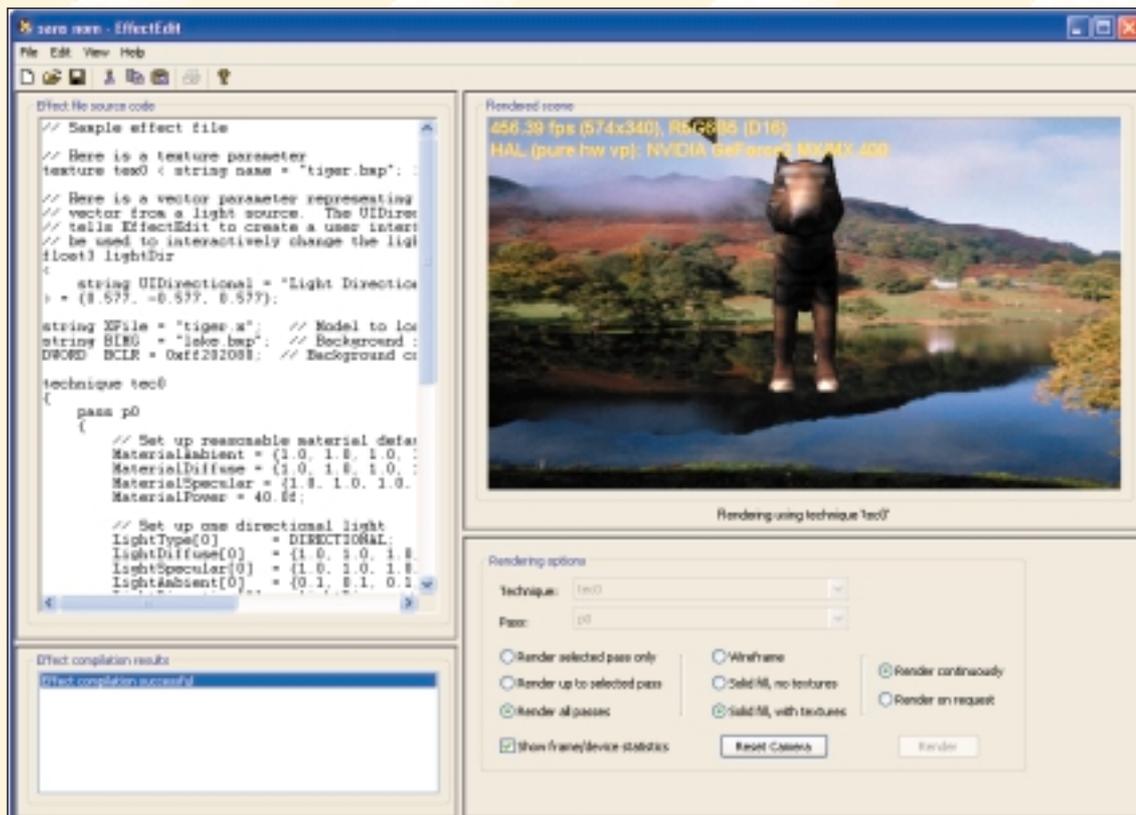


Figure n°5 : le programme EffectEdit du SDK permet d'écrire des effets shaders en HLSL et de les visualiser en temps réel.

Laurent Jayr
dev.net@online.fr