Aperçu des expressions régulières sous .NET

De nombreux langages implémentent les expressions régulières. Mais leurs interprétations et appels peuvent différer selon le langage. .NET offre de son côté une bibliothèque partagée d'expressions régulières qui a pour avantage d'unifier leur utilisation pour l'ensemble des langages de cette plate-forme. Nous allons vous décrire comment utiliser des expressions régulières en C# sous VS.NET ou avec Mono sous Linux.

Une expression régulière est constituée de " motifs " qui permettent de vérifier qu'une chaîne de caractères respecte bien certaines règles syntaxiques. Avec une expression régulière, vous pourrez identifier du texte, repérer des doublons, réaliser des remplacements, ou même scinder facilement une chaîne. Un utilisateur encode une chaîne : comment facilement valider l'expression, en y repérant un motif valable ? Exemple : l'utilisateur tape une adresse de courrier électronique, un code postal, une adresse IP, un numéro de carte VISA, et vous devez en vérifier la validité. Bref, comment détecter dans un flux de données quelconque un certain motif ?

.NET incorpore un moteur d'expressions régulières. Celui-ci est atteignable par la classe Regex de l'espace de nom "System.Text.Regular Expressions". Pour valider une adresse IP, nous créerons une nouvelle expression régulière "maregexp". Notez qu'elle est précédée en C# de "@" pour éviter la production par le compilateur de l'erreur "séquence d'échappement non reconnue".

```
VALIDATION.CS
using System;
using System.Text.RegularExpressions;

namespace Validation
{
  public class ProgCS
  {
  static void Main()
  {
    string MaChaine = "";
    Regex maregexp = new Regex(@"^(([01]?\d\d?|2[0-4]\d|25[0-5])
\.([01]?\d\d?|2[0-4]\d|25[0-5])\.([01]?\d\d?|2[0-4]\d|25[0-5])
\.([01]?\d\d?|2[0-4]\d|25[0-5]))$");
   Console.WriteLine("Encodez une adresse IP valide ou non valide pour test");
    MaChaine = Console.ReadLine();

    bool bresultat = maregexp.IsMatch(MaChaine);

    if (bresultat)
      Console.WriteLine("La chaîne est valide");
    else
      Console.WriteLine("La chaîne n'est pas valide");
    }
  }
}
```

La méthode IsMatch validera ou non l'expression, en renvoyant une valeur booléenne qui sera ensuite testée pour en déterminer le résultat. L'expression régulière que nous avons appliquée semble relativement difficile à appréhender. Réfléchissons : nous devons tester la présence de quatre nombres séparés par des points. Si nous plaçons notre regex entre les caractères "^" et "$", nous nous assurons que la chaîne ne contient que l'adresse IP. Ensuite nous décrivons la situation : un nombre, plus un point, plus un deuxième nombre, plus un second point, plus un troisième nombre, plus un troisième point, plus un quatrième nombre.

Ce qui donne : "^[0-9]+\. [0-9]+\. [0-9]+\. [0-9]+$".

Le "." ayant une signification particulière dans le "mini langage" des expressions régulières (*), vous devez le faire précéder de "\" pour qu'il soit interprété comme un point. Le caractère "+" veut dire " au moins un chiffre". Il est possible d'écrire ceci autrement en remplaçant [0-9] par "\d" ce qui revient au même.

Ce qui donne à son tour : "^\d+\.\d+\.\d+\.\d+$"...

Mais cette expression régulière demeure trop permissive, il nous faut ajouter une règle à notre description du problème : chaque nombre doit être compris entre 0 et 255. Le caractère "?" désigne un groupe, le caractère "|" une alternative et {n} signifiant " n fois exactement".

Si le nombre composé de trois chiffres débute par 2, il est valide, à condition que sa valeur soit inférieure à 255. Le second chiffre dans ce cas devra être inférieur ou égal à 5. Et si le second est égal à 5 alors le dernier sera inférieur à 6...

Ce qui donne : "2[0-4]\d|25[0-5]", ou bien "\d|\d\d[01]\d|\d|2[0-4]\d|25[0-5]", ou encore "[01]?\d\d?|2[0-4]\d|25[0-5]".

Au final nous pouvons par conséquent écrire :
Regex(@"^(([01]?\d\d?|2[0-4]\d|25[0-5])\.([01]?\d\d?|2[0-4]\d|25[0-5])\.([01]?\d\d?|2[0-4]\d|25[0-5])\.([01]?\d\d?|2[0-4]\d|25[0-5]))$");

A la recherche de l'expression régulière idéale

Si vous n'avez pas le temps ou l'expérience des expressions régulières, il existe sur le Web une ressource incontournable qui pourra vous faire gagner énormément de temps, en proposant des expressions régulières clé en main (http://regexlib.com/Search.aspx). Non seulement, vous avez

à votre disposition un moteur de recherche, mais en sus, vous pouvez tester n'importe quelle expression régulière à partir de votre navigateur. En recherchant "IP" nous trouvons :

"\b((25[0-5]|2[0-4]\d|[01]\d\d|\d?\d)\.){3}(25[0-5]|2[0-4]\d|[01]\d\d|\d?\d)\b"

Ou encore :

"^(\d{1,2}|1\d\d|2[0-4]\d|25[0-5])\.(\d{1,2}|1\d\d|2[0-4]\d|25[0-5])\.(\d{1,2}|1\d\d|2[0-4]\d|25[0-5])\.(\d{1,2}|1\d\d|2[0-4]\d|25[0-5])$"

Ces expressions sont plus alambiquées que notre solution initiale, mais peu importe : elles fonctionnent.

Dans la même veine, il est possible de tester un numéro de téléphone :

"^0[1-6]{1}(([0-9]{2}){4})|((\s[0-9]{2}){4})|((-[0-9]{2}){4})$"

Ou bien un code postal :

"^(F-)?((2[A|B])|[0-9]{2})[0-9]{3}$"

Scinder et remplacer

La classe Regex de l'espace de nom "System.Text.RegularExpressions" possède d'autres tours dans son sac, comme la méthode Split() qui permet de scinder une chaîne en sous-chaîne. Vous "splitez" la chaîne en précisant le, ou les caractères de séparation (ici le "." c'est à dire "\.").

```
SPLIT.CS
using System;
using System.Text.RegularExpressions;

namespace SPLIT
{
  public class ProgCS
  {
  static void Main()
  {
    string MaChaine = "";
    Regex maregexp1 = new Regex(@"[\.]");
    Regex maregexp2 = new
        Regex(@"^(([01]?\d\d?|2[0-4]\d|25[0-5])\.([01]?\d\d?|2[0-4]
\d|25[0-5])\.([01]?\d\d?|2[0-4]\d|25[0-5])\.([01]?\d\d?|2[0-4]
\d|25[0-5]))$");

    Console.WriteLine("Encodez une adresse IP valide :");
    MaChaine = Console.ReadLine();

    bool bresultat = maregexp2.IsMatch(MaChaine);

    if (bresultat)
      {
      Console.WriteLine("La chaîne est valide");

      string[] separeip = maregexp1.Split(MaChaine);
      foreach (string partie in separeip)
        Console.WriteLine(partie);
    }
    else
      Console.WriteLine("La chaîne n'est pas valide");
    }
  }
}
// mcs split.cs ; mono split.exe
//Compilation succeeded
//Encodez une adresse IP valide :
//192.168.0.29
//La chaîne est valide
//192
//168
//0
//29
```

(*) Notes :

- *Le "." signifie n'importe quel caractère sauf le caractère de fin de ligne "\n".*
- *Le "( )" représente un bloc ou une sous- expression.*
- *Le "^ " en début d'expression signifie que cette dernière doit être en début de ligne, $ en fin signifie qu'elle termine la ligne.*
- *Le "*" signifie répétition o ou plusieurs fois*
- *Le "\w" exprime un mot (équivalent à [a-zA-Z_o-9])*
- *Le "\W" veut dire " tout sauf un mot" (équivalent à [^a-zA-Z_o-9])*
- *Le "\s" signifie un "blanc" (équivalent à [\f\n\r\t\v])*
- *Le "\S" signifie un "non blanc" (équivalent à [^ \f\n\r\t\v])*
- *Le "\d" signifie un chiffre (équivalent à [o-9])*
- *Le "\D" signifie un "non chiffre" (équivalent à [^o-9])*

Exemples :

Recherche de "é" ou de "è" : "^(élè)$"

Recherche d'une lettre minuscule comprise en "x" et "z" : "[x-z]"

Recherche d'une lettre comprise en "x" et "z" minuscule ou majuscule : "[x-zX-Z]"

Tous caractères, excepté les lettres minuscules : "[^a-z]"

Maintenant, vous pouvez également remplacer une sous-chaîne dans une chaîne avec la méthode Replace() qui prend comme argument optionnel le nombre de remplacements. Voici un petit exemple relatif à cette méthode :

```
REPLACE.CS
using System.Text.RegularExpressions;

namespace Remplace
{
        public class ProgCS
        {
                public static void Main()
                {
                        Regex  maregexp   = new Regex("(rationelle|cartésienne)");
                         string machaine = "Les expressions cartésiennes permettent
de parser du texte. Dot NET supporte les expressions cartésiennes.";
                        Console.WriteLine(maregexp.Replace(machaine, "régulière", 2));
                }
        }
}
//# mcs remplace.cs ; mono remplace.exe
// Compilation succeeded
// Les expressions régulières permettent de parser du texte. Dot NET
supporte les expressions // régulières.
```

Matcher et spliter

Mais n'y aurait-il pas moyen de "matcher" une chaîne avec une expression régulière, de manière à la valider et à la "spliter" en sous- chaînes en une seule opération ? Si nous modifions l'expression régulière afin de la "baliser" par les caractères "‹...›", nous créons des groupes. Nous regroupons ainsi chaque nombre en le baptisant (ip1,ip2,ip3,ip4). Ensuite la méthode Match() applique l'expression régulière en scindant la chaîne en 4 groupes : nous n'avons effectivement plus besoin de la méthode Split().

```
SPLIT2.CS
using System;
using System.Text.RegularExpressions;

namespace SPLIT2
{
  public class ProgCS
  {
  static void Main()
  {
    string MaChaine = "";
    Regex maregexp = new
Regex(@"^((?<ip1>[01]?\d\d?|2[0-4]\d|25[0-5])\.(?<ip2>[01]?\d\
d?|2[0-4]\d|25[0-5])\.(?<ip3>[01]?\d\d?|2[0-4]\d|25[0-5])\.(?<ip
4>[01]?\d\d?|2[0-4]\d|25[0-5]$

    Console.WriteLine("Encodez une adresse IP valide :");
    MaChaine = Console.ReadLine();

    Match resultat = maregexp.Match(MaChaine);

    if (resultat.Success)
      {
      Console.WriteLine("La chaîne est valide");
      Console.WriteLine(resultat.Groups["ip1"].Value);
      Console.WriteLine(resultat.Groups["ip2"].Value);
      Console.WriteLine(resultat.Groups["ip3"].Value);
      Console.WriteLine(resultat.Groups["ip4"].Value);
    }
    else
      Console.WriteLine("La chaîne n'est pas valide");
    }
  }
}
// mcs split2.cs ; mono split2.exe
// Compilation succeeded
// Encodez une adresse IP valide :
// 192.168.0.1
// La chaîne est valide
// 192
// 168
// 0
// 1
```

Resultat.Sucess joue le même rôle que valeur booléenne IsMatch(). Si la chaîne est valide, celle-ci est scindée, en indiquant le nom de chaque groupe comme Groups["ip4"], mais nous aurions pu très bien indiquer Groups[4+1] (+1 car Groups[0] représente la chaîne non splitée).

La recherche de l'expression régulière idéale peut occuper un programmeur de nombreuses heures, surtout si celui-ci n'en maîtrise pas tous les mécanismes sous-jacents. Alors, il vaut mieux garder la tête froide : une bonne expression régulière sera celle qui sera la plus simple, la plus facile à définir, sinon vous risquez de perdre trop de temps à la constituer.

■ **Xavier.Leclercq**
Xavier.Leclercq@programmez.com

Apple : la nouvelle plate-forme de développement ?

À côté de Linux ou Windows, MacOS X constitue une plate-forme applicative et de développement de plus en plus intéressante, ouverte et performante. De plus, plusieurs technologies sont disponibles en Open Source et sur diverses plates-formes. A l'heure d'Apple Expo, faisons le point sur le marché Apple et sur MacOS X, en tant que plate-forme de développement.

Tiger : le prochain MacOS X v10.4

Prévue dans moins d'un an, la prochaine grande version de MacOS X estampillée 10.4, alias Tiger, introduira plus de 150 nouvelles fonctions, dont certaines risquent de faire grand bruit. Une fois n'est pas coutume, Apple prend de vitesse Microsoft qui ne sortira Longhorn qu'en 2006, si tout va bien.

On trouvera dans Tiger, un nouveau système de recherche. Jusque-là, rien d'extraordinaire. Or, quand on regarde de plus près la technologie employée, on se rend compte qu'elle ressemble à celle que Microsoft va utiliser dans Longhorn pour son nouveau file system, WinFS.

Spotlight (le nom du moteur de recherche de Tiger) s'appuie sur une nouvelle manière de présenter les données de dossiers et documents des disques. Le système crée des index de metadatas, qui seront ensuite comparés et analysés par Spotlight. Cette technique doit permettre d'accéder aux informations demandées bien plus rapidement qu'aujourd'hui. Le seul hic est le temps CPU nécessaire aux indexations. Apple avait déjà prévu une technologie un peu similaire dans le projet Copland (il y a plus de 10 ans déjà). BeOS avait réussi à créer une telle architecture fonctionnelle. Apple propose là une solution comparable à celle de Microsoft, mais avec une certaine avance. On sait aussi qu'un kit de développement sera disponible, afin d'intégrer Spotlight dans son application.

L'un des soucis d'Apple, de MacOS X et des applications très lourdes graphiquement est la capacité du moteur graphique à supporter des charges de travail. Quartz Extreme proposait déjà une réponse au problème en offrant une accélération matérielle aux applications compatibles. Avec Tiger, on ira encore plus loin avec le Core Image. Il s'agit là de pouvoir exploiter au mieux les possibilités des cartes graphiques pour les images. À cela se couple le Core Video dédié uniquement à la vidéo. Cela promet des performances radicalement différentes.

Côté développeur, il y en aura pour tous les goûts et niveaux. Tout d'abord, pour l'utilisation de tous les jours, on bénéficiera d'un nouvel outil, Automator. Il permettra d'automatiser l'ensemble des tâches répétitives du système. Comme MacOS X est avant tout un système Unix, Tiger met à jour un certain nombre de fondations Unix. Ainsi, le cœur du noyau évoluera en BSD 5.x, support important car cette version améliore la montée en charge sur plusieurs processeurs, ainsi que de nouveaux daemons réseaux. Les nouveaux pointeurs 64 bits inclus au cœur de Tiger permettront de mieux tirer profit des G5, processeurs 64 bits. On aura aussi droit à l'implémentation de libxslt pour la transformation XML et de SQLLite.

Pour la programmation, on aura droit à l'ensemble de développement Xcode 2.0 (la 1.5 est actuellement en bêta). Cette version supportera bien mieux le 64 bits, tout en assurant la compatibilité 32. On pourra même générer des

Suite page 18 ▸

Apple Expo 2004

A l'heure où nous écrivons, Apple Expo n'a pas encore débuté. On sait depuis début juillet qu'un nouvel iMac doit arriver durant l'Apple Expo. Plusieurs conférences auront lieu sur le développement (et l'optimisation G5) ainsi que sur les serveurs et la haute disponibilité. Comme tous les ans, un espace développeur sera là pour voir et essayer les outils de programmation Apple, avec la présence d'éditeurs indépendants. Linux et l'univers Open Source / Libre devraient se faire remarquer. Autre innovation bien involontaire (semble-t-il suite à un problème de calendrier du Parc des Expositions), Apple Expo se déroule bien plus tôt, tout début septembre...

CONFERENCES APPLE-EXPO

Conférence Développeurs
Mac OS X, outils de développement et optimisation des performances
Mercredi 1er septembre, 12h00 – 13h00 et 13h30 – 14h30

Compatibilité Mac OS X et Windows en toute simplicité
Jeudi 2 septembre, 10h30 – 11h30 et 16h30 – 17h30
Vendredi 3 septembre, 13h30 – 14h30

Mac OS X dans l'enseignement supérieur : l'Open Source simple d'accès
Mercredi 1er septembre de 10h30 à 11h30

Xserve, Mac OS X Server et Xsan
Vendredi 3 septembre, 10h30 – 11h30

Apple et le High Performance Computing
Mardi 31 août, 17h30 – 18h30

Mac OS X et la Sécurité
Jeudi 2 septembre, 18h00 – 19h00

Inscriptions :
www.apple-expo.com, espace "particuliers" & espace "professionnels".

Agenda des conférences :
http://www.apple-expo.com/fr/conferences_apple/

applications 32 – 64 (le fat binary). Surtout, pour la première fois, Apple va introduire un développement par modélisation, avec conception itérative à la clé. Une bonne nouvelle pour les développeurs. Surtout, pour simplifier les recherches dans la volumineuse documentation technique d'Apple, les Apple Reference Library seront enfin intégrées dans l'environnement ! Côté compilateur, on retrouvera gcc 3.5. On bénéficiera aussi de la fonction auto-vectorisation. Cela signifie que l'application tirera parti du Velocity Engine (moteurs présents dans le processeur pour le traitement vectoriel des images, audio et vidéo) sans écrire de code spécifique ! Le développement bénéficiera aussi d'une cure de jeunesse. On aura droit au support de Ant. Le portage de Java 1.5 est déjà bien engagé par Apple. Des pré versions sont disponibles depuis le printemps dernier. Dans Xcode 2.0, on pourra aussi utiliser du remote debugging.

Tiger Server plus agressif

La déclinaison serveur sortant en même temps aura, outre les nouveautés de la version desktop, de nouveaux modules bien pratiques. Ainsi, on disposera d'un outil de migration Windows NT vers MacOS X Server. La migration annoncée des serveurs NT attise les appétits de Microsoft, Linux et maintenant Apple. Avec la fin du support de ce système, les entreprises devront faire des choix à court et moyen terme. Il y aura aussi, en standard, l'outil Xgrid, permettant de mettre en place du calcul partagé (marché très courtisé par Apple avec une certaine réussite, tout comme pour le cluster). Apple implémentera aussi une version serveur de logiciel de vidéo-conférence, iChat. La meilleure prise en compte du 64 bits devrait améliorer très sensiblement les performances sur le Xserve G5.

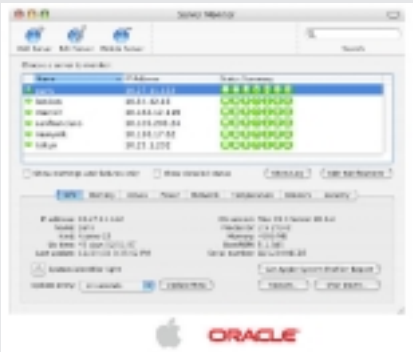
MacOS X : une véritable alternative ?

La réponse à apporter à cette question n'est guère évidente. La plate-forme Mac est aussi valable pour développer et déployer des sites web statiques et dynamiques, que Windows ou Linux. Si on utilise uniquement des technologies communes, guère de souci. Pour l'emploi de technologies plus spécifiques, cela se fera au cas par cas. Bref, certains types de développement sont possibles, mais là aussi, cela va dépendre du type d'application à créer. (voir tableau) Bref, sous MacOS X, on peut faire beaucoup de choses. Le portage d'outils libre / Open Source se développe énormément. Les principaux projets fonctionnent sous OS X. Le marché entreprise est sans doute le plus porteur pour le développement sur Mac. Il faudra observer de près l'évolution de MacOS X et des serveurs Xserve en entreprise. Je suis développeur indépendant, ou une SSII, ou encore un service informatique, est-il raisonnable et possible d'utiliser MacOS X comme poste de développement ? Généraliser à tous les postes, non, sans doute, sauf si vous travaillez pour le Web et le marché Mac. MacOS X peut représenter quelques postes de développement, notamment en Java, PHP ou Python, voire Mono dans les prochains mois. Avec le portage d'importantes solutions comme Oracle 10g, MacOS X peut aussi servir dans les grosses applications d'entreprise. MacOS X est comme Linux, adapté à certains développements, mais pas à d'autres. À

Type application	Faisabilité
Site web	Oui. Support des standards et des langages les plus courants
Application web	Oui. Plusieurs serveurs J2EE disponibles, ainsi que des outils de développement. Par contre, absence de tout IBM ou BEA
SGBD	Oui. L'offre est plutôt bonne et Oracle a annoncé Oracle 10g.
Application / serveur critique	Possible, selon l'application
Montée en charge, cluster, haute disponibilité	Oui. De plus en plus d'outils disponibles
Développement Java / J2EE	Oui, pas de gros soucis à avoir surtout du côté IDE libre
Développement C# / ASP.NET	Théoriquement oui avec Mono 1.0
Application multi-plate-forme	Oui, que ce soit avec Mono, Java ou plusieurs environnements Basic (ex. : REALBasic)
Intégration en environnement hétérogène	Oui, mêmes problèmes qu'avec les autres systèmes

Apple & Oracle : l'attente du marché

Si on savait qu'Oracle avait déjà flirté avec Apple par le passé, ces dernières années, la situation était loin d'être bonne. Excepté une version développement d'Oracle 9i, l'éditeur n'avait pas fait grand- chose pour la plate-forme Apple. Mais la nouvelle stratégie de la Pomme sur le marché serveur (gamme Xserve) et la disponibilité d'un vrai système serveur (MacOS X Server) ont radicalement changé la donne. " Apple propose des serveurs très compétitifs, compatibles avec Linux, où nous sommes leaders. Le Xserve G5 est une machine performante. " déclare Pascal Rawsin (marketing manager, Oracle Database). Cette nouvelle orientation d'Apple pour l'entreprise a incité Oracle à porter sa base de données Oracle 10g sous MacOS X. Si, de l'aveu même de l'éditeur, le marché est moins important en volume et donc en valeur, que sous Linux ou Windows, " le marché Mac est très captif et fidèle " précise Pascal Rawsin. La relance des outils Oracle sur plate-forme Mac vient de deux éléments : les contacts entre les deux dirigeants, Jobs et Ellison, et la demande des clients. Il existe en effet un besoin de gérer de grosses bases de données. Aujourd'hui, Oracle est le seul SGBD d'importance à être disponible sous MacOS X.



La version finale de la base de données est annoncée pour septembre. L'option phare d'Oracle 10g, la fonction cluster RAC, fera aussi partie du portage, ainsi que l'outil de développement Jdeveloper.

vous d'évaluer le système et les outils. Cependant, l'absence d'outils phares d'UML, RUP, MDA, SOA, de Web Services (des annonces sont attendues courant 2005) ou encore de debug et de tests peut constituer un problème fonctionnel. Et c'est bien la principale faiblesse de MacOS X. Peu d'outils de développements commerciaux fonctionnent sur Mac. La disponibilité de Java Studio Creator de Sun est un gage pour la plate-forme, mais c'est loin d'être suffisant. Tant que d'importants leaders comme Compuware, Telelogic, Parasoft, IBM Rational, ne porteront pas leurs environnements, MacOS X aura toujours une faible image, même si aujourd'hui, avoir un poste de développement Mac n'est nullement une tare. Notons aussi que les outils de développement Apple sont gratuits (environnement Xcode) et livrés en standard avec MacOS X. À la question de savoir si Mac est encore une alternative viable même face à Linux, Michel Sutter d'Apple France est direct : "aujourd'hui, la question ne se pose plus. La plate-forme Apple est appréciée, notamment par les développeurs Unix, même si nous n'avons pas beaucoup de contacts avec eux. On organise régulièrement en Europe des conférences techniques et des workshops où le développeur vient avec son code. Aujourd'hui, on voit de nouveaux développeurs arriver, ce fut sensible à la dernière WWDC. Ils ont de nouvelles attentes sur les environnements. On a vu l'arrivée d'Eclipse, une nette amélioration de Java. Metrowerks est une perte de vitesse, car Apple propose son propre environnement de développement (ndlr : Xcode)."

■ **F.T.**

4e Dimension 2004

En attendant la version nom de code Goldfinger qui ne devrait pas arriver avant un an, 4D sort en septembre la version 2004 de son SGDB. Pas de révolutions, mais des améliorations ici et là. Même si la prochaine grande évolution de la base de données n'arrivera pas avant longtemps, le projet Goldfinger va tout de même profiter au SGBD actuel, ainsi, dans la prochaine version, le moteur de Goldfinder sera repris. Cette version 2004 doit permettre à l'environnement de revenir aux fondamentaux du projet, en privilégiant la productivité. On peut voir une certaine rationalisation dans l'ergonomie et la présentation des menus et fonctions. Il est possible maintenant d'exécuter directement une méthode sans devoir tout lancer. On dispose aussi d'une génération de code pour certaines fonctions de la base de données. On a droit également à une bibliothèque de code réutilisable. Une des grosses critiques sur les listes a été largement supprimée. On dispose d'un objet liste bien plus souple et personnalisable, en évitant au maximum le code à créer. De plus, on bénéficie d'un objet, à mi-chemin entre une liste et les cellules tableur de 4D View. 4D a aussi décidé d'intégrer le module 4D Back-up pour pouvoir mettre en place des procédures de sauvegardes. Autre module intégré, 4D Custumizer. Pour faciliter la gestion des clients 4D en déploiement, on peut maintenant centraliser le versionning installé des clients 4D et procéder aux mises à jour automatiquement. Sur XML, 4D 2004 bénéficie d'un support amélioré, notamment avec la présence de DOM, SAX, Xpath et XSLT. ODBC a été directement intégré, mais le module ODBC externe demeure disponible pour une intégration plus poussée. Pour le Web Service, 4D peut se transformer en véritable serveur de Web Services. En un clic souris, on peut exposer en service web une méthode (on active ou non l'option WSDL).

Plongez tête la première dans l'univers Java !

Nouveau

Une méthode d'apprentissage révolutionnaire, amusante et incroyablement efficace pour apprendre Java

Écrit par deux célèbres gourous Java :
Kathy Sierra est formatrice de formateurs chez Sun Microsystems, et Bert Bates est membre de l'équipe de développement des certifications Sun.

Niveau : *Débutant en Java.*

Avec *Java Tête la première*, tous les moyens sont bons pour capter votre attention et la tenir en éveil. Voici quelques-uns des principes de son approche pédagogique :

- **Un style conversationnel** plutôt que formaliste : ne prêteriez-vous pas davantage d'attention à un dîner en bonne compagnie qu'à une conférence ?
- **Une approche visuelle :** une image se mémorise mieux que 1024 mots, et encore mieux lorsqu'elle est associée à du texte.
- **Des illustrations au ton décalé, surprenant et humoristique :** ne seriez-vous pas plus attentif aux principes clés de la programmation orientée objet s'ils étaient illustrés par une fille dans son bain ?
- **De nombreux exercices, jeux, énigmes, mots-croisés, puzzles :** pour fixer une fois pour toutes vos connaissances.

En vente en librairie

Pour recevoir notre catalogue papier, envoyez un email à info@editions-oreilly.fr en indiquant le nom de ce magazine.

L'INFORMATIQUE À LA SOURCE
O'REILLY®
www.oreilly.fr

Facile Avancé Expert

Dressez la JTable pour servir le café

SOURCES DE L'ARTICLE
WWW.PROGRAMMEZ.COM

Le composant Swing JTable est puissant et riche en interface et en fonctions, mais également complexe à utiliser. Apprenons à le manipuler.

Afficher des données sous la forme d'une grille bidimensionnelle est un travail que le programmeur rencontre fréquemment. Il peut s'agir de présenter des valeurs maintenues par des objets, eux-mêmes contenus dans une liste ou un vecteur, il peut s'agir d'afficher les résultats d'une requête à une base de données, ou encore de construire une feuille de calcul de tableur. Toutes les boîtes à outils modernes proposent un composant 'grille' déchargeant en grande partie de l'horrible et fastidieuse programmation de l'interface graphique. Java et Swing nous proposent un composant, la JTable, qui est peut-être plus complexe à appréhender que ses équivalents dans d'autres librairies, mais qui à l'usage se révèle remarquablement souple et puissant, quand on a saisi les bases de son maniement. Nous allons donc nous familiariser avec la JTable en passant en revue les applications les plus courantes.

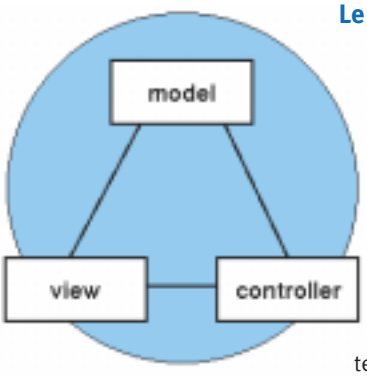


Fig. 1 : L'architecture Modèle Vue - Contrôleur.

Le modèle MVC

Swing applique le principe MVC (Modèle - Vue - Contrôleur) à ses composants (Fig 1). Ceci signifie que la gestion des données (modèle), leur représentation à l'écran (Vue) et l'interaction avec l'utilisateur (Contrôleur) sont découplées. Cette approche présente de nombreux avantages. Par exemple, sans rien changer aux données, on pourra représenter celle-ci sous la forme d'images, de textes, de camemberts en couleur, etc. En fait, sous Swing nous n'avons pas du MVC pur et dur, car l'affichage et la gestion de l'utilisateur sont couplés dans le composant, mais en revanche, le stockage des données est effectivement géré par un modèle qui est bien une classe séparée, ce qui est finalement le plus important.

Pour se faire la main

La représentation des données est encapsulée dans une classe abstraite AbstractTableModel, qui s'occupe de la mise en place des écouteurs d'événements et autres détails fastidieux. Cette classe est déclarée comme, implémentant l'interface TableModel, ce qu'elle ne fait pas totalement. Le programmeur est donc placé devant un choix : soit dériver AbstractTableModel et compléter l'implémentation de l'interface, soit utiliser une classe toute prête, telle que DefaultTableModel, qui fait ce travail pour lui. La première solution permet la gestion la plus fine, tandis que la deuxième solution est la plus immédiate. Nous commençons par cette dernière pour notre premier exemple. Le code partiel illustre

Fig. 2 : Une table fonctionnelle dans un composant JScrollPane.

ceci. Le lecteur voudra bien tenir compte du fait que les exemples présentés dans l'article sont tous partiels. Notamment, les parties générées automatiquement par l'outil de développement (ici JBuilder) ont été supprimées. Le lecteur voudra bien se reporter au site de Programmez pour obtenir les listings complets. Revenons à l'exemple. Le but du jeu est de présenter des données relatives à des pilotes de Formule 1, à savoir, nom, prénom, écurie et nombre de titres de champion du monde (la validité des données est sujette à caution :-). Nous voyons que pour utiliser un DefaultTableModel nous devons fournir deux tableaux. D'abord un tableau de chaînes pour les intitulés des en-têtes de colonnes, puis un tableau bidimensionnel d'objets pour les données. Enfin, il ne reste plus qu'à passer ces deux tableaux au constructeur du modèle, puis finalement passer le modèle au constructeur de la table, pour obtenir une table fonctionnelle (Fig 2). Pleinement fonctionnelle ? Pas sûr. Swing contient quelques bizarreries. En voici une concernant la JTable: si celle-ci n'est pas intégrée à un composant JScrollPane, les en-têtes de colonnes n'apparaissent pas à l'écran. Que faire alors si vous estimez que votre table de dimensions modestes ne justifie pas l'emploi d'un JScrollPane ? Obtenir les composants des en-têtes et les ajouter manuellement au cadre. Par exemple, en supposant que nous travaillons avec un BorderLayout nous faisons :

```
getContentPane().setLayout(borderLayout1);
getContentPane().add(table, BorderLayout.CENTER);
getContentPane().add(table.getTableHeader(),
BorderLayout.NORTH);
```

Le lecteur voudra bien se reporter au listing d'exemple JtableDemoo.java. Si vous employez un JScrollPane, peut-être vous demanderez vous pourquoi la barre de défilement horizontal refuse d'apparaître. Tout simplement, parce que par défaut une JTable redimensionne ses colonnes pour s'inscrire dans son conteneur. Pour assurer la venue de la barre horizontale, écrivez :

```
table.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
```

et ne manquez pas de faire quelques essais avec les différents modes de redimensionnement automatique. Pour en terminer avec le compor-

tement de base d'une table, remarquons qu'il est possible de déplacer une colonne avec le pointeur de souris. Ceci implique que l'ordre des colonnes n'est pas fiable et qu'en programmation plus avancée, il est impératif de s'assurer de la colonne concernée, ainsi que nous le verrons plus loin. Le modèle de données par défaut permet d'ajouter des lignes ou des colonnes (méthodes addRow et addColumn) et maintient les données dans un vecteur de vecteurs, ce qui n'est pas nécessairement la solution la plus économique en ressources, pour les tables de grande taille. En outre, son comportement est basique, ce qui nous fait deux bonnes raisons d'écrire notre modèle de données, personnalisé.

Un modèle de données, personnalisé

Nous partons du principe que notre grille ne peut croître horizontalement, du fait du type de données représentées dans notre exemple. Tout ce que nous voulons est pouvoir ajouter ou retirer des lignes. Pour cela, un vecteur est suffisant. Nous commençons par écrire une classe 'Pilote' qui conservera implicitement les données d'une ligne (Encadré 2). Nous remarquons deux choses. D'abord, une donnée (victoires) est maintenue dans un entier. Un modèle de données par défaut, parce que travaillant avec un vecteur de vecteurs aurait converti notre entier en un objet. Nous réalisons donc une économie de ressources, petite économie dans le cas présent, mais qui peut s'avérer substantielle avec de gros volumes de données. Ensuite, comme la JTable elle-même ne connaît que les objets lors de l'affichage, nous donnons une méthode getData construisant un tableau d'objets à son intention, à la volée.
Nous arrivons au modèle de données proprement dit (Encadré 3). Nous dérivons de AbstractTableModel et nous complétons l'implémentation de l'interface TableModel, ce qui consiste à écrire trois méthodes publiques getColumnCount, getRowCount, getValueAt. Ces méthodes sont automatiquement invoquées par la JTable au fur et à mesure des besoins. Etant entendu que rien ne nous interdit de redéfinir d'autres méthodes si nous souhaitons affiner notre modèle. Mais avec ces 3 méthodes, nous pouvons déjà bien nous amuser :

- getColumnCount doit retourner le nombre de colonnes.
- getRowCount retourne le nombre de lignes. Mais après tout, rien ne nous empêche de donner une ligne de plus que la réalité. Nous pourrions alors avoir une dernière ligne "Total" à la manière d'une feuille de calcul et c'est là que nous voyons tout l'intérêt qu'il y a à créer un modèle personnalisé.
- getValueAt retourne un objet représentant une donnée dans une cellule de la grille. Dans notre exemple (c.f JTableDemo2.java) nous invoquons getData de la classe Pilote pour obtenir notre objet. Si nous construisions une feuille de calcul, c'est ici que l'on retournerait, pour la ligne supplémentaire ; la chaîne "Total" ou un Integer représentant le total, selon la cellule impliquée dans la ligne... Quant à nous, nous utilisons cette méthode pour afficher un rang d'enregistrement qui ne fait pas partie des données proprement dites.

Notre modèle comporte encore deux méthodes pour ajouter ou retirer un enregistrement. Le point intéressant est, dans ces méthodes, l'invocation de fireTableRowsInserted ou fireTableRowsDeleted, ce qui a pour effet d'informer la table, de la modification et provoque ainsi son rafraîchissement.
Le troisième exemple (JTableDemo3.java) enrichit notre modèle de données. Nous voulons pouvoir modifier les cellules de la colonne 'Titres' et aucune autre cellule. Nous redéfinissons pour cela la méthode isCellEditable, qui reçoit en argument les coordonnées de la cellule, sous la forme de nombres entiers, et retourne un booléen. A nouveau, comme mentionné plus haut, ces nombres ne correspondent pas à l'organisation des données dans le modèle, si l'utilisateur a déplacé une colonne. Nous devons donc chercher le nom de la colonne, ce qui nous conduit à redéfinir également getColumnName. Enfin lorsqu'une modification est faite, la méthode setValueAt sera automatiquement invoquée par la table. Nous devons donc redéfinir cette méthode, en prenant garde qu'elle reçoive un objet en argument que nous convertissons en un entier. Le code est trivial et n'est pas donné ici.

Trier une table

C'est une opération courante lorsque l'on travaille avec une table. Bien sûr, on peut trier les données dans le modèle et informer ensuite la table du changement. Mais il est des cas où il n'est pas souhaitable, voire pas

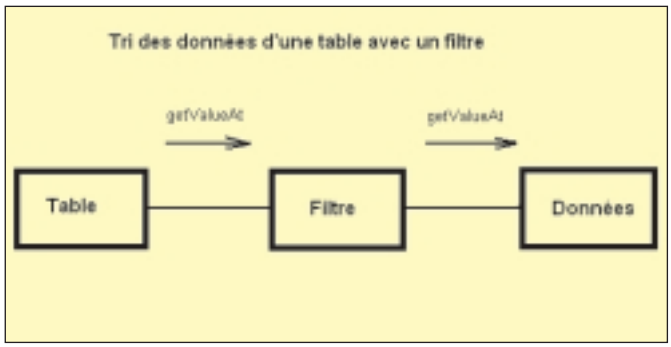


Fig. 3 : Un modèle de données intermédiaire permet d'afficher des données triées sans toucher aux données initiales.

possible, de procéder ainsi. On utilise alors une technique de filtre. Cela consiste à intercaler un autre modèle de données entre la table et le modèle de données initial (Fig 3). Dans notre exemple (JTableDemo4.java) nous voulons qu'un tri se déclenche, lorsque l'utilisateur clique sur un en-tête de colonne. La nature du tri effectué (ordre alphabétique, ordre de grandeur) dépendra de la colonne. La première chose à faire est de doter le modèle de données intermédiaire d'un écouteur d'événement de souris sur la table. Venons-en au tri proprement dit. Partisans du moindre effort, nous préférons utiliser l'algorithme de tri tout prêt de java.util.Arrays. Pour cela, nous définissons une classe interne encapsulant la notion de ligne (Encadré 4) et implémentant l'interface Comparable. Le seul membre de la classe est le rang de la ligne dans les données réelles. Dans le constructeur, nous remplissons un tableau de lignes simulant les lignes réelles. C'est ce tableau qui sera trié sur la base des données réelles. L'implémentation de Comparable invoque la méthode getValueAt du modèle de données réelles (on suppose que celles-ci sont également "Comparable" cela tombe sous le sens). La comparaison est effectuée sur les données réelles, mais c'est la ligne fictive qui est positionnée dans le tableau en conséquence. L'opération effectuée, on invoque fireTableDataChanged pour rafraîchir l'affichage qui se fait en demandant au modèle de données initial les données dans l'ordre des lignes du tableau trié et le tour est joué. L'exemple ne traite pas le cas où des données sont ajoutées ou retirées, afin de ne pas alourdir le code, mais le traitement de ce cas ne présente pas de diffi-

Fig. 4 : Personnalisation de l'affichage des cellules.

cultés. Le reste des méthodes du modèle de filtre se contente d'invoquer les méthodes du modèle de données initial.

Un rendu de cellule personnalisé

Une JTable n'est pas une simple grille remplie de texte. C'est plutôt un véritable conteneur. Ainsi, chaque cellule de la grille est un emplacement contenant un composant Swing. Par défaut, le composant est un JLabel. Swing offre la possibilité d'attribuer un afficheur de cellules à une table. Cette fonctionnalité donne une puissance extraordinaire à la table, moyennant très peu de travail. En effet si, dans notre exemple (JTableDemo5.java) nous nous limitons à créer des labels personnalisés (Fig 4) pour rendre la table plus lisible, rien n'interdit de placer des composants plus complexes dans les cellules. Par exemple, des boîtes à cocher ou des boîtes listes déroulantes. Tout ce qu'il y a à faire, comme on le voit ci-contre (Encadré 5) est d'implémenter l'interface TableCellRenderer. Pratiquement, cela consiste à écrire une méthode getTableCellRendererComponent qui va instancier un composant de notre choix à la volée, à la demande de la table.

Nous sommes loin d'avoir fait le tour du composant JTable. Celui-ci emploie encore d'autres modèles, comme le modèle de colonne qui maintient des objets TableColumn, encapsulant le comportement d'une colonne et permettant d'en définir la taille, autoriser le redimensionnement, etc. Toutefois, en se basant sur les principes et exemples donnés dans cet article, le lecteur ne devrait pas éprouver de difficultés à aller plus loin dans le maniement de la JTable.

Encadré 1

Une JTable avec un modèle de données par défaut

```
public class JTableDemo1 extends JFrame {

  private String[] colonnes = {
      "Nom", "Prénom", "Ecurie", "Titres"
  };

  private Object[][] cellules = {
      {"Senna", "Ayrton", "Mac Laren", new Integer(3)},
      {"Alesi", "Jean", "Prost GP", new Integer(0)},
      {"Schumacher", "Mikael", "Ferrari", new Integer(6)},
      {"Schumacher", "Ralf", "Williams", new Integer(0)},
      {"Barichello", "Rubens", "Ferrari", new Integer(0)},
      {"Prost", "Alain", "Williams", new Integer(4)}
  };

  JScrollPane jScrollPane1 = new JScrollPane();
  JTable jTable1 = new JTable(cellules, colonnes);

  public JTableDemo1()
          // Installer les composants dans le cadre
  }

          public static void main(String[] args) throws HeadlessException {
    JTableDemo1 jtd = new JTableDemo1();
          //etc
  }
```

Encadré 2

La classe Pilote que manipulera notre modèle de données

```
class Pilote {

  private String nom;
  private String prenom;
  private String ecurie;
  private int victoires;

  Pilote(String n, String p, String e, int v) {
    nom = n;
    prenom = p;
    ecurie = e;
    victoires = v;
  }

  public void setVictoires(int v) {
    victoires = v;
  }

  public Object[] getData() {
      Object[] data = {nom, prenom, ecurie, new Integer(victoires)};
      return data;
  }
}
```

Encadré 3

Notre modèle de données

```
class PiloteModel extends AbstractTableModel {
  private static final String[] titres = {
      "N°", "Nom", "Prénom", "Ecurie", "Titres"
  };
  private Vector pilotes;

  PiloteModel() {
    pilotes = new Vector();
    Pilote p = new Pilote("Senna", "Ayrton", "Mac laren", 3);
    pilotes.add(p);
```

```
    // etc, construction pour tous les pilotes.
  }

  public Object getValueAt(int ligne, int colonne) {
    if(getColumnName(colonne).equals(titres[0]))
      return new Integer(ligne+1);
    else
      return ((Pilote)pilotes.elementAt(ligne)).getData()[colonne-1];
  }

  public int getColumnCount() {
    return titres.length;
  }

  public int getRowCount() {
    return pilotes.size();
  }

  public String getColumnName(int colonne)
  {
      return titres[colonne];
  }

  public void ajouterPilote () {
    pilotes.add(new Pilote("Montoya", "Pablo", "Williams", 0));
    fireTableRowsInserted(getRowCount(),getRowCount());
  }

  public void retirerPilote() {
    pilotes.removeElementAt(6);
    fireTableRowsDeleted(6,6);
  }
}
```

Encadré 4

Une classe pour trier les lignes de données

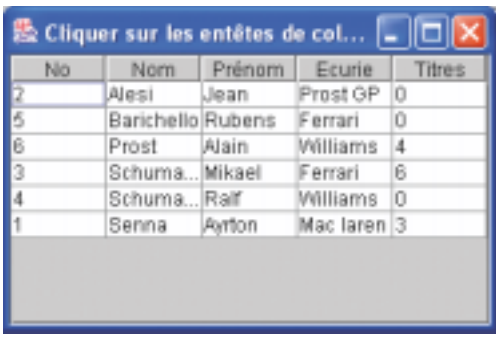
```
private class Ligne implements Comparable {
  private int rang;
  public int compareTo(Object l) {
    Ligne ligne_courante = (Ligne)l;
    Object a = pm.getValueAt(rang, colonne_a_trier);
    Object b = pm.getValueAt(ligne_courante.rang, colonne_a_trier);
    // On suppose que tous les objets sont "comparables"
    if(getColumnName(colonne_a_trier).equals(PiloteModel1.titres[4])) {
      return ((Comparable)b).compareTo(a);
    }
    else
      return ((Comparable)a).compareTo(b);
  }
}
```

Encadré 5

```
class MonTableCellRenderer implements TableCellRenderer {
  public Component getTableCellRendererComponent(JTable table,
      Object value, boolean isSelected, boolean hasFocus, int ligne,
int colonne) {
    if(table.getColumnName(colonne).equals(PiloteModel1.titres[0])) {
      label.setHorizontalAlignment(SwingConstants.CENTER);
    }

    // JLable est transparent par défaut
    label.setOpaque(true);

    if((ligne % 2) == 0)
      label.setBackground(Color.cyan);
    else
      label.setBackground(Color.pink);

    label.setText(value.toString());

    return label;
  }

  private JLabel label = new JLabel();
}
```

Notre modèle de données sait ajouter ou retirer des enregistrements.

Curieusement, les en-têtes de colonnes disparaissent si la table n'est pas incorporée à un JScrollPane.



Exemple JTableDemo4.java. Cliquez sur une colonne pour trier la grille. Ici les noms sont triés par ordre alphabétique.

Frédéric Mazué
fmazue@programmez.com

Groupe Optium :

SSII

conseil, qualification, formation

SSII de 300 personnes, le groupe Optium développe trois activités : le conseil et l'ingénierie des systèmes d'information, métiers classiques pour une SSII, mais également la qualification logicielle, un secteur en très fort développement, et dont Optium a fait son fer de lance, explique Thierry Boccara, directeur général du groupe.

Créé en 1993, Optium s'est développé par croissance organique jusqu'en 2000, avant de réaliser plusieurs acquisitions, dont celle de Métris, dont le nom a été conservé pour l'activité conseil.
Groupe Optium a réalisé l'an dernier un chiffre d'affaires de 23 M€, pour un effectif de 300 personnes, essentiellement basé en région Parisienne, avec deux agences récentes à Strasbourg et Nantes. Le pôle conseil, Metris, compte une soixantaine de personnes et génère environ 30 % du chiffre d'affaires.

L'activité qualification logicielle emploie 120 personnes et représente 40 à 45 % du chiffre d'affaires, et l'ingénierie, un peu plus d'une centaine de personnes pour 30 à 35 % du chiffre d'affaires.

Thierry Boccara explique qu'Optium s'est positionné sur la qualification logicielle il y a cinq ans déjà, alors que la Tierce Recette Applicative n'était pas encore très médiatisée... il indique en effet n'avoir reçu un premier appel d'offres, concernant la TRA, qu'en juin 2003... *" La progicialisation du marché nécessite de s'organiser pour intégrer des briques de plus en plus structurantes "* souligne-t-il. D'où d'importants besoins de formation et de méthodologie. C'est là qu'intervient Optium.

La qualification logicielle dès 1996

L'origine de l'activité de qualification logicielle remonte à 1996, époque où Optium utilisait l'AGL NSDK de Natsystem et a découvert les outils de test. Quelques partenariats plus tard, avec Mercury notamment, la société s'est imposée comme spécialiste des outils et automates de tests. Et comme formateur et intégrateur de ces outils.
Plus tard, en entrant en relation avec les maîtrises d'ouvrage, la société a élevé son niveau d'intervention à la conduite du changement et la TRA.
Côté Ingénierie, Optium s'est largement réorienté sur les technologies java pour des projets métiers (CRM par exemple) ou liés aux RH. Optium dispose d'un département Web qui a développé une expertise, en collaboration avec Braillenet, sur l'accessibilité des sites, qui devient selon Thierry Boccara un enjeu politique. La société propose des prestations d'audit des sites pour valider leur accessibilité. Enfin sur la partie conseil, il s'agit essentiellement d'assistance à maîtrise d'ouvrage sur des projets de portail d'entreprise, ou encore du pilotage de projets. La société compte également un département dédié au conseil en matière de monétique, qui compte 25 personnes qui font de la maîtrise d'ouvrage et de la maîtrise d'œuvre.

Les métiers du testing

Selon Thierry Boccara, il y a de réelles possibilités de faire carrière dans le domaine spécifique de la qualification logicielle.
Le pôle Qualification logicielle compte plusieurs types de profils : les experts outils et les analystes qui bâtissent les plans de test et les exécutent, ce qui suppose des compétences spécifiques au testing, mais également des compétences fonctionnelles. Optium propose également un cursus complet de formation, depuis la méthodologie jusqu'à la mise en œuvre, et ne compte pas moins de 45 jours de stage pour 12 sessions de formation différentes ! Les outils sont ceux concernant les tests unitaires (Compuware Dev, par exemple), les tests fonctionnels (Mercury testDirector, Winrunner, Compuware QA-Director, QA-Run, Princeton) et les tests de charge (Mercury Loadrunner, Compuware Q-Load). S'y ajoutent les outils de gestion de configurations logicielles, comme Continuus (Telelogic), PVCS (Serena), ou encore Clearcase.
À terme, Optium souhaite avoir 300 personnes dédiées à la qualification logicielle ; Les recrutements se feront en fonction de la montée en charge de cette activité, qui semble toutefois très rapide actuellement. Thierry Boccara parle " d'explosion de la qualification logicielle dans toutes ses composantes ".
www.groupeoptium.com

■ Carole Pitras

Abonnez-vous en ligne !
www.programmez.com/abonnement

JAVA La simplicité dans la complexité

Java compte 4 millions de développeurs enregistrés sur Java.net ! Une expo JavaOne très " technique ", avec l'annonce de J2SE 5.0 (et oui, adieu J2SE 1.5) et surtout de Java Studio Creator, représentait l'occasion de se pencher sur Java, son environnement, son évolution, les futurs marchés prometteurs, etc.
Java est désormais présent partout : le marché potentiel véritable de la plate-forme réside moins dans l'informatique en tant que telle, que dans les terminaux mobiles, les décodeurs, les objets intelligents et demain l'automobile (surtout dans le service), où Sun affiche son ambition. Et là, les besoins en développeurs experts seront considérables.

Java a certes convaincu l'entreprise et notamment le serveur avec J2EE, mais le chemin est loin d'être terminé. Il doit à la fois se simplifier dans son développement et son approche, tout en intégrant toujours plus de fonctions et de technologies.
Le défi pour les développeurs, et surtout les éditeurs, est immense. Le retard sur la plate-forme .NET parait évident.

Il y a là une sorte de paradoxe Java depuis quelques mois. On tente de simplifier le haut, pour complexifier toujours plus le bas. La disponibilité de Java Studio Creator le démontre, faisant en quelque sorte un " Java pour les nuls " à la sauce VB, pour attirer encore plus de développeurs.
NetBeans 4.0 correspond aussi à cette démarche.
Mais il est maintenant clair que .NET aura sa place aux côtés de J2EE, et que C# aura la sienne auprès de Java.
Une autre tendance est la généralisation des frameworks et des solutions libres, bousculant ainsi les habitudes, et ce n'est pas forcément un mal...

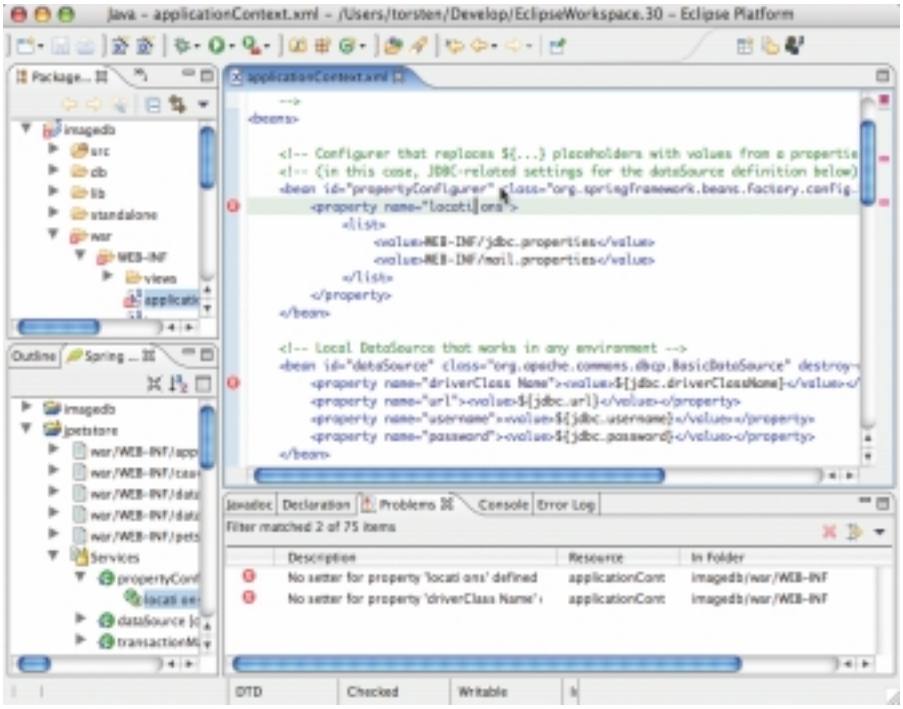
Pour ce dossier de rentrée, nous avons voulu nous focaliser sur plusieurs problématiques qui nous paraissent importantes : les tendances et nouveautés de Java, le marché des terminaux mobiles et des décodeurs numériques, la technologie JSF ainsi que les " mobilettes ". Avec un soupçon de prospective pour tenter de savoir ce qui va se passer dans les mois et années à venir.

■ François Tonic

Java - J2EE : évolutions et marchés

J2EE est plus qu'une réalité et Java reste incontournable dans nombre de projets. Mais l'avenir est-il pour autant certain ? Pour continuer à pénétrer le marché, Java est condamné à évoluer et J2EE doit pouvoir répondre à .NET. 2005 constituera une étape importante.

Un des axes de travail autour de J2EE consiste à simplifier sa mise en œuvre et son développement. Car, plus on ajoute de couches, de fonctions et de technologies, plus Java et J2EE se complexifient et nécessitent des développeurs aguerris et experts. Pour cacher cela, et faire en quelque sorte du Java pour les nuls, il faut disposer d'outils adaptés pour rendre la machinerie plus transparente. Les vives critiques autour de la trop grande présence de Sun dans l'évolution de Java ont été récemment prises en compte dans la nouvelle version du Java Community Process afin de rendre tout le processus plus transparent. D'autre part, il faut aussi en finir avec la guerre entretenue entre Java et Microsoft. L'accord entre Sun et Microsoft va sans aucun doute dans le bon sens, reste à le voir concrètement.



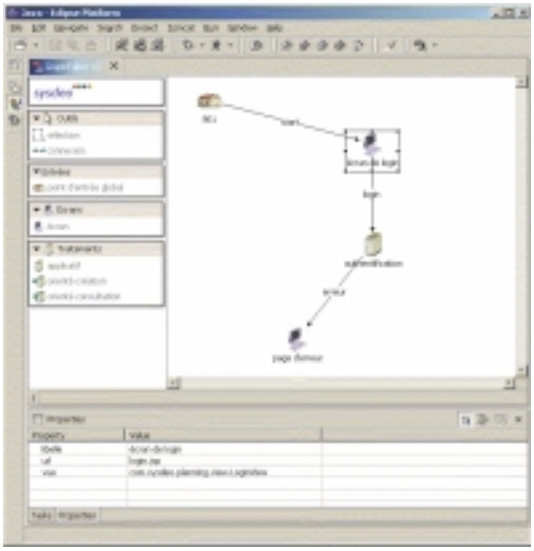
Serveurs d'application & SOA

L'orienté service sera-t-il une possibilité d'évolution de J2EE pour s'étendre encore plus en entreprise ? En tout cas, il s'agit pour les grands éditeurs J2EE de conserver une avance fonctionnelle sur les outils libres / Open Source. Car, les serveurs J2EE commerciaux souffrent réellement de la concurrence du monde libre, tandis qu'IBM semble s'installer fermement à la première place des serveurs J2EE commerciaux, et que BEA prend des parts significatives de ce marché. D'ailleurs, certains éditeurs, comme IBM, CA ou Novell, tentent aussi de profiter de l'Open Source. BEA a, pour sa part, lancé le projet Beehive, le framework de développement de WebLogic Workshop, avec une dose de SOA. Comme le notent les analystes, cette stratégie est à double tranchant. Oui, cela peut aider BEA à asseoir son marché et à trouver de nouveaux clients. Non, car, le développeur ou l'entreprise pourraient parfaitement délaisser WebLogic pour un serveur bien moins cher... On verra aussi apparaître de plus en plus de certifications des applications Java et non plus uniquement des serveurs. Avec le support du XML et des Web Service, le SOA en Java est une réalité exploitable dès maintenant. À cela, on pourra aussi ajouter depuis quelques mois une poussée (relative) du concept de programmation par aspect (ou AOP). L'AOP doit permettre d'éviter au maximum la redondance (donc la répétition du code). Il s'agit donc de factoriser. Bien entendu cela n'est pas la recette miracle, mais une technique de plus à utiliser dans les projets. Aujourd'hui, les performances se sont nettement améliorées, avec les nouveaux compilateurs JIT ou encore l'utilisation de HotSpot, mais si le code est mal conçu, il sera toujours lent. Même si aujourd'hui, les outils AOP sont rares, on s'attend à une disponibilité assez rapide.

.NET vraie menace ou partenaire ?

La plate-forme de Microsoft inquiète, intrigue, intéresse. On a vu depuis plusieurs mois, des annonces très fortes autour de .NET : SAP, IBM, Oracle... J2EE et .NET sont chacun une alternative à l'autre. Nous pensons que la montée en puissance de .NET cette année, et surtout à partir de 2005, va contribuer à équilibrer le marché et le poids de chaque plate-forme dans les projets. La sortie de Mono offre de plus un autre langage multi plate forme, plutôt performant. Il sera très intéressant de suivre les réactions autour de Mono. Ce sont deux plates-formes "ennemies", et pour autant, elles sont condamnées à être de plus en plus interopérables et compatibles. L'accord passé en Sun et Microsoft va dans ce sens, même si pour le moment l'accord reste très largement un flou artistique. On

verra sans aucun doute dans les mois à venir des solutions de passerelles bien plus intégrées qu'aujourd'hui même si une solution de type Janeva (Borland) est parfaitement fonctionnelle. Et cela est une nécessité absolue. J2EE / Java souffre de deux points noirs : le client lourd – riche et la complexité. Sur le client riche, J2EE ne propose pas une offre aussi intégrée que .NET. Utiliser Swing représente un défi dans les performances et le développement, même si aujourd'hui, Swing est bien plus performant que par le passé. L'avantage d'un .NET est de proposer un seul modèle de développement aussi bien en WinForm qu'en WebForm. En Java, on peut réaliser la même chose, mais on utilise plusieurs outils et frameworks. Dans les applications Web de type site, l'arrivée de Java Studio Creator peut enfin laisser espérer une meilleure diffusion du JSP sur les sites web, même comparé à PHP5 ou ASP.NET. Avec Java Studio Creator, on dispose enfin d'un bon outil, rapide d'utilisation ! Cette simplification du développement est un des grands défis des éditeurs pour l'avenir proche et lointain !



Le framework technique, métier ou de développement : l'avenir ?

En quelques mois, nous avons vu l'arrivée de plusieurs frameworks métiers de développement, notamment ceux de Sysdeo et Ideo. Tout d'abord, le framework, d'une manière générale, permet d'avoir une couche " d'abstraction " par rapport à la plomberie de bas niveau. Il cache en partie toute la complexité dans un contexte, des fonctions données. Le champ d'utilisation d'un framework est donc spécialisé et défini. Le framework doit permettre d'économiser du temps de développement et rendre cohérentes les différentes applications si le framework est généralisé. Cependant, le framework est productif quand celui-ci est parfaitement maîtrisé. Depuis longtemps, les frameworks J2EE se généralisent un peu partout. Mais, le choix du framework ne doit pas se faire à la va-vite. Il faut une évaluation sur l'impact de son utilisation dans le processus de développement, dans son intégration au niveau applicatif, dans les performances. Le framework en Java / J2EE ne peut que s'imposer. Il faudra aussi choisir entre du payant et du libre (très en vogue et de plus en plus nombreux). Revers de la médaille, le framework peut ne pas faciliter le portage du code Java sur plusieurs plates-formes ! Cette limitation est à prendre en compte dans son choix. Un framework Java / J2EE facilite aussi la mise en œuvre de méthodes de développement et d'architecture (SOA, orienté aspect, test unitaire, etc.). Depuis 2003, on annonce la mort de Struts, cela se concrétisera dans les prochains mois avec l'arrivée en force de JSF (notamment implémenté dans Java Studio Creator). Cela doit simplifier le développement Web, tout en faisant du MVC sans le savoir. Dans une autre idée, Java s'adaptera au 64 bits. On voit déjà apparaître des outils dédiés, notamment dans les JIT avec le cas de CACAO, un JIT 64. Les principaux éditeurs Java proposent ou proposeront rapidement des solutions adaptées.

Quelques frameworks J2EE disponibles

Borneo (Sysdeo)	Framework technique
SweetDev (Ideo Technologies)	Framework technique
Hivemind (Jakarta)	Framework orienté SOA
Cactus (Jakarta)	Framework de test côté serveur
BeeHive (BEA)	Framework de développement basé sur WebLogic Workshop
Application Development Framework (Oracle)	Framework de développement de Jdeveloper 10g d'Oracle
Spring	Framework technique
Struts (Apache)	Framework MVC, remplacé par JSF
Avalon (Apache)	Framework de développement
Hibernate	Framework de mapping O/R

Upgradez-vous !

À Programmez !, nous vous le répétons depuis plusieurs mois, formez-vous, mettez-vous à jour dans vos connaissances, apprenez les méthodes et les architectures. Pour qu'un développeur Java soit performant et efficace, la formation et la mise à niveau régulière sont INDISPENSABLES. Dans le cas d'un serveur J2EE, cela ne s'invente pas, il faut avoir de solides connaissances en la matière. Avec les grandes migrations des applications Cobol et ADA à venir, il faudra former des bataillons entiers de développeurs non formés à Java et à l'objet. Si vous avez décidé d'utiliser des frameworks, là aussi il faut être formé. Cela passe au moins par un ouvrage dédié et la pratique des exemples. ■ **F.T.**

Programmez! LE MAGAZINE DU DÉVELOPPEMENT ABONNEMENT :

CADEAU

VISUAL EXPERT 5.0 pour VB

Visual Expert est un outil d'aide au développement et à la maintenance des projets Visual Basic. Il génère la documentation technique, vérifie la qualité du code et aide à l'exploration du projet (quand le code est développé par quelqu'un d'autre...).
Visual Expert réalise également des analyses d'impact : vous pouvez lister tous les composants affectés par une modification du code.

Pour plus d'informations sur Visual Expert : http://www.visual-expert.com

Licence 1 an, 1 poste.
Envoi électronique, par mail.
Image non contractuelle

Un marché de 17 milliards d'objets intelligents sous Java ?

Qui dit Java / J2EE, dit Sun. Pour un point sur Sun et sa plate-forme fétiche, nous avons rencontré Emmanuel Obadia (Director, Software Product Marketing - Global Sales Organization) et Éric Mahé (Technology Advisor - Sun Microsystems).

Programmez ! : A combien évaluez-vous la communauté de développeurs Java et quels seraient les marchés les plus prometteurs ou en forte croissance ?

Emmanuel Obadia : Il y a 4 à 4,5 millions de développeurs Java. Java est fort partout. On vise une communauté de 10 millions de développeurs. Les développeurs vont sans doute être amenés à travailler dans les mondes Java et .NET. On veut toucher le développeur en entreprise, chez un éditeur, dans une SSII... Choisir Java, c'est s'affranchir de la plate-forme. Côté marché, j'en vois 2 : le premier, le téléphone portable, avec un milliard d'unités livré avec Java en 2007-08. Ce marché est déjà lancé. Le marché de masse suivant est l'automobile.

Eric Mahé : avec un seul code, on peut générer plusieurs canaux. On voit une convergence entre le PDA et le téléphone. La demande fait que la plate-forme MIDP est trop limitée, d'ou le projet Tsunami, qui est de fournir une plate-forme quasiment J2SE d'ici un an environ.

P ! : Le téléphone portable est donc un marché fort et lancé. Comment Java est-il utilisé ? Dans les jeux ?

EO : On réfléchit aux marchés pour les utilisateurs. Les opérateurs réalisent 3 milliards de dollars sur les sonneries et images. Plus un téléphone peut se personnaliser, plus l'opérateur peut faire du chiffre d'affaires, ouvrir de nouveaux services. IDC en 2012 prévoit 17 milliards d'objets intelligents actifs ! Notre marché cible est là, les 17 milliards d'objets ! Aujourd'hui, le marché le plus important est celui des cartes à puce.

EM : Il faut séparer le système et la JVM. Le marché des fournisseurs de système est phénoménal. Tout le monde a une licence Java, excepté Microsoft.

EO : Les jeux représentent environ 90 % des applications Java mobiles, un peu moins en France. Au Japon, le chating commence à dépasser le jeu ! Le jeu n'est pas anecdotique. On disait Java pas graphique, pas performant. On supprime ces deux arguments. Un développeur de 20 ans peut trouver des défauts à Java. Les développeurs sur carte SIM par exemple sont des développeurs pro et aguerris. Là, il n'y a que du Java. Dans le marché de masse, la sécurité est primordiale.

P ! : Avec la sortie de Java Studio Creator, Sun propose enfin un outil RAD pour faire du JSP simplement. Est-il là pour concurrencer PHP, ASP.NET ? Qu'en est-il des liens Microsoft – Sun ?

EO : Java Studio Creator change la donne, c'est l'objectif. L'outil est là pour draguer les développeurs VB. Lors de la dernière JavaOne, pour la première fois me semble-t-il, Microsoft était là. Nous avons montré une utilisation de l'interface native d'une application Java sous le futur Windows, Longhorn. N'oublions pas Java Studio Enterprise qui permet de repackager des applications C++ en applications Java.

P ! : Sun dispose aussi d'un autre IDE Java, NetBeans, mais jusqu'à présent, on n'a pas l'impression que vous l'aviez mis réellement en avant, tandis qu'Eclipse occupe largement le marché et s'impose comme un leader, voire le leader des IDE Java.

EM : La communication autour de NetBeans change depuis quelques mois. Creator est basé sur lui. La version 4 intègre J2SE 5.0 (ndlr : anciennement 1.5), propose aussi un pont vers J2EE, en option jusqu'à présent, un RAD, etc.

P ! : depuis quelques mois, un débat se fait jour autour d'EJB 3.0 et de la prise en compte de JDO comme modèle de persistance. Faux débat ou réelle opposition ?

EM : sur la " bataille " autour de JDO, ce sont des querelles de clocher. Au niveau de la spec, il ne s'agit pas de JDO, mais de celle d'EJB. Qui va gagner ? À voir. On choisira ce qu'il y a de mieux. Quoi qu'il en soit, JDO continue à évoluer.

■ **F.T.**

OFFRES LIMITÉES

Coupon d'abonnement p. 79

Recevez un livre !

Choisissez un des 4 titres indispensables, sélectionnés dans la collection Ressources Informatiques des Editions ENI.

Ces 4 livres de références, de plus de 360 pages en moyenne, ont un prix de vente de 27,14 €.

Réservé aux 250 premiers demandeurs

-50%

soit 1 an d'abonnement + 1 livre

46 € + 8€ frais d'envois

Recettes pour améliorer les performances et la qualité

Pour faire le point sur les performances Java et les lacunes du langage, nous avons posé quelques questions à Thomas Gil, consultant senior à Valtech, et codéveloppeur des projets Oxerp et AspectDNG.

Programmez ! : La comparaison des performances de Java / J2EE et de C# / .NET est-elle possible ? Sur quels critères ?

Thomas Gil : Faire des comparatifs, comme l'ont montré www.theserverside.com et nous-mêmes sur www.dotnetguru.org, c'est techniquement possible. La difficulté réside dans le choix des tests et dans l'interprétation des résultats. Plus concrètement, DotNetGuru avait choisi de comparer des éléments très simples, comme des parcours de listes, de tableaux... On pourrait donc avoir l'impression que les tests sont objectifs. TheServerSide avait choisi un benchmark ensembliste qui couvre une architecture multicouche : le PetShop contre le PetStore. Pour être clair, ces deux plates-formes sont quasiment impossibles à mettre en concurrence. Il faudrait comparer une application .NET à une application Java. Cependant, les applications .NET tournent sur un CLR et Java sur un JRE. Il faut choisir le bon CLR et JRE pour ne pas invalider les résultats. De plus, il semble logique que le CLR de Microsoft sur Windows soit plus intégré à l'OS qu'un JRE. Il faudrait donc refaire les tests sur le CLR de Mono sous Linux, et le comparer aux JRE de BEA/SUN/IBM/Blackdown... Quel travail de titan ! pour un résultat fatalement contestable.

P ! : Selon votre expérience, Java / J2EE doit-il encore améliorer sa disponibilité, la montée en charge et la qualité de ses performances ?

TG : Absolument. Il y a de la réserve de puissance dans les JRE. Plusieurs techniques pourraient être mises en oeuvre pour améliorer les performances globales de la machine virtuelle telles que la gestion d'un recyclage transparent et proactif des objets (je parle de ceux qui ne sont pas gérés dans un pool quelconque). D'autre part, côté plate-forme, J2EE peut être amélioré sur de nombreux aspects. Les EJB Entité, en standard, nécessitent toujours l'exécution de N+1 requêtes SQL, pour charger un ensemble d'objets en mémoire depuis une base de données relationnelle. Les améliorations nécessaires sont connues, et apparaissent en ce moment même dans les drafts de la version 3.0 des EJB. Il faut également distinguer J2EE, la spécification, et les serveurs d'application qui implémentent cette spec. Eux aussi peuvent être tunés, optimisés. Ne serait-ce que l'un des moteurs de JSP les plus en vue dans le monde OpenSource, celui de Jakarta-Tomcat : on se plaint des performances des Servlets générées à partir de JSPs par Tomcat ? Il suffit d'en lire le code source pour comprendre que certaines techniques d'optimisation bien connues n'ont pas toujours été mises en oeuvre. Sur l'accès aux données, nous nous évertuons à mapper les objets sur des bases relationnelles. Certains mécanismes tels que la prévalence permettraient d'accéder à d'autres niveaux de disponibilité et de performances, mais sommes-nous prêts à remettre en question les architectures actuelles et la sacro-sainte place de la base de données relationnelle dans nos systèmes d'information ? Cette remarque dépasse J2EE, et s'applique à toute plateforme de développement à objectif généraliste.

P ! : Quels sont les conseils que vous pourriez donner à un développeur Java pour améliorer les performances ?

TG : Avant d'apprendre les astuces de bas niveau, comme écrire new HashMap(89) plutôt que d'invoquer le constructeur par défaut de HashMap, essayer de faire des boucles décroissantes, ou mettre final un peu partout. Il vaut mieux s'améliorer en conception Objet. Appliquer certains Design Patterns améliorera vos performances dans une mesure inaccessible aux mini-optimisations précédentes. De même, à un niveau plus haut, une bonne architecture offrira un meilleur potentiel de performances qu'une mauvaise architecture, mais dont chaque couche, même bien conçue, ne pourrait pas approcher. Donc, développeurs Java, étudiez les Patterns. Recyclez vos objets "usagés" au lieu de les "jeter à la poubelle" ramassée par le GarbageCollector. Pensez synchrone, pro-actif, concevez vos applications en gardant à l'esprit qu'elles devront peut-être supporter une montée en charge importante. Ensuite, vous pourrez étudier les paramètres de votre machine virtuelle (taille du tas, des frames de threads), tuner vos serveurs d'applications, implémenter les techniques d'optimisation de bas niveau.... Mais cela ne doit venir qu'en fin de développement.

■ F.T.

Sondage

Etes-vous satisfait du magazine ?

www.programmez.com

L'adoption de J2EE a été freinée

Avec les questions qui se posent autour de J2EE et de la forte présence des outils libres, nous avons sondé un leader du marché, BEA en interrogeant Pierre-Olivier Chotard (directeur marketing France de BEA).

Programmez ! : Depuis plusieurs années, si J2EE a su s'imposer dans l'ensemble, on a souvent critiqué sa complexité croissante. Les éditeurs cherchent-ils à en simplifier l'usage ?

Pierre-Oliver Chotard : Sur J2EE, notre approche était devenue un peu trop complexe par rapport à .NET de Microsoft. Nous avons sorti un IDE dans lequel l'approche de mise en œuvre a été simplifiée. Avec SOA, via le projet BeeHive, nous avons ajouté une couche supplémentaire à J2EE. L'adoption de J2EE a été freinée par son approche. Il faut une approche plus simple.

P ! : Pensez-vous que Sun va devenir un réel concurrent dans les outils de développement ?

P-O C : Le problème de Sun, est de vendre du logiciel. S'ils y arrivent, c'est tant mieux. Aujourd'hui, ils ont un bon serveur d'application, mais on ne les voit pas sur le marché.

P ! : Depuis pas mal de temps, dans le monde J2EE on parle beaucoup de serveurs et d'outils libre / Open Source. Quel est le point de vue de BEA ?

P-O C : On les prend en compte, en particulier en France. JOnAS, franchement, on ne le voit pas. JBoss, ils ont un bon marketing en France, mais sur les applications critiques, ce n'est pas encore ça. C'est une structure qui se met en place. L'Open Source est plus une vision technologique intéressante. JBoss, par exemple, est entre les deux : une approche Open Source et une approche commerciale. On attend de voir notamment dans les applications critiques.

P ! : J2EE continuera-t-il à s'imposer dans les entreprises ? Peut-il étendre son audience ?

P-O C : Aujourd'hui, la cible de BEA concerne avant tout les grands comptes. Sur ces clients-là, J2EE n'est pas encore généralisé. Beaucoup d'applications restent à porter. Ce sont des projets à long terme. Maintenant, J2EE se déploiera aussi dans les entreprises plus petites s'il se simplifie.

P ! : Les analystes voient dans l'avenir une sorte d'équilibre entre .NET et J2EE dans l'entreprise. Faites-vous le même constat ?

P-O C : actuellement, il n'existe pas d'autres technologies. On observera comment le marché va se comporter. Un DSI doit de toute manière regarder les deux solutions.

P ! : BEA envisage-t-il comme d'autres éditeurs de s'intégrer à .NET ? L'accord Sun – Microsoft améliorera les échanges entre J2EE et .NET ?

P-O C : Pour moi, je ne pense pas que l'on aura des plug-ins Workshop pour .NET. Mais on dispose déjà de composants permettant de communiquer avec l'environnement .NET, notamment avec des Web Services. De plus en plus, on parlera de compatibilité entre les deux plates-formes. L'accord est bon signe vis-à-vis des clients. Mais, attendons les aspects concrets.

■ **F.T.**

Java Server Faces, le RAD du Web selon Sun

JSF est l'aboutissement de 2 ans de travail pour apporter au monde J2EE, un framework orienté interface. Jusqu'à maintenant seule la technologie " .net " disposait avec les webforms d'un tel modèle événementiel reposant sur des composants graphiques. Le monde Java se devait de mettre à disposition un modèle orienté événements, pour permettre une amélioration de la productivité. En effet, les temps de développement de la couche présentation des applications MVC classiques (code html, code dynamique en JSP et contrôles JavaScript) représentent une part importante du coût final d'une application.

Qu'est-ce que JSF ?

JSF repose sur un modèle de composant pour développer des interfaces WEB. Ainsi, une application Web est vue comme un ensemble de composants et d'événements s'appliquant à ces derniers. Le développement d'interface WEB s'en trouve simplifié, notamment grâce à :

- L'utilisation de composants graphiques. JSF permet d'assembler des composants graphiques existants et offre une architecture pour en créer de nouveau.
- La gestion des événements. Le modèle JSF relie les événements générés du côté client à la logique applicative, de façon transparente.
- La gestion et la sauvegarde automatique des états des composants. Le mécanisme de gestion des données en session, permet de réafficher automatiquement les données saisies dans les composants. Ceci est très pratique en cas de retour arrière.
- La séparation de la couche logique et la couche de présentation. Le rendu d'un composant est externalisé hors du composant.

Cependant, les spécifications JSF n'ont pas été créées uniquement pour le développement d'application WEB. Il s'agit d'une architecture puissante et ouverte, pour afficher les composants sur différents médias, grâce à l'utilisation de " renderer ". En standard dans les implémentations de la spécification JSF, un " renderer " HTML 4.0 est fourni qui s'appuie sur la technologie JSP. Toutefois, JSF est conçu pour être indépendant des pages JSP. Une application JSF peut utiliser d'autres technologies pour interagir avec l'utilisateur et les composants graphiques peuvent être restitués sur différents médias. Il existe déjà sur le marché un " renderer " XML. Il est également possible ce créer ses propres " renderers".

Comment JSF fonctionne-t-il ?

JSF est constitué de deux parties :

- Une API qui permet la représentation des composants, la gestion de leur état, le traitement des événements, la validation des saisies, la description de la navigation entre les pages et l'internationalisation de l'application.
- Une librairie de tags JSP spécifiques permettant d'afficher les composants au sein des pages JSP

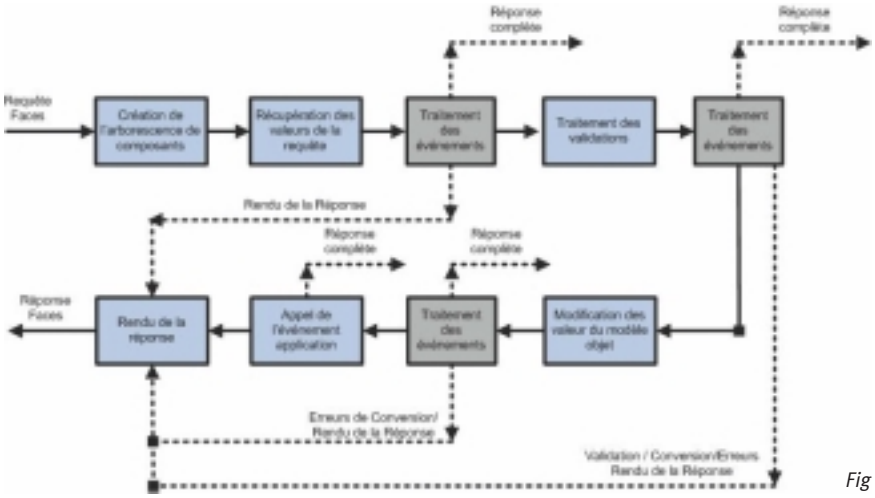


Fig 1

Le framework JSF propose également :

- un mécanisme de validation des champs de saisie très pratique. Des " validators " sont fournis en standard dans les implémentations : LengthValidator, DoubleRangeValidator et LongRangeValidator. Il est également possible de créer simplement ses propres composants de validation, il suffit d'implémenter l'interface " Validator " et d'écrire la méthode " validate ()" qui effectue le contrôle de validation.
- un mécanisme de conversion de valeur de champ qui s'intègre directement dans les composants. Les données sont ainsi transformées avant d'être fournies aux composants ou lors de la restitution de ces composants dans la page. En standard, de nombreux " Converters " sont fournis. De façon similaire au " Validator ", un développeur peut facilement créer de nouveaux composants de conversion. Pour cela il faut implémenter l'interface " Convertor " et écrire les deux méthodes : getAsObject( ) et getAsString().

Les composants de validation et de conversion sont associés simplement aux champs de saisie dans le code html en utilisant les tags JSF.

```
<h:inputText binding="#{MyPageBean.textField}" id="textField"
converter="#{MyPageBean.doubleConverter}"
validator="#{MyPageBean.doubleRangeValidator.validate}"/>
```

Puisque JSF est une application web, il est nécessaire de définir un fichier de description de déploiement : " web.xml ", pour mapper la servlet JSF " controller " avec l'url /faces/.

```
<servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>javax.faces.webapp.FacesServlet
  </servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
```

```
    <url-pattern>/faces/*</url-pattern>
  </servlet-mapping>
```

De plus, toutes les requêtes JSF suivent le cycle de traitement suivant afin de construire la réponse. Il se décompose en 6 phases distinctes. Les lignes en plein décrivent le cycle normal et les lignes en pointillés, le traitement des erreurs. (Fig; 1)

Phase 1 :

Création de l'arborescence de composants

À la réception d'une requête, la hiérarchie de composants de la page demandée est créée.

Phase 2 :

Récupération des valeurs de la requête

Les valeurs sont récupérées dans la requête et stockées dans les composants. La servlet "controller" parcourt la hiérarchie de composants et appelle systématiquement la méthode decode() des composants. De plus, des événements et des validators sont générés et stockés dans le contexte JSF.

Phase 3 :

Traitement des validations

Une phase de validation est effectuée. La méthode validate() est appelée pour tous les validators stockés dans la hiérarchie de composants associée à la requête.

Phase 4 :

Modification des valeurs du modèle objet

Les valeurs contenues dans les composants sont recopiées dans les objets métiers éventuellement associés.

Phase 5 :

Appel de l'événement application

L'événement de type "form" ou de type "command", correspondant à la demande d'une nouvelle page est traité.

Phase 6 :

Rendu de la réponse

Une hiérarchie de composants est créée, représentant la réponse.

Mise en œuvre de JSF

La construction d'une application JSF comprend les étapes suivantes :

- Concevoir et implémenter le modèle objet :

Ce dernier se compose de l'ensemble des JavaBeans représentant les données de l'application.

- Décrire le modèle objet dans JSF:

Le modèle objet est décrit dans le fichier de configuration " faces-config.xml ". Chaque composant doit être déclaré spécifiquement.

```
  <managed-bean>
    <managed-bean-name>myData</managed-bean-name>
    <managed-bean-class>
    com.myCompagny.myBean.myUserDataBean
    </managed-bean-class>
    <managed-bean-scope>session</managed-bean-scope>
  </managed-bean
```

Il est possible de définir 3 scopes différents pour les Beans : request, session et application. La mention " none " permet la simple instanciation d'un composant.

- Créer la représentation graphique de l'application :

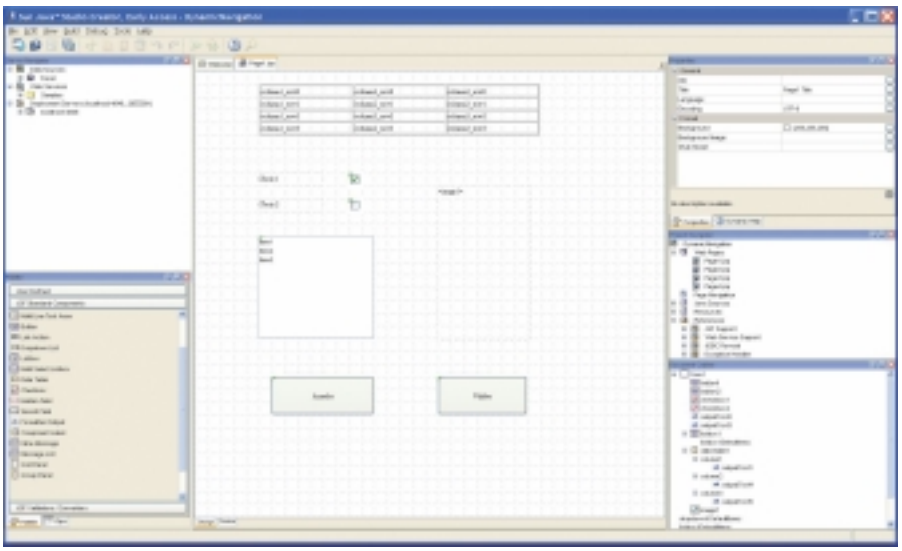


Fig. 2 - Exemple d'IDE : Java Studio Creator

Les JSPs sont le mode de représentation standard de JSF. Il existe deux types de librairies de tag :

- core permet la réalisation des conversions entre types, la capture les événements déclenchés par l'utilisateur et la validation des données saisies.
- html prend en charge la génération du code html représentant les composants JSF dans les pages.

- Définir la navigation :

La navigation entre les différentes pages est définie dans le fichier " faces-config.xml " par un ensemble de règles, dont un exemple est présenté ci-dessous.

```
  <navigation-rule>
   <from-view-id>/jsp/saisie.jsp</from-view-id>
    <navigation-case>
     <from-outcome>validation</from-outcome>
      <to-view-id>/jsp/resultat.jsp</to-view-id>
    </navigation-case>
  </navigation-rule>
```

Une règle est déclenchée si toutes les conditions sont remplies, si la page de provenance contenue dans le tag " ‹from-view-id› " correspond et si l'action contenue dans le tag "‹from-outcome›" est effectivement déclenchée.

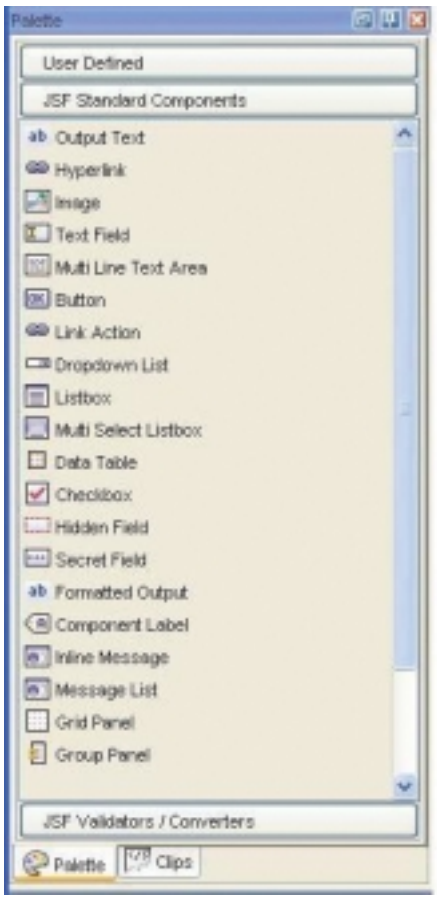
Quels sont les atouts de JSF ?

JSF est un standard J2EE, supporté par de nombreux éditeurs, à contrario des autres frameworks du monde J2EE.

Les principaux éditeurs proposent des environnements de développement JSF, notamment :

- IBM avec la version 5.1.2 de WSAD
- Sun avec son nouvel environnement de développement Java Studio Creator.
- La prochaine version de JBuilder devrait disposer de fonctionnalités relatives au JDK 1.5, de nouveaux concepteurs graphiques pour le développement de services Web, le support de Struts et des JavaServer Faces.

De plus, des plug-ins existent, permettant le développement JSF avec Eclipse.



Exemple d'une palette de composants.

Les nouveaux IDEs spécifiques RAD accélèrent encore le développement d'applications web. Il devient par exemple, possible avec quelques clics et quelques lignes de créer une application simple de saisie, de contrôle et de validation de formulaire. (Fig. 2)

Une offre de composants JSF, est en train de se mettre en place. En effet, la liste des composants fournis en standard est relativement pauvre, mais de plus en plus d'éditeurs proposent des bibliothèques de composants réutilisables. Il est également possible de développer ses propres composants. Cela permet d'augmenter considérablement le taux de réutilisation de la couche présentation d'une application WEB. De plus, la séparation du rendu de son composant permet de tester unitairement les composants comme une application Java indépendante, ce qui permet de développer des applications plus robustes.

Conclusion

JSF est une architecture puissante et ouverte, reposant sur un modèle événementiel. A la différence des frameworks actuels orientés navigation tels que Struts, JSF permet un niveau d'abstraction supérieur. En effet, il réduit fortement la complexité induite par le protocole http. Le développeur JSF manipule des composants graphiques, des événements et des données. Le développement d'une application web complexe s'en trouve donc fortement accéléré, surtout avec les nouveaux environnements de développement RAD.

JSF est une technologie encore jeune, mais des améliorations sont attendues. Notamment, la couche de présentation reposant sur la technologie JSP n'est pas idéale. Entre autres, il est impossible d'imbriquer des tags JSTL et JSF. La JSR pour la norme JSP 2.1 s'efforce de réduire les différences entre JSF et JSP. On peut toutefois se demander s'il n'est pas préférable d'intégrer dans JSF une couche de restitution spécifique.

L'existence de nombreuses applications développées avec Struts, soulève la question de la coexistence. On peut d'ores et déjà dire que leur superposition apparaît peu rentable et risque d'apporter plus d'inconvénients que d'avantages.

Liens utiles

http://java.sun.com/j2ee/javaserverfaces/

http://www.jsfcentral.com/

■ Fabien DOREAU

consultant chez Neoxia, cabinet de conseil en architecture des systèmes d'information.

fabien.doreau@neoxia.com

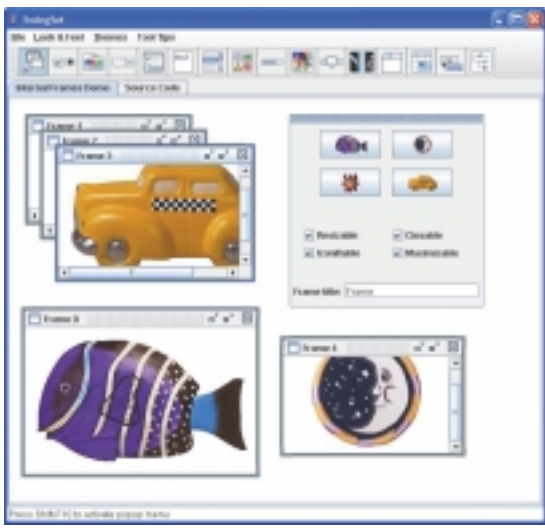
J2SE 1.5 (J2SE 5.0): le prochain SDK

J2SE 1.5, nom de code Tiger, est la prochaine distribution majeure de la plate-forme et du langage Java. Près de 15 composants tirés des JSR, avec une centaine d'autres améliorations développées pour le JCP seront de la partie. Tiger s'axe principalement sur la facilité de développement, avec de nouvelles fonctionnalités allant dans ce sens (metadata, autoboxing, types énumérés, imports statiques, etc.). Au-delà du développement, la 1.5 s'occupe aussi des performances, de la surveillance ou encore du management ainsi que le poste client.

Le langage java a subi quelques améliorations impliquant pour certaines instructions, un comportement différent ou étendu qui permet une meilleure compréhension et lisibilité. Notons qu'un des leitmotive des développeurs de la plate-forme Java est l'intégration minimale de nouveaux mots clés ou instructions. L'une des évolutions apportées au langage est l'utilisation des "Generics". Il s'agit d'un portage à la sauce Java des templates du C++. L'interface "Collection", implémentée dans toutes les classes de collection, suggère que les objets stockés peuvent être de différents types. Le développeur doit alors, lors du retrait d'un élément d'une collection, forcer celui-ci à un type spécifique (cast). Le " Generic ", connu aussi sous le nom de type paramétré, permet de spécifier le type d'une collection (String, Integer,...). Ainsi, une erreur sera levée à la compilation, lors de l'ajout à une liste d'un objet dont le type ne serait pas celui attendu. Cela élimine aussi les tâches fastidieuses de forçage de type (cast).

Exemple :

```
1)      // Définition de la liste, ici, Liste de String
2)      List<String> list = new ArrayList<String>();
3)      list.add(new String("Hello world!"));
```



```
4)    list.add(new String("Good bye!"));
            / / cette ligne provoque une erreur de
compilation
5)    list.add(new Integer(95));
6)
7)    / / Ecriture de la liste
8)    Iterator<String> i = c.iterator();
9)    while(i.hasNext()) {
10)                    String item = i.next(); /    /
notons l'absence de cast ici
11)       System.out.println("Item: "+item);
12)   }
```

L'utilisation des " generics " permet une vérification accrue des types lors de la compilation impliquant un débogage plus simple, un code plus lisible, plus compréhensible et simplifie considérablement les tâches de refactoring par exemple.

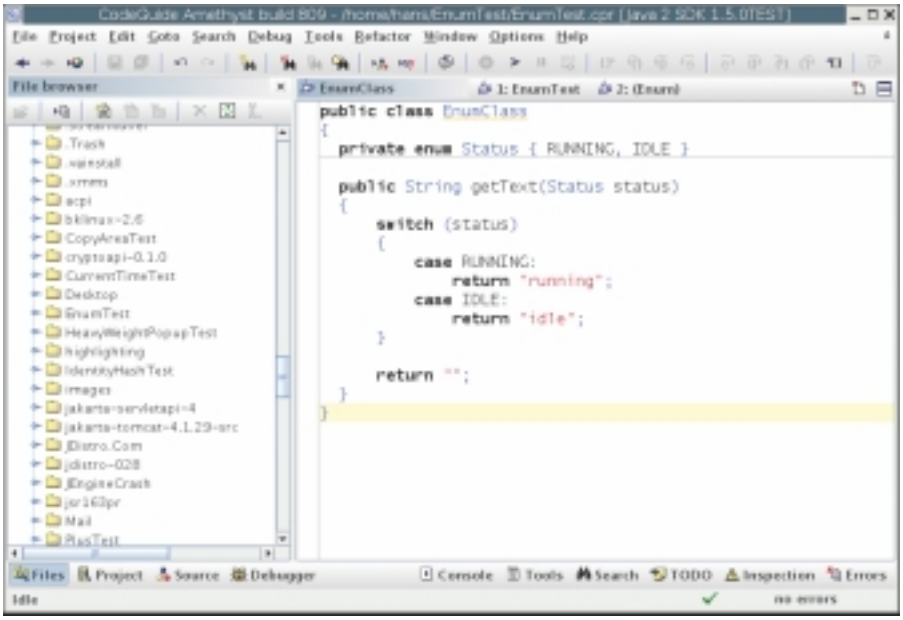
La conversion manuelle entre les types primitifs (tel " int, float, ... ") et les classes wrapper (telle que " Integer, Float, ... ") est nécessaire lors de l'insertion d'un type primitif dans une collection. La fonctionnalité de "Autoboxing/unboxing" élimine cette conversion manuelle. Ainsi, dans l'exemple, la ligne 5 se transforme en : " list.add(95); ". De même, la récupération de l'objet " Integer " stocké dans la liste peut s'effectuer de la sorte : " int total = list.get(2); ". Cependant, il convient de noter que la classe wrapper doit être utilisée comme un type " generic " : List‹Integer› list = new ArrayList‹Integer›();.

L'instruction " for " permet d'itérer sur des tableaux et des collections sans manipuler d' "Iterator". En effet, celui-ci ne sert qu'à obtenir les éléments de la collection et son fonctionnement n'est pas trivial. La nouvelle instruction " for " permet donc d'itérer sur une collection ou un tableau à la manière d'un "foreach". Ainsi, on peut remplacer, dans l'exemple 1, la ligne 8 à 12 par le code suivant :

```
for(String item : list)
  {
     System.out.println("Item: "+ item);
}
```

Le fonctionnement est le même pour les tableaux. Cette amélioration est l'une des plus importantes et permettra de simplifier énormément la lecture des boucles.

Pareillement à Doclet, la fonctionnalité de "Metadata" permet d'associer des informations additionnelles aux classes, interfaces, méthodes ou champs. Ensuite, un outil, tel



"apt" (Annotation Processing Tool), peut utiliser les " metadata " pour générer du code additionnel ou pour fournir des informations supplémentaires lors du débogage. Cette nouvelle fonctionnalité, basée sur une syntaxe particulière, le " @ ", a un fort potentiel (la plateforme .NET l'a prouvé), mais nécessite un temps d'adaptation avant de rentrer dans les moeurs des développeurs.

La fonctionnalité de " Static Import " permet aux développeurs d'importer statiquement des membres d'une classe ou d'une interface et de les utiliser, sans les préfixer par leur nom. Précédemment, une classe devait implémenter l'interface pour avoir accès aux constantes définies dans celle-ci.

```
/ / import statique
import static com.name.XYZ.*;
import static java.lang.System.*;

public class MyClass {
  ...
  / / Constant1 est définie dans une classe ou
 interface XYZ
  double value = 2 * Constant1;
  / / out est la variables System.out
  out.println(value);
}
```

L'intérêt se situe surtout au niveau des bibliothèques mathématiques, où l'utilisation des méthodes sin(), cos() sera beaucoup plus explicite. Attention toutefois à la mauvaise pratique concernant la généralisation d'interfaces à membres constants qui seraient ensuite importées statiquement.

Ajout d'un type pour les énumérations statiques, nommé " enum " qui s'approche fortement de son équivalent en C/C++ pour la syntaxe. De nombreux avantages sont cependant spécifiques à Java. Les développeurs peuvent dorénavant utiliser le " printf " pour générer des sorties formatées. La plupart des formatages C les plus communs sont disponibles. Cela peut être utile lors de la migration d'applications C, en gardant le formatage avec peu ou pas de modifications. Pour la lecture de flux d'entrée, l'API scanner fournit des fonctionnalités basiques pour lire les données depuis la console ou n'importe quel flux. S'il faut traiter une entrée plus complexe, il est possible de définir des algorithmes basés sur le pattern-matching, disponible à partir de la classe java.util.Formatter. La fonctionnalité varargs permet à de multiples paramètres, dont le nombre est indéterminé, d'être passés comme 'paramètres' à une méthode. Cela nécessite la simple notation " ... " dans la définition de la méthode afin de récupérer la liste des arguments.

Machine Virtuelle, la tête...

Le Class Data Sharing (CDS) est une nouvelle fonctionnalité de J2SE 1.5 et a pour but de réduire le temps de démarrage pour les applications java, particulièrement les petites. A l'installation, l'installeur Sun charge un jeu de classes de base dans une représentation propriétaire, puis décharge cette représentation dans un fichier appelé " archive partagée "

(Shared Archive). Lorsque la JVM invoque des classes contenues dans l'archive, celle-ci est mappée en mémoire, sauvegardant ainsi le temps de chargement de ces classes et permettant aux méta données de ces dernières d'être partagées parmi plusieurs processus. Pour l'instant, il semble que cette fonctionnalité demande des améliorations qui seront, sans doute, apportées lors des prochaines versions. Déjà, toutefois, des augmentations de performances ont été notées pour des programmes légers. Pour le " garbage collector ", le collecteur parallèle a été amélioré pour gérer et s'adapter aux besoins de l'application en terme de mémoire. On peut maintenant spécifier les objectifs de performances et la JVM adaptera la taille de sa mémoire, pour atteindre ces objectifs avec la plus petite implication pour l'application. Le paramétrage des options en ligne de commande n'est donc plus nécessaire. La performance du garbage collector a continuellement et discrètement augmentée durant les versions successives de J2SE, et elle semble atteindre avec cette version 1.5 un bon niveau de performance et une consommation mémoire réduite.

Les API et outils, le corps...

La nouvelle API réseau fournit aux applications Java le support de IPv6 et un mode de lancement par inetd. Elle offre de plus, la possibilité de paramétrer des timeout sur les opérations de connections et de lecture pour les protocoles handlers. L'API donne aussi un moyen à l'application de tester la disponibilité de l'hôte, l'équivalent d'un ping en java, plus un support amélioré pour les " cookies ", la configuration améliorée du serveur proxy et diverses autres fonctionnalités.

Cette version de J2SE fournit de nombreuses améliorations concernant la sécurité, comme un meilleur support pour les librairies de sécurité, plus de standards (SASL, OCSP, TSP), une évolution pour la scalabilité (SSLEngine) et la performance, et enfin, de nombreuses améliorations pour la cryptographie (JCE) et le domaine de Java GSS (Java Generic Security Services). L'une des modifications de JAXP est l'incorporation de DOM (Document Oriented Model) niveau 3. Cette nouvelle version de JAXP intègre la recherche via des expressions XPath, grandement attendue par les développeurs (et présente sur la plate-forme Microsoft depuis longtemps), ainsi que la possibilité d'utiliser d'autres validateurs de schéma. Enfin, XSLTC, le compilateur de transformation, devient le moteur par défaut.

La bibliothèque de gestion de la concurrence est issue du package de Doug Lea. Elle est bien intégrée et très performante sans être vraiment très conviviale. Elle fournit des constructions haut niveau et puissantes pour les threads, représentées, par exemple, par le framework de tâches pour les threads, les queues mutli-threads, les timers, les locks et autres primitives de synchronisation.

Un point clé est celui du monitoring et du management intégré dans RAS (Reliability, Availability, Serviceability). En ce domaine, Tiger apporte beaucoup de nouveautés: instrumentation pour observer la JVM en exécution, framework JMX (Java Management Extensions) et des protocoles d'accès distants (SNMP). Le tout prêt à être utilisé " out-of-the-box ".

Les améliorations JDBC concernent l'interface RowSet qui a été implémentée en cinq classes, dont deux très utiles, CachedRowSet et WebRowSet. Celles-ci peuvent fonctionner, sans être connectées en permanence à la base ou une autre datasource, et sont plus légères que les objets ResultSet. La classe CachedRowSet contient en mémoire une collection de lignes obtenues de la base et qui peut, si besoin, être synchronisée ultérieurement. La classe WebRowSet peut, quant à elle, écrire et lire les RowSet dans un format XML. Il est possible que cette API fasse l'objet de plusieurs améliorations dans les versions suivantes. En effet, elle manque de répondant face à son équivalent .NET. Toutefois elle semble bien intégrée avec JSF, ce qui laisse présager le meilleur.

Fondamental, JVMTI (Java Virtual machine Tool Interface), remplace JVMDI (debugger) et JVMPI (profiler). C'est une nouvelle interface de programmation utilisée pour les outils de développement et de monitoring. Il fournit un moyen de voir l'état et de contrôler l'exécution d'application tournant dans la JVM, ainsi qu'une interface pour accéder à l'état de la VM, incluant : outils de profiling, de debugging, de monitoring et d'analyse de thread.

Interface utilisateur, la robe

Concernant Swing, peu d'améliorations, on peut cependant noter l'intégration d'un nouveau thème, " ocean " et un support pour l'affichage des JTable. On pourra regretter les problèmes de fonte, toujours présents, nuancés cependant par un Look&Feel XP de plus en plus intéressant.

Les bibliothèques AWT et java 2D Technology se dotent de quelques améliorations. La Java Sound Technology, quant à elle, arrive dorénavant à remplir les attentes des utilisateurs, de nombreux bogues ayant été corrigés.

J2SE 1.5 est une étape majeure pour la plate-forme Java et un nombre impressionnant de fonctionnalités, d'améliorations du langage et d'outils ont été intégrés pour faciliter le développement d'applications Java. Toutefois, on peut regretter que des efforts n'aient pas été fournis sur l'amélioration de la gestion du système de fichier et sur la gestion des dates, réputées depuis longtemps pour leur manque d'ergonomie. Malgré cela, on dénote un très grand professionnalisme, notamment une intégration avec Windows nettement améliorée. Le processus d'installation et de désinstallation est efficace et au niveau déploiement, Java WebStart est beaucoup mieux intégré avec une bonne gestion des proxies et des standards. Bref cette version orientée développeur tient ses promesses, tant au niveau convivialité qu'au niveau sérieux et on sent une réelle volonté d'instaurer J2SE 1.5 comme première plate-forme de production. On s'en lècherait presque les babines...

■ **Christophe Bailly-Maître**
Consultant chez Neoxia,
cabinet de conseil en architecture
des systèmes d'information.
c.baillymaitre@neoxia.com

✓ L'actualité en ligne
www.programmez.com

✓ Abonnez-vous en ligne
www.programmez.com/abonnement

JAVA TV : Terminaux numériques, un avenir radieux ?

Java s'impose aussi en dehors des applets, serveurs et autres applications. Dans le monde de la " Set Top Box ", décodeurs numériques " intelligents ", la plate-forme et le langage Java sont parvenus à s'imposer assez largement. Microsoft a toujours du mal à prendre pied dans ce marché stratégique qui connaît une croissance importante.

La société NDS France illustre parfaitement la percée de Java dans ce monde interactif. " Depuis 1998-99, nous avons fait le choix de Java dans les terminaux. Avant, nous utilisions un langage propriétaire. DVB a standardisé en 2000 les API basées sur JavaTV ". Explique Cédric Monnier (Senior Middleware Product Manager, NDS). DVB est un consortium spécifiant l'ensemble des APIs Java pour le monde de la télé. Comme les applications Java ne fonctionnent pas telles quelles dans les terminaux, il a fallu aussi définir un nouveau modèle d'application : les "Xlets".Le standard MHP (Multimedia Home Platform) décrit alors l'ensemble du modèle d'exécution et des APIs de ces Xlet dans une architecture terminal TV. Si par le passé, les constructeurs et fournisseurs de contenus utilisaient des formats et outils propriétaires, aujourd'hui et demain, ce ne sera plus du tout le cas. Java (via MHP) s'est réellement imposé, comme un standard ouvert. " Ce standard permet de créer un véritable écosystème entre le fabricant de décodeurs, l'opérateur de télévision, le fournisseur de middleware et les développeurs et de booster le marché de la télévision numérique. Bien sûr, MHP continue à évoluer. La version couramment déployée est la 1.0.2. L'ensemble des décodeurs certifiés MHP l'implémente. " " La prochaine version, poursuit Cédric Monnier, aura pour ambition d'apporter de nouvelles API, notamment pour les fonctions de magnétoscopes numériques, dites " Personal Video Recording ". Actuellement, ces API sont propriétaires."

MHP est un standard défini en Europe. Aux USA, les principaux opérateurs sont encore sur des systèmes propriétaires. Mais depuis quelques années, le monde du câble américain a décidé une convergence vers un vrai standard. Aux USA, comme l'explique Cédric Monnier, il y a une culture câble très forte, contrairement à l' Europe. Il y a une voie montante et descendante et un fort développement de la Video On Demand. Les sous-titres (closed caption) sont très présents aux USA. Au lieu de refaire le monde, les Américains ont pris le meilleur de MHP et ont développé des fonctions locales. Il en résulte aujourd'hui la définition du standard OCAP (Open Cable Application Platform). Au final, le développeur et les fournisseurs disposent du même jeu d'API fameux, avec pour seules différences, des API spécifiques à chaque zone. Ainsi, des applications Java MHP pourront fonctionner aux USA. C'est l'avantage de posséder des API ouvertes et standards.

Temps réel

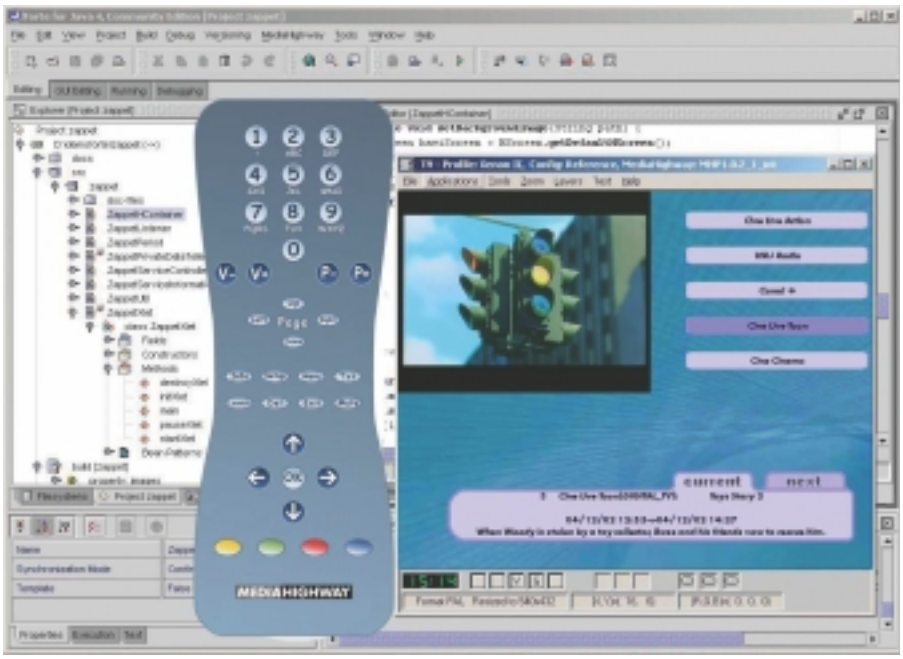
Nous sommes là dans un marché à part, l'embarqué. Les contraintes sont fortes. " Nous faisons du temps réel. Nous avons deux types d'applications, les applications résidentes, le cas idéal, mais réservé au fournisseur de middleware, où l'application est stockée dans le décodeur et l'application diffusée. Dans ce cas, il y a un souci de bande passante et la place disponible pour les données est restreinte. De plus, comme on ne sait quand l'utilisateur va se servir d' une application, on est obligé de faire une diffusion cyclique, généralement toutes les 3 minutes. Ainsi,on ne surcharge pas la bande passante et l'utilisateur attend très

JavaTV & Cie

JavaTV est un ensemble d'API diffusé par Sun, dédié à la télévision interactive. À la base, JavaTV s'occupe de : streaming audio/video, accès par condition, accès aux chaînes de données, accès aux services d'informations, contrôle du tuner, contrôle graphique sur l'écran. À cela, on peut aussi ajouter les fonctions de gestion de la vie de l'application, et la synchronisation. Ces API sont de haut niveau, ils reposent sur la plate-forme Java (dont la JVM), la couche du système (temps réel) et enfin la couche matériel. Au-dessus de JavaTV, on trouve les applications. D'autres outils pour la télévision interactive existent, utilisant ou non Java : OpenTV, NPTV, pour ne citer qu'eux. Sur la partie USA, L'initiative OCAP (OpenCable Application Platform) mise sur Java. Un groupe de travail (JCP 242) travaille dessus pour spécifier un OCAP Java et permettre d'avoir une couche middleware bien définie. Ce travail fournira un nouveau profil CLDC (connected limited device configuration). Le tout s'appuyant sur J2ME. Surtout, les initiateurs du projet précisent que ce nouvel ensemble sera compatible avec les spécifications actuelles. Cet OCAP Java est une étape importante pour pouvoir imposer Java comme base commune à une grande partie des opérateurs de télévision.

peu de temps son application. Il y a des contraintes dans l'embarqué et le temps réel. Il faut utiliser les threads, et d'autres fonctions sont à bannir comme le garbage ou la sérialisation " précise Cédric Monnier.

Bien entendu, n'allons pas croire que Java est utilisé de bout en bout dans l'OS jusqu'à l'application. NDS s'occupe surtout de la partie middleware des décodeurs. Celui-ci est écrit en C, pour des questions de performances. Java est utilisé dans la partie haute. La JVM est implémentée en C. Pour faciliter le travail du développeur, NDS dispose d'un outil RAD pour designer visuellement les écrans des applications et les tester. Pour le navigateur, la société utilise un browser Opera 6.0. Le kit de développement de NDS est uniquement disponible sous Windows. " Linux, on regarde " avoue Cédric Monnier.



La création d'une application pour décodeur est simple :

- disposer d'un IDE. NDS propose des plug-ins pour Jbuilder et NetBeans.
- Un simulateur de décodeur constitué d'une télécommande et d'un écran de télé. Le simulateur simule le soft et le matériel. Pour les tests, il simule aussi un flux de chaîne.

Bien entendu, on dispose aussi d'un mode debug, notamment via un trace. NDS a développé des API de debug spécifique. " Nous avons créé un système de profiling. Bien qu'il soit propriétaire, on propose un .JAR ", complète Cédric Monnier. Bien entendu, impossible de tout créer uniquement sur son PC. À un moment donné, il faut procéder à un test réel, dans un premier temps, en local, pour tester l'application, puis dans une dernière étape, on teste l'application chez l'opérateur.

Faut-il avoir des connaissances particulières pour se lancer dans ce type d'applications ? Déjà, bien maîtriser Java, comprendre les API (ce qui est plus facile à dire qu'à faire) et bien entendu comprendre les contraintes fonctionnelles et techniques des décodeurs. C'est donc essentiellement un investissement en temps. NDS propose même un cycle de 5 jours de formation. Et l'état du marché ? La réponse de Cédric Monnier est sans détour : " le marché est en train d'exploser. Avec le passage de l'analogique au numérique et le fort engouement pour le PVR, nous estimons qu'environ 60 millions de décodeurs vont être changés. NDS a une vraie maturité, la technologie est stable et MHP est maintenant stabilisé ".

Et la concurrence de Microsoft ? Une fois n'est pas coutume, Microsoft propose ou tente de proposer une alternative Windows sur ce marché. Pour le moment, les différents essais et tests de l'éditeur se sont soldés par des échecs. La réponse de NDS est très claire : "on n'y croit pas".

■ **François Tonic**

Java : premier acteur du smartphone ?

Autre marché très en forme en ce moment et dans les mois à venir, le smartphone. Grâce au support de J2ME dans plusieurs dizaines de millions de portables, par le soutien de dizaines d'opérateurs télécoms et grâce à sa portabilité, Java est en passe de devenir un langage et une plate-forme de base pour le développement sur smartphone dans les applications et les jeux. Java profite de plusieurs facteurs : l'échec du WAP, le développement de services de plus en plus lourds, le retard pris par Microsoft pour proposer une plate-forme fonctionnelle. Les jeux vont exploser à partir de 2005 pour totaliser un marché estimé à plus de 7 milliards de dollars en 2008 (sources Nokia). Des analystes vont jusqu'à prédire un revenu de 10 milliards de dollars pour le téléchargement d'applications Java en 2008 (sources Nokia). Ainsi en 2003, le support en Europe de Java par les opérateurs a été multiplié par deux en moins de 10 mois ! Tout cela confirme la place aujourd'hui incontournable de Java dans le smartphone. Il y a un an, selon les chiffres fournis par Sun, il existait plus de 200 modèles Java de smartphone ! Cette tendance se vérifie aisément quand on voit que des éditeurs sortent des IDE Java, spécifiques aux mobiles. Le succès croissant des mobilettes confirme aussi cela.

Si aujourd'hui, J2ME et MIDP 2.0 apportent une réponse aux attentes des opérations et utilisateurs, Sun se rend compte qu'il est nécessaire d'obtenir sur des terminaux mobiles une plate-forme quasi identique au J2SE. Ce sera une réalité courant 2005. Le jeu connaît notamment une forte explosion et Java tire bien entendu son épingle du jeu (notamment avec la disponibilité du Gaming API). Pour compléter l'offre fonctionnelle, Sun a sorti le J2ME Wireless Toolkit (actuellement en 2.2 bêta), valable pour MIDP et CLDC. Porter une application desktop sur Smartphone ou PDA est possible. Mais là, il faudra réfléchir aux fonctions à intégrer, et à l'interface.

Mettez une mobilette dans votre téléphone

En 2 ans, le monde de la téléphonie mobile s'est métamorphosé, mais cela s'est fait en douceur et si discrètement que même les vendeurs de téléphones ne s'en sont toujours pas rendu compte. En effet, il y a 2 ans, les téléphones mobiles étaient... des téléphones, alors que désormais plus de 70 % des nouveaux modèles, même les moins chers, sont des ordinateurs programmables en Java qui permettent bien sûr de téléphoner, mais aussi d'utiliser des applications, des jeux, de se connecter à des serveurs web pour recevoir ou envoyer des informations, et pour télécharger de nouveaux programmes.

Vous êtes sans doute sceptique devant ces affirmations, alors je vous propose de partager un de mes petits jeux. Regardez les téléphones de vos amis : s'ils ont moins d'un an, et qu'ils sont de marque Alcatel, LG, Motorola, Nokia, PalmOne, Panasonic, Philips, Sagem, Samsung, Sendo, Sharp, Siemens ou Sony Ericsson, alors il est probable qu'ils soient compatibles Java. La chance de vos amis dépend de la marque qu'ils ont choisie : si c'est un Nokia et qu'il est couleur, il est forcément compatible ! S'il est noir et blanc et que c'est un 3410 ou un 6310i, alors il incorpore aussi Java. Dans les autres marques, Java est présent dès la gamme moyenne chez Siemens, Motorola, Samsung, LG ou Sony Ericsson. Et depuis 2004, même les marques plus tournées vers l'entrée de gamme proposent des modèles compatibles Java, comme Sagem, Philips ou Alcatel. Enfin, tous les téléphones Vodafone Live vendus par SFR sont systématiquement compatibles Java, car un des services de Vodafone Live est le téléchargement de jeux Java. Alors, si vous n'avez pas la chance d'avoir un téléphone Java, je suis sûr que désormais vous avez remarqué que vous avez de nombreux amis qui ont, sans le savoir, un téléphone Java.

Des mobilettes dans les téléphones

Vous connaissiez les applets, ces petites applications Java qui tournent dans votre navigateur web. En France on parle de mobilette pour désigner un programme Java conçu pour les téléphones mobiles. Java a depuis longtemps été adapté aux appareils itinérants, avec Java 2 Micro Edition. J2ME est découpé en configurations et en profils, afin de s'adapter aux besoins spécifiques et aux contraintes des périphériques mobiles (cartes à puces, téléphones, décodeurs télé, tableaux de bord des BMW,...).

La configuration définit un sous-ensemble de Java, un minimum garanti en quelque sorte, car toutes les classes de Java 2 Standard Edition (J2SE) ne pourraient loger dans de petits périphériques et la plupart d'entre elles seraient inutiles (gestion de l'écran sur une carte à puce par exemple). Les mobiles utilisent la configuration CLDC : Connected Limited Device Configuration, qui est conçue pour les périphériques disposant d'une connexion internet par HTTP à bas débit et qui peut être interrompue (n'oubliez pas les tunnels). Actuellement, la version la plus répandue est la 1.0. Elle modifie la Java Language Specification, car elle ne supporte pas les nombres à virgule, donc ni les " float " ni les " double ", car ils nécessiteraient une trop grande puissance de calcul (souvenez-vous de vos processeurs 386, ils ne les géraient pas non plus sans émulation 387). CLCD 1.0 impose au minimum 160Ko de RAM disponible pour votre application. Toutes les classes de base de Java sont là (String, StringBuffer, InputStream, Vector...), mais tout le reste a été éliminé pour économiser de la mémoire. Vous ne trouverez donc pas les classes moins indispensables ou plus élaborées comme les BufferedInputStream, Collection ou encore la classe URL.

CLDC ajoute quelques classes, dont la classe Connector qui permet d'ouvrir une connexion, modélisée par l'interface Connection, à partir d'une unique méthode open(String). Ceci permet aux mobilettes d'utiliser des algorithmes génériques quels que soient les protocoles utilisés (http, https, port série, sockets,...).

Motorola, Nokia, PalmOne et Sony Ericsson proposent chacun un modèle compatible CLDC 1.1 qui a comme principal intérêt de supporter les types float et double, bien utiles aux programmes affichant des animations en 3D.

Maintenant que nous avons vu la configuration en détail, voyons le profil. Il définit des extensions à Java, spécialement conçues pour la catégorie des périphériques concernés. Pour les téléphones il s'agit de MIDP : Mobile Information Device Profile. Ces extensions sont placées dans 4 packages :

- javax.microedition.midlet : pour le cycle de vie de la mobilette (notez qu'en anglais un programme compatible MIDP s'appelle une midlet, mais gardons le mot français mobilette)
- javax.microedition.rms : pour le stockage persistant (gestion de la mémoire Flash)
- javax.microedition.io.HTTPConnection : pour le réseau
- javax.microedition.lcdui : pour l'affichage, la gestion du clavier et du stylet (quand le téléphone en a un).

Si vous voulez faire une mobilette, il vous suffit donc de connaître les classes de CLDC 1.0 (le sous-ensemble du Java traditionnel) et les classes ajoutées par MIDP 1.0, soit au total seulement 58 classes (+ les exceptions), dont seulement 24 classes complètement nouvelles. Vous l'avez compris, apprendre à écrire une mobilette est extrêmement facile et rapide,

même pour ceux qui ne connaissent pas Java !
La nouvelle norme MIDP 2.0 ajoute de nombreuses classes et des concepts assez évolués, en particulier pour la gestion de l'écran. Si tous les nouveaux modèles de téléphones annoncés sont systématiquement compatibles MIDP 2.0, les modèles actuellement en boutique sont encore en grande partie compatibles MIDP 1.0. Je vous conseille donc de commencer par faire des programmes pour MIDP 1.0, puis quand vous serez à l'aise (et que tout le monde aura des téléphones MIDP 2.0) de progressivement apprendre et profiter des nouveautés de MIDP 2.0.

Les outils de développement

Tout d'abord, il vous faut le SDK 1.4.2 pour avoir un compilateur Java compatible avec les besoins des outils pour MIDP. C'est une version très récente, (novembre 2003) donc même si vous utilisez déjà Java, il est probable que vous deviez télécharger cette nouvelle version :
http://java.sun.com/j2se/1.4.2/download.html
Sur cette page, cliquez sur " Download J2SE SDK " (anciennement appelé JDK).

Ensuite, vous avez besoin des classes de CLDC et de MIDP pour pouvoir compiler votre code. Le plus simple est de prendre l'outil dédié de Sun, le Wireless Toolkit 2.1 :
http://java.sun.com/products/j2mewtoolkit/download-2_1.html
Sur cette page, cliquez sur " Download " tout en bas.
Le Wireless Toolkit est un outil basique, mais utile : il vous crée une arborescence pour votre projet, vous permet de choisir les versions de CLDC et de MIDP que vous souhaitez utiliser. Puis d'un clic, vous pouvez compiler, exécuter votre mobilette dans des émulateurs, et quand tout est prêt, générer les fichiers à installer dans le téléphone. Il n'y a pas d'éditeur de code, vous utiliserez donc un éditeur texte de votre choix, ou votre outil de développement préféré.
Personnellement, j'utilise Eclipse 2.1. Après avoir créé un projet dans le Wireless Toolkit, je crée un projet Java du même nom dans Eclipse. Dans l'assistant, je retire le dossier "source" par défaut (src), puis j'ajoute un nouveau dossier source que je crée un utilisant l'option avancée " link to folder in the file system ", le dossier cible choisi étant le dossier source " src " créé par le Wireless Toolkit. Ainsi, les 2 outils travaillent avec le même code source. Ensuite, dans l'onglet "libraries" des propriétés du projet, je supprime " JRE System Library " et j'ajoute 2 fichiers jars externes qui se trouvent dans le répertoire lib du Wireless Toolkit : cldcapi10.jar et midpapi10.jar. Voilà, je dispose de toute la puissance d'Eclipse quand je code (assistance à l'écriture du code par CTRL-Espace, messages d'erreur de compilation explicites, suggestions). Mais je dispose aussi du Wireless Toolkit pour tester mon application dans les émulateurs et surtout pour générer les fichiers pour le téléphone (n'essayez jamais par un autre moyen que le Wireless Toolkit, des détails techniques vous feraient perdre beaucoup de temps).
Pour ceux qui préféreraient un outil tout-en-un, il y a des versions de Netbeans, JBuider ou WSAD spéciales pour le développement en J2ME. Comme je conçois avec mon esprit et non pas avec un

PHP 5 avancé
Cyril Pierre de Geyer
Eric Daspet
PHP 5 avancé
Eric Daspet
Cyril Pierre de Geyer
EYROLLES
730 pages,
457 versions,
30 cas pratiques,
26 chapitres,
36 mois de travail,
10 ans d'experience,
5ème version,
2 auteurs,
un livre
recommandé par :
NeXeN
PHPdebutant.COM
PHP France

débogueur, ces outils ne m'apporteraient aucun avantage ;-)

Pour finir sur les outils, n'oubliez pas de consulter la documentation de CLDC/MIDP qui se trouve dans le sous-répertoire docs/api/midp/index.html du Wireless Toolkit. Cette documentation mentionne systématiquement " Since MIDP 2.0 " ou " Since CLDC 1.1 " pour les fonctions non présentes dans CLDC1.0/MIDP1.0.
Voilà, vous savez l'essentiel, il est temps de passer à la pratique.

Votre première mobilette

1. Lancez le " J2ME Wireless Toolkit " depuis le menu démarrer, ou en lançant l'exécutable ktoolbar.
2. Cliquez sur le bouton New project. Pour Project Name saisissez EcoLogic pour MidletClassName entrez aussi EcoLogic et cliquez sur Create Project.
3. Le Wireless Toolkit crée alors un répertoire pour contenir votre projet, puis affiche une fenêtre de paramétrage du projet. Dans la liste Target Platform choisissez MIDP 1.0 afin que votre projet fonctionne sur tous les téléphones Java, y compris les modèles les plus anciens. Cliquez ensuite sur l'onglet Required, et dans la colonne Value de la ligne MIDlet-Vendor, saisissez votre nom et si vous le souhaitez votre adresse mail. Pour finir, cliquez sur l'onglet Optional et dans la colonne Value de la ligne MIDlet-Description entrez la description de la mobilette : Calcul de la consommation de votre voiture. Enfin, cliquez sur le bouton OK.
4. Dans la fenêtre du J2ME Wireless Toolkit, notez le nom du répertoire indiqué après "Place Java source files in". Dans ce répertoire, copiez le fichier EcoLogic.java qui se trouve sur le site. Il contient le code source de la classe EcoLogic. Le code source est documenté et devrait être assez simple à comprendre, surtout si vous ouvrez la documentation de MIDP dans votre navigateur web.
5. Retournez dans le J2ME Wireless Toolkit et cliquez sur le bouton Build. Le texte " Build complete " doit apparaître dans la fenêtre. Si ce n'est pas le cas, corrigez l'erreur dans le source Java, sauvez et recommencez le build.
6. Testez si tout fonctionne bien en cliquant sur le bouton Run. Un émulateur apparaît alors. Attention, il simule un vrai téléphone, donc inutile de cliquer sur l'écran, les téléphones n'ont en général pas d'écran tactile ! Le téléphone s'utilise donc en cliquant sur ses boutons, en particulier sur les 2 boutons situés sous l'écran. Pour démarrer la mobilette, cliquez sur le bouton affiché sous le texte " Launch ". Le curseur de la souris se transforme en main quand vous êtes au-dessus du bouton (tant que vous restez sur l'écran du téléphone c'est une flèche et les clics ne fonctionnent pas). Vous pouvez aussi utiliser la touche Entrée, les flèches et les chiffres directement sur votre clavier.
7. Quand les tests sont concluants, vous pouvez préparer les fichiers nécessaires pour l'installation sur votre téléphone. Pour cela, utilisez le menu Project _ Package _ Create Package. Si tout se passe bien, l'outil vous indique où il a stocké les deux fichiers EcoLogic.jad et EcoLogic.jar.
8. Si votre téléphone est livré avec les utilitaires adéquats, vous pouvez installer directement ces fichiers sur votre téléphone (avec un câble, par infrarouge ou par Bluetooth). Passez à l'étape 11.
9. Sinon, placez ces deux fichiers sur un serveur web dans un seul répertoire. Le serveur web doit fournir le bon type MIME. Si vous utilisez Apache, créez simplement un fichier .htaccess contenant les 2 lignes suivantes :
   AddType text/vnd.sun.j2me.app-descriptor jad
   AddType application/java-archive jar

En cas de problème, consultez la notice de votre serveur ou votre webmaster.

10. Le téléchargement se fait depuis le téléphone, en allant dans le navigateur wap (parfois appelé " Internet " ou encore "Services" chez Nokia). Saisissez l'URL complète de votre fichier jad (en respectant minuscules et majuscules) par exemple :http://wap.club-java.com/ota/EcoLogic.jad
    Pour plus de facilité, vous pouvez aussi créer des pages wap incluant des liens vers vos fichiers jad. C'est ce que nous avons fait sur notre site wap http://wap.club-java.com jetez-y un coup d'œil, ça vous donnera des idées. Le wap (et son langage le WML) sont un sujet indépendant de Java et ne seront donc pas couverts dans cet article.
11. La mobilette est maintenant installée dans votre téléphone, vous pouvez l'utiliser autant de fois que vous le désirez, c'est gratuit, puisqu'elle ne fait qu'utiliser les ressources de votre téléphone (des mobilettes plus évoluées utilisent des connexions HTTP qui sont elles, facturées par votre opérateur). En général les mobilettes sont dans un menu "Applications " (Nokia) ou " Jeux et applications " ou " Divertissements ". Là il vous suffit de choisir votre mobilette EcoLogic pour qu'elle démarre.

Ça y est, vous avez ajouté des fonctions à votre téléphone. Vous constaterez qu'il est extrêmement simple de réaliser de la même manière tous les convertisseurs dont vous pourriez avoir besoin (franc/euro, miles/km, Fahrenheit/Celsius, MHz/nano-secondes,...). Maintenant que vous avez réalisé que vous avez toute la journée un ordinateur dans la poche, aussi puissant que celui que vous aviez sur votre bureau il y a 15 ans, votre imagination est la seule limite pour créer de nouvelles mobilettes.

Sur le web :
Liste des téléphones compatibles Java disponibles en France, en Europe et en Afrique:
http://www.club-java.com/Public/telephones/telephones_mobilettes.html
TastePhone : Mobilette Open Source de benchmark de téléphones avec publication des résultats des téléphones en ligne :
http://tastephone.dev.java.net/

■ Thibaut REGNIER
Secrétaire Général du Club des Utilisateurs de Java - mobilette@club-java.com

bioMérieux trace le bug

cas d'entreprise

Acteur majeur du diagnostic biologique (numéro 8 mondial), bioMérieux, fournit des instruments et kits de diagnostic dans le monde entier. Le département IT de l'entreprise est donc un lieu sensible où le bug est traqué dans tous les sens, pour fournir des applications de diagnostic les plus fiables possible...

Le département IT se divise en 5 départements : ingénierie du logiciel, biomathématique, bioinformatique, administration systèmes et le management. Autrefois réalisé manuellement, avec l'informatique, le diagnostic s'est entièrement informatisé et automatisé, mais les contraintes légales sont énormes. "C'est un domaine très réglementé où la qualité revêt une grande importance" précise Roland Chevalier (Responsable architecture système au service ingénierie du logiciel de bioMérieux).

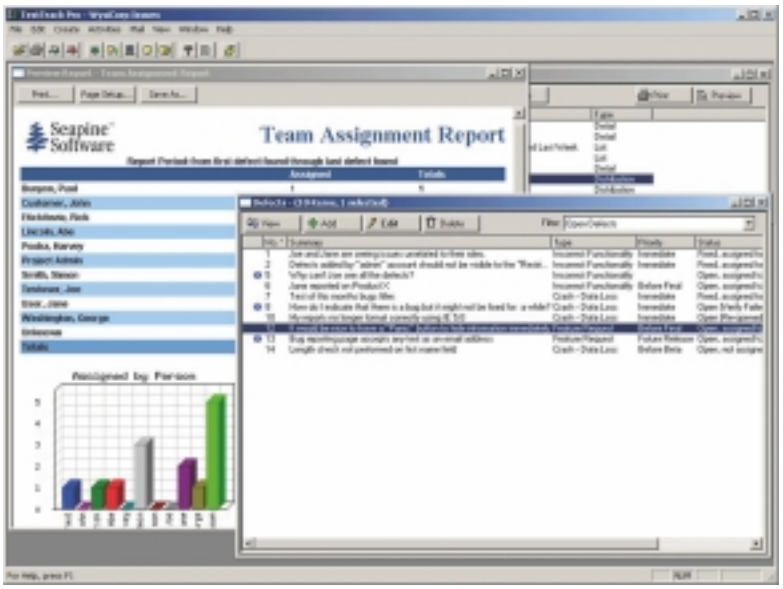
Les applications réalisées par le département IT sont passées par plusieurs périodes : C, C++, puis maintenant Java. Même si, comme le dit Roland Chevalier, avoir un langage portable n'était pas l'objectif premier, cela peut être utile. " Il nous faut une pérennité " poursuit R. Chevalier. Aujourd'hui, une très grande partie des applications est passée à Java, même si des morceaux demeurent en C++. Le choix de Java se porta dans un premier temps pour sa partie graphique, puis pour sa simplicité de programmation comparée au C++. Pour le développement, si au départ Jbuilder fut largement utilisé, aujourd'hui, Eclipse prime. Sur le nombre de bugs, il y a toujours l'espoir d'en réduire un peu. Ce fut en partie le cas, mais en réalité, " on a constaté que l'on n'avait pas les bugs au même endroit. On a effectivement moins de bugs en Java, mais cela est difficile à mesurer " poursuit R. Chevalier.

Les tests des applications font partie de la culture maison. Il faut toujours prouver que les résultats sont bons. Dans le diagnostic, l'erreur n'est pas permise. Si auparavant les tests d'analyse étaient manuels, aujourd'hui, tout est automatisé. Mais dans le service IT, il fallait aussi améliorer le processus de suivi du bug. Il était réalisé manuellement, et cela prenait naturellement du temps. C'est pour cela que le choix d'un outil dédié fut décidé. Deux outils étaient en lice. Le département IT américain utilisé déjà TestTrack Pro (Distribué en France par IDEO Logiciels).

Alléger le traçage du bug

Les collègues américains en parlèrent donc à leurs homologues français. Le but de cet outil est d'améliorer le suivi du bug, sa traçabilité entre sa découverte et sa correction, tout en mettant en place un workflow interne au service IT. " Les deux produits pressentis étaient assez proches. Il y avait un petit avantage pour TestTrack Pro sur le prix. Le mode de déploiement nous a convaincus, avec la licence flottante " confie Roland Chevalier. Cette licence a permis de tester grandeur réelle l'outil, sans pour autant mobiliser un budget trop important et d'adapter le nombre de licences au fur et à mesure. Aujourd'hui, environ 70 personnes l'utilisent en France, dont plus de la moitié sont des développeurs. Même si de l'aveu de R. Chevalier, le processus de traçage de bug n'a pas été amélioré notoirement, il a été allégé. Comme future évolution, il est envisagé de réaliser des métriques à partir des données des bugs collectés.

■ F.T.



L'outil TestTrack Pro, la traçabilité du bug.

Eclipse 3.0, socle des nouveaux outils Rational

IBM developerWorks Live!

Pour contrer ses concurrents et investir le marché de Microsoft, IBM prépare une plate-forme de développement complètement intégrée, basée sur Eclipse 3.0.

Durant la conférence annuelle des utilisateurs Rational, fin juillet à Dallas (Texas), IBM a présenté sa nouvelle stratégie pour conquérir les développeurs.

Avec le rachat de Rational Software il y a 18 mois, IBM dispose d'une offre d'outils de développement, modélisation, tests, complète. Après avoir lancé fin 2001 la plate-forme de développement open source Eclipse (basée sur le code source de WebSphere Studio Application Development), il restait donc à IBM d'intégrer l'ensemble de ces outils au sein d'une suite d'outils de développement parfaitement intégrée.

Michael Devlin, General Manager Rational software, IBM Software Group.

C'est aujourd'hui une réalité avec la suite connue sous le nom de code Atlantic. Présentée officiellement en octobre prochain, Atlantic " aidera tous les participants – analystes, développeurs, testeurs, administrateurs, etc. - d'un projet à collaborer entre eux " a expliqué Michael Devlin, General Manager Rational software, IBM Software Group, lors de la conférence inaugurale de l'événement.

Support d'UML 2.0, Hyades, JSF et C#

Atlantic est une version améliorée d'Eclipse 3.0 à laquelle IBM a ajouté des fonctions de collaboration et de travail en équipe, de test et de modélisation. Certaines de ces fonctionnalités (collaboration notamment) sont directement intégrées au noyau de l'outil et nécessitent un serveur pour stocker les données des projets et gérer les interactions entre participants.

Geoffrey Bessin is Market Manager for IBM Rational's software quality products.

Parmi les autres nouveautés d'importance, on peut également citer l'ajout d'un modeleur UML 2.0 dont le code source devrait être rapidement reversé à la communauté open source et le support des JavaServer Faces. IBM a également indiqué qu'il s'appuiera désormais sur le framework open source Hyades pour ses modules de test de qualité. "Hyades est la roue que tous les éditeurs réinventent à chaque fois qu'ils développent un outil d'assurance qualité" a précisé Geoffrey Bessin, Market Manager de l'offre " qualité " chez IBM Rational.

Plus important, Atlantic reposant sur Eclipse, les langages .NET – C# et VB.NET notamment – et les différentes versions de C – C, C++, etc. – sont supportées. Avec Atlantic, IBM déborde du marché Java / J2EE. L'éditeur s'attaque directement à Microsoft et Borland.

■ **David Thévenon**

Abonnez-vous en ligne
www.programmez.com/abonnement

JAVA

JavaOne 2004 : technique avant tout

On attendait beaucoup de cette JavaOne édition 2004 qui se déroulait sur quatre jours à San Francisco (en face de la WWDC d'Apple). Cette année, la grande conférence Java fut plus technique que les années précédentes. Outre J2EE 1.4 qui continue d'être implémentée par les éditeurs, il y a eu deux vedettes : J2SE 5.0 (Tiger) et le nouvel IDE Sun que l'on attendait avec impatience : Java Studio Creator (et plus discrètement NetBeans 4.0). La grande annonce de cette JavaOne est J2SE 5.0 (alias J2SE 1.5, alias Tiger). Pour le moment il s'agissait de montrer et de rendre disponible la seconde bêta, et d'introduire une centaine de nouvelles fonctions (voir notre dossier Java). L'autre grande sensation est la mise en Open Source du projet Looking Glass, l'interface 3D de Sun en Java. Cette annonce a un peu surpris, car on ne s'y attendait pas réellement. Cette technologie fonctionne pour le moment en X11 sous Linux/Unix et Solaris.Si dans la technologie smart card, Java ne fait pas de vagues, Sun est néanmoins très actif, pour preuve la disponibilité d'une préversion de Java Card Enterprise Software. Dans la mobilité et notamment dans les smartphones, Sun a conclu divers accords. L'un des plus importants est celui le liant à Nokia pour travailler sur la simplification du développement des jeux multi joueurs en Java pour les téléphones. Nokia intégrera (début 2005) dans son SNAP Mobile la plate-forme Java 2 Platform Micro Edition Wireless Toolkit. Côté entreprise, BEA a rendu disponible une version Solaris x86 de WebLogic. Scott McNealy durant sa keynote à rappelé certains chiffres et faits et alliances, notamment sur le changement de stratégie processeur avec Fujitsu, et sur son nouvel ami, Microsoft. Pour le patron de Sun, une des grandes opportunités est la sécurité utilisant Java Card. Il a aussi rappelé la stratégie autour de Linux et surtout sur le Java Desktop System. Une JavaOne plus pragmatique que l'année dernière, avec presque 200 sessions techniques et une vingtaine d'autres, axées sur la stratégie et le business.

Pertti Korhonen (département technologie de Nokia) durant la présentation des technologies Nokia

Site : http://java.sun.com/javaone/

MySQL 5 et les procédures stockées

Une des raisons de l'absence de MySQL dans le département IT en entreprise est le manque de procédures stockées. Aujourd'hui cet argument est sur le point de tomber, car MySQL 5 les incorpore. Mais il s'agit encore d'une version bêta…

Les procédures stockées représentent une suite d'instructions SQL précompilées qui sont "stockées" sur serveur, et plus précisément dans une table de la base de données. Elles sont ensuite exécutées par une application cliente externe, ou bien activées par un événement. SQL étant un langage non procédural, si vous n'utilisez pas de procédures stockées, vous devez porter votre code sur le poste client avec la non-portabilité que cela implique. Si vous écrivez du côté client une procédure contenant du SQL, le traitement du code est externe à la base de données. Par contre, si vous écrivez une procédure stockée, le traitement se déroule en "interne ". Non seulement, vous réduisez le trafic réseau entre le client et le serveur, mais ainsi vous obtenez un temps d'exécution quasi identique, quel que soit le client utilisé. Logiquement, en employant des procédures stockées vous augmentez les performances de vos applications.

La version 5 de MySQL, actuellement en bêta, ne comprend pas non plus les triggers, mais un grand nombre de mécanismes propres aux procédures stockées sont déjà fonctionnels. C'est ainsi que vous pouvez :

- déclarer des variables (DECLARE);
- construire des itératives (IF , ELSEIF, ELSE, END IF) ;
- élaborer des répétitives (WHILE, END WHILE) ;
- appeler une autre procédure stockée (CALL).

MySQL sous Windows

Récupérez l'archive MySQL et lancez l'installation (répertoire C:\MySQL par défaut). Si en exécutant WinMySQLAdmin vous n'arrivez pas à exécuter le service MySQL, c'est sans doute en raison de l'erreur "1067". Celle-ci se manifeste en cas de mauvaise déclaration du chemin vers MySQL, comme indiqué dans le fichier "my.ini". En fait si vous avez déjà installé MySQL par le passé (par exemple avec EasyPHP), le fichier "my.ini" initial n'a sans doute pas été écrasé : il conserve la trace d'un chemin erroné à MySQL. Vous pouvez dans ce cas éditer ce fichier "my.ini" avec WinMySQLAdmin pour le corriger (ou éditer directement avec le bloc-notes ce fichier C:\Windows\my.ini). Ce qui donne :

```
…
##basedir = C:/Program Files/EasyPHP1-7/mysql/
##datadir = C:/Program Files/EasyPHP1-7/mysql/data/
basedir = C:/mysql/
datadir = C:/mysql/data/
…
```

Ouvrez maintenant une invite de commande, et créez par exemple le répertoire de travail "PROC" ("C:\MySQL\PROC").

Création d'un jeu de données d'essai

Commençons par exécuter MySQL :

```
C:\mysql\PROC>C:\mysql\bin\mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 22 to server version: 5.0.0-alpha-nt
```

Le logiciel d'administration MySQL sous Windows WinMySQLAdmin.

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Le prompt "mysql›" apparaît. Il n'y a pas de mot de passe par défaut pour l'utilisateur root, mais évidemment vous pourrez toujours lui en attribuer un plus tard, en accédant à la base de données système "mysql" (et aussi créer un nouvel utilisateur). Créons maintenant une base de données "villes" :

```
mysql> CREATE DATABASE villes;
Query OK, 1 row affected (0.00 sec)
```

Vérifions qu'elle existe :

```
mysql> show databases;
```

Database
mysql
test
villes

3 rows in set (0.01 sec)

Ouvrons la base :

```
mysql> USE villes;
Database changed
```

Nous allons créer une nouvelle table par importation d'un fichier source. Sur le site de phpnuke-europe (http://www.phpnuke-europe.org) vous pouvez récupérer une archive des codes postaux de France, contenant les instructions SQL nécessaires à la création de la table des codes postaux, ainsi que l'insertion des enregistrements.

```
mysql>source C:\mysql\proc\villes_france_1.sql
```

Ce premier fichier contient les instructions SQL suivantes :

```
CREATE TABLE villes_france (
  commune varchar(35) NOT NULL,
  codepost varchar(5) NOT NULL,
  departement varchar(25) NOT NULL,
  latitude varchar(20),
  longitude varchar(20),
  altitude varchar(20),
  pop varchar(7),
  dens varchar(5),
  prefere char(1)
);

INSERT INTO villes_france VALUES ( 'L ABERGEMENT CLEMENCIAT',
'01400', 'AIN', '46.14948', '4.92189', '255', '579', '36', '');
...
```

Les deux fichiers SQL suivants contiennent la suite des INSERT :

```
mysql>source C:\mysql\proc\villes_france_2.sql
mysql>source C:\mysql\proc\villes_france_3.sql
```

Vérifions que notre table est valide :

```
mysql> show tables;
Tables_in_villes
  villes_france
```

1 row in set (0.03 sec)

Pour connaître le détail de la structure de la table, employez DESCRIBE villes_france;

Note :
Si l'erreur 'max_allowed_packet' se produit, vous devez modifier la valeur par défaut de la variable "max_allowed_packet".

```
set-variable = max_allowed_packet=16M
```

Créer la première procédure stockée

La création d'une procédure stockée se réalise par le biais de la commande CREATE PROCEDURE.

```
CREATE PROCEDURE MaProcedure ( [liste de paramètres] )
[Caractéristiques]
BEGIN
[corps]
END;
```

MySQL peut d'ailleurs créer également des fonctions stockées, en utilisant la syntaxe très proche CREATE FUNCTION.

```
CREATE FUNCTION MaFonction ( [liste de paramètres] ) RETURNS type
[Caractéristiques]
BEGIN
[corps]
END;
```

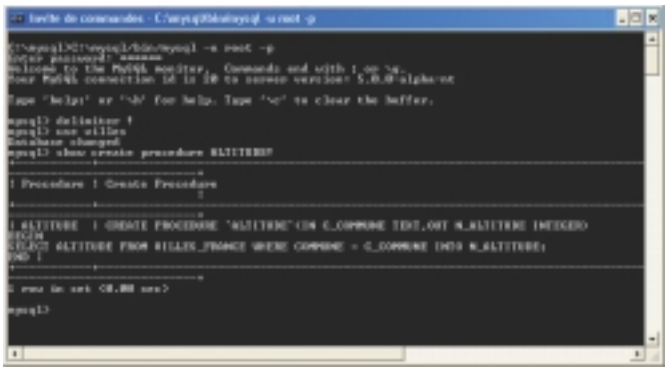
Les "caractéristiques " sont optionnelles. Vous avez une caractéristique de sécurité qui indique que l'exécution de la procédure ou fonction s'effectue avec les droits de son créateur (definer) ou de celui qui l'invoque (invoker).

```
slq security definer
```

Vous pouvez aussi ajouter un commentaire (mot-clé "comment "), préciser le langage utilisé (seulement SQL pour l'instant, mais un support PHP est prévu !) avec le mot-clé "language", et enfin indiquer qu'une fonction est déterministe ou non (si le résultat renvoyé est toujours le même, ce qui permet d'optimiser).

```
comment 'ceci est un commentaire'
language sql
not deterministic
```

Entre parenthèses, l'ensemble de ces caractéristiques est modifiable avec la commande ALTER.
Cette procédure stockée multiplie deux nombres passés en arguments. Une fonction serait plus judicieuse, mais nous allons en profiter pour mettre en évidence le passage à la procédure de plusieurs paramètres. Nous devons commencer par redéfinir le délimiteur utilisé par le langage SQL. En effet, c'est le point virgule qui est employé, à la fois par MySQL pour délimiter ces propres instructions, et par le langage SQL utilisé pour créer une procédure.



L'instruction SHOW CREATE PROCEDURE affiche le contenu de la procédure "ALTITUDE".

En indiquant par exemple

```
mysql> delimiter !
```

Vous spécifiez explicitement que la définition de la procédure stockée se terminera par un "!" (Vous pouvez utiliser n'importe quelle autre combinaison de caractères). Si vous ne réalisez pas cette démarche, MySQL affichera un message d'erreur à l'encodage de la ligne "SET NBR3 = NBR1 * NBR2;".

```
mysql> CREATE PROCEDURE MULTIPLIE(IN NBR1 INTEGER,IN NBR2
INTEGER,OUT NBR3 INTEGER)
    -> BEGIN
    -> SET NBR3 = NBR1 * NBR2;
    -> END
    -> !
Query OK, 0 rows affected (0.02 sec)
```

Nous pouvons vérifier son existence en affichant la liste des procédures stockées en invoquant la commande SHOW PROCEDURE STATUS !

```
mysql> SHOW PROCEDURE STATUS!
```

Name	Type	Definer	Modified	Created	Security_type	Comment
MULTIPLIE	PROCEDURE	root@localhost	0000-00-00 00:00:00	2004-06-09 14:17:44	DEFINER	

1 row in set (0.00 sec)

Notre procédure stockée est altérable par l'instruction ALTER qui mal-

heureusement ne peut que modifier ses caractéristiques (comme le commentaire associé par exemple) :

```
mysql> ALTER PROCEDURE MULTIPLIE
    -> COMMENT "ceci est un essai"
    -> ;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW PROCEDURE STATUS;
```

Name	Type	Definer	Modified	Created	Security_type	Comment
MULTIPLIE	PROCEDURE	oot@localhost	2004-06-09 14:27:35	2004-06-09 14:27:35	DEFINER	ceci est un essai

1 row in set (0.00 sec)

En fait la procédure stockée est, comme son nom l'indique, "stockée" effectivement dans une table système. Pour preuve, nous pouvons en éditer le champ "body" comme ceci :

```
mysql> use mysql;
Database changed
mysql> select body from proc where name ="MULTIPLIE";
```

body
BEGIN SET NBR3 = NBR1 * NBR2; END

1 row in set (0.00 sec)

```
mysql>
```

Pour appeler la procédure il est nécessaire de définir une variable RESULTAT. Chaque paramètre passé dans la procédure "multiplie" a été noté comme étant soit en entrée (IN), soit en sortie (OUT), ou en entrée/sortie (INOUT).

```
mysql> set @RESULTAT=0!
Query OK, 0 rows affected (0.03 sec)
```

Appelons maintenant notre procédure en lui passant les paramètres adéquats :

```
mysql> CALL MULTIPLIE(32,2,@RESULTAT)!
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> SELECT @RESULTAT!
```

@RESULTAT
64

1 row in set (0.00 sec)

```
mysql>
```

Une deuxième procédure stockée

Comme la procédure stockée est renseignée comme un élément unique d'une table, une tentative de générer une procédure stockée de même nom se soldera par un échec. C'est pourquoi si vous redéfinissez votre procédure stockée dans un fichier "client.sql", vous prendrez la précaution de vérifier au préalable son existence. Si c'est le cas, vous devrez d'abord l'effacer avec l'instruction DROP.

```
DROP PROCEDURE IF EXISTS MULTIPLIE!
CREATE PROCEDURE MULTIPLIE(IN NBR1 INTEGER,IN NBR2 INTEGER,OUT
NBR3 INTEGER)
...
```

Nous allons créer une procédure stockée qui renvoie l'altitude d'une commune. Nous allons utiliser le type "TEXT" pour passer à la procédure le nom de la commune (qui doit être dans ce cas unique ; sinon une erreur est générée : idéalement c'est donc le code postal (unique) qui devrait être passé).

```
mysql> CREATE PROCEDURE ALTITUDE(IN C_COMMUNE TEXT,OUT
N_ALTITUDE INTEGER)
    -> BEGIN
    -> SELECT ALTITUDE FROM VILLES_FRANCE WHERE COMMUNE =
C_COMMUNE INTO N_ALTITUDE;
    -> END
    -> !
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

Essayons avec le nom de la commune "MERLETTE" :

```
mysql>SET @COMMUNE="MERLETTE"!
mysql>SET @ALTITUDE=0!
mysql>CALL ALTITUDE(@COMMUNE,@ALTITUDE)!
Query OK, 0 rows affected (0.03 sec)
```

Et affichons le résultat obtenu :

```
mysql> SELECT @ALTITUDE!
```

@ALTITUDE
1925

1 row in set (0.00 sec)

Et ainsi de suite :

```
mysql> SET @COMMUNE="TOULOUSE";!
Query OK, 0 rows affected (0.00 sec)
mysql> CALL ALTITUDE(@COMMUNE,@ALTITUDE)!
Query OK, 0 rows affected (0.04 sec)
mysql> SELECT @ALTITUDE!
```

@ALTITUDE
144

1 row in set (0.00 sec)

D'autres instructions utiles

Nous avons examiné CREATE PROCEDURE, CREATE FUNCTION, ALTER PROCEDURE, ALTER FUNCTION, DROP PROCEDURE, DROP FUNCTION, SHOW CREATE PROCEDURE, SHOW CREATE FUNCTION, SHOW PROCEDURE STATUS, SHOW FUNCTION STATUS, CALL, BEGIN ... END, DECLARE, SET, SELECT ... INTO

D'autres instructions sont indispensables dès que l'on veut pousser plus avant la programmation d'une procédure stockée. Par exemple : il est on ne peut plus courant de lancer une itérative de traitement sur un curseur... Pour ouvrir un curseur, vous utiliserez l'instruction OPEN, tandis que l'instruction FETCH lira le prochain enregistrement de ce curseur. Enfin, l'instruction CLOSE le fermera.

Pour contrôler le flux d'exécution, vous avez à votre disposition la structure classique IF/,ELSEIF,ELSE/END IF. De même que la structure CASE ... WHEN. Pour initier une itérative, outre WHILE, vous pouvez utiliser REPEAT ou LOOP (avec LEAVE et ITERATE).

■ Xavier Leclercq
Xavier.Leclercq@programmez.com

Programmer le planificateur de tâches de Windows avec C++

SOURCES DE L'ARTICLE
WWW.PROGRAMMEZ.COM

Habituellement utilisé via le panneau de configuration, le planificateur de tâches de Windows expose une API puissante dont le programmeur peut tirer profit... s'il est familiarisé avec COM.

Comme son nom l'indique, le planificateur de tâches de Windows permet de planifier l'exécution d'applications. Il est en quelque sorte le pendant du démon cron (ou at) des systèmes UNIX. A cette différence près que cron se programme avec un script nommé crontab tandis que le planificateur de Windows expose une API sous la forme d'objets COM. Une application peut trouver pertinent de programmer une tâche à intervalles réguliers, comme une recherche de mises à jour, ou ouvrir une connexion Internet à heures fixes, pour ne prendre que ces exemples. Pour cela nous devons être à l'aise avec COM. Nous allons donc voir au cours de cet article les grandes lignes de l'architecture COM et les règles de programmation qui la concerne, puis nous appliquerons cela au planificateur de tâches.

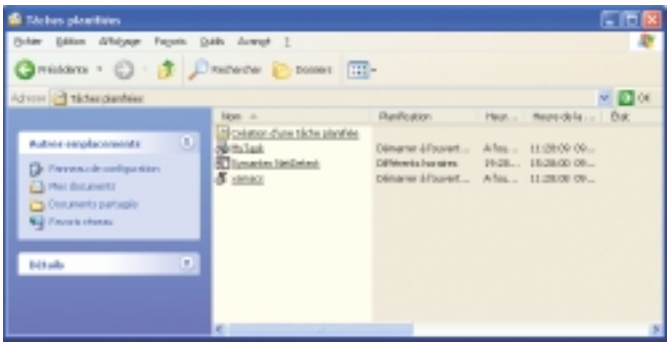


Fig 1 : Le planificateur de tâches, vu depuis le panneau de configuration.

L'architecture COM

Il est (fort heureusement...) possible de travailler avec COM sans en connaître le détail, mais seulement les grandes lignes que voici. COM (Component Object Model) résout le problème de la communication de composants logiciels entre eux. Les composants n'ont pas besoin d'être écrits dans le même langage, mais seulement de supporter les pointeurs et pointeurs de fonctions. Ainsi, vous pouvez faire du COM avec C, C++, Pascal, etc. (Microsoft a créé Automation, une sorte de surcouche à COM pour les langages de haut niveau sans pointeurs, tels que VBA ou Python). Un composant logiciel COM est une classe (pas nécessairement au sens C++ du terme) qui expose des interfaces, qui elles-mêmes exposent des pointeurs de fonctions organisés en tables. Les interfaces dérivent, au sens programmation objet du terme, d'au moins une interface : IUnknown. Une classe COM est un agrégat d'interfaces (Fig 1). Au passage, puisque l'on parle d'interfaces, d'héritage, de tables de pointeurs de fonctions et d'agrégat, on saisit que C++ est un langage tout trouvé pour manipuler COM, les tables de pointeurs se transformant comme par magie en tables de fonctions membres de classes. Les interfaces sont contractuelles. Cela signifie qu'à partir du moment où un objet a été documenté, son implémentation peut être changée, mais pas les tables de pointeurs de fonctions exposées par les interfaces.

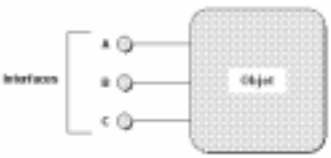


Fig 1 : Un objet COM est un agrégat d'interfaces contractuelles.

La tuyauterie COM

COM met en place toute une plomberie pour que les composants logiciels puissent communiquer. Cela peut consister en un simple chargement d'une dll dans l'espace d'adressage du client qui utilise l'objet COM, ou en marshalling de paramètres entre processus séparés. Ceci est quasiment transparent au programmeur côté client, qui utilise un jeu d'API très limité :

- CoInitialize met en place la plomberie COM.
- CoGetClassObject et IFactory::CreateInstance pour obtenir respectivement un pointeur sur l'interface fabrique de classe de l'objet, puis instancier l'objet et obtenir un pointeur sur son interface pas défaut. Compliqué ? Alors, nous utiliserons simplement l'API CoCreateInstance à elle seule, qui fait le travail précédent.
- CoUninitialize débranche la plomberie COM et libère les ressources.

Remarque : Nous ne donnons que les grandes lignes dans cet article, pour approfondir COM, consultez le site MSDN en ligne à http://msdn.microsoft.com.

Que l'on emploie CoCreateInstance ou les autres APIs il faut fournir un

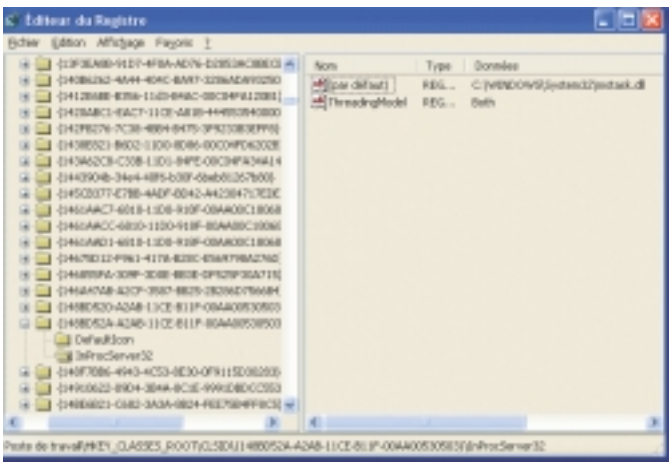


Fig 2 : Comme tout objet COM, le planificateur de tâches est répertorié dans la base de registre.

GUID, ou identificateur unique, qui est un nombre codé sur 128 bits. Ce nombre est généré au cours du développement du composant COM par des APIs dédiées et à partir du numéro de série de la carte mère de la machine de développement, de l'heure, de la vitesse du vent, de l'âge du capitaine... et autres paramètres ésotériques, de telle façon que l'unicité du GUID soit garantie. Ensuite, quand un composant COM est installé sur une machine, la relation entre le GUID et la localisation de la dll, de l'exe, ou autres informations, est inscrite dans la base de registre (Fig 2). CoCreateInstance fait une recherche dans la base de registre pour être en mesure de charger le composant en mémoire et fournir à l'application appelante un pointeur sur l'interface par défaut du composant (Fig 3). En outre, un GUID est enregistré pour chacune des interfaces du composant. Le programmeur a besoin de connaître ces GUIDs pour travailler. En principe ils sont fournis par la documentation. Sinon dans les fichiers en-têtes. Pour notre planificateur c'est le fichier mstask.h.

Fig 3 : Après appel à CoCreateInstance, une application obtient un pointeur sur l'interface principale (ou par défaut) de l'objet COM.

L'interface IUnknown

Toute interface dérive obligatoirement de IUnknown, ce qui signifie que les 3 méthodes de cette dernière sont systématiquement implémentées. En C++, la déclaration de IUnknown pourrait ressembler à ça :

```
struct IUnknown {
   virtual   HRESULT   QueryInterface(IID& iid, void** ppvObj) = 0;
   virtual   ULONG   AddRef() = 0;
   virtual   ULONG   Release() = 0;
 };
```

QueryInterface permet d'obtenir un pointeur sur n'importe quelle autre interface faisant partie de l'agrégat. La valeur retournée n'est pas le pointeur, mais un code de retour, signifiant le succès ou l'échec de l'opération.

Le pointeur est fourni dans ppvObj, qui est un pointeur sur un pointeur void. Microsoft ne fournit pas l'aspirine, mais Programmez! fournit une classe sympathique devant vous éviter les maux de tête (voir plus loin). En principe QueryInterface est le seul moyen utilisé pour obtenir ces fameux pointeurs d'interface. Hormis bien sûr CoCreateInstance pour l'interface par défaut, mais cette API passe par QueryInterface en sous- main.

Quel que soit le moyen par lequel le pointeur d'interface a été obtenu (QueryInterface, ou CoCreateInstance), AddRef a été automatiquement invoquée pour incrémenter un compteur de références. Le compteur est décrémenté par des appels à Release, mais cet appel n'est pas automatique.

Au contraire, il est à la charge du programmeur. Quand le compteur tombe à zéro, l'objet est détruit par COM et les ressources sont libérées. Cette contrainte d'invocation de Release, alliée à la gestion du code de retour HRESULT conduit souvent à du code épouvantablement structuré, comme on peut le voir dans la plupart des exemples officiels de la MSDN. Par exemple, nous pouvons avoir, en pseudo-code :

```
CoInitialize()
CoCreateInstance(Class c, pointeur p)
        Si échec CoUnitialize()
        sinon
        QueryInterface(Interface i, pointeur p1)
        si échec
                p->Release()
                CoUninitialize()
        sinon
QueryInterface(Interface i1, pointeur p2)
        si échec
                p1->Release()
                p->Release()
                CoUninitialize()
```

ce qui conduit à un code volumineux, rapidement inextricable, et présente un gros risque de voir une ressource non libérée, par erreur.

Un Pointeur intelligent

Puisque nous travaillons avec C++ nous allons profiter de sa puissance en écrivant une classe de pointeur COM intelligent. Le code est donné (encadré 1). Il s'agit d'une classe générique (template) classique, baptisée TSmartComPtr, et qui encapsule un pointeur d'interface quelconque, fourni par les APIs. Nous fournissons un constructeur de copie et un opérateur d'affectation qui prennent bien sûr soin d'invoquer la méthode de AddRef de l'interface manipulée. Bien évidemment, car c'en est tout l'intérêt, le destructeur de notre pointeur intelligent invoque Release. Enfin, pour que l'utilisation de notre pointeur intelligent soit la plus aisée et la plus transparente possible, nous fournissons un opérateur de sélection de membre (->) et un opérateur & retournant un pointeur sur le pointeur d'interface à l'intention des APIs. On note que l'opérateur de sélection de membre lève une exception construite autour du code d'erreur COM E_POINTER si le pointeur d'interface est invalide. La classe d'exception est également fournie et est aussi utilisée dans les exemples, si la valeur de retour HRESULT n'est pas convenable. La classe connaît les codes d'erreur COM les plus courants, mais rien ne vous empêche de l'étoffer. Tout ce code est dans le fichier comutil.h accompagnant les exemples, et les fichiers sont sur le site www.Programmez.com. Nous sommes maintenant outillés pour écrire un code client COM propre, sans plus avoir à nous soucier de la libération des interfaces. Exemple :

```
if(::CoInitialize(NULL) != S_OK)
   return 0;

try
{
   HRESULT hr;

   TSmartComPtr<IObjectCom> oc;
   TSmartComPtr<IAutreInterface> iai;

   hr = ::CoCreateInstance(CLSID_ObjetCom,
                           NULL,
                           CLSCTX_INPROC_SERVER,
```

```
                    IID_IObjetCom,
                    (void **)&oc);
  TestOk(hr);

  hr = oc->QueryInterface(IID_IAutreInterface, (void **)&iai);
  TestOk(hr);

  hr = iai->FaireQuelqueChose()
  TestOk(hr);
 }
 catch (TComException ce)
 {
  ce.Show();
 }

 ::CoUninitialize();
```

Du point de vue des API Windows, les pointeurs intelligents sont utilisés comme des pointeurs classiques, mais si l'on sort de la portée du bloc try normalement, ou si une exception est levée par TestOk les interfaces sont automatiquement libérées. Dans nos exemples la fonction TestOk est tout simplement :

```
void TestOk(HRESULT r)
{
 if(r != S_OK)
 {
  TComException ce(r);
  throw ce;
 }
}
```

Démarrer le planificateur de tâche

Nous allons maintenant mettre les mains dans le cambouis du planificateur de tâche. Pour programmer celui-ci, deux choses sont requises. D'abord, pour la plupart des appels aux APIs, vous devez disposer des droits de l'administrateur. Si le code de retour HRESULT contient E_ACCESSDENIED, c'est certainement votre problème. Ensuite, pour être programmé, le planificateur doit tourner sur le système, ce qui n'est pas du tout garanti par défaut, surtout sur les Windows 9x. Il est possible de lancer le planificateur par programmation, en dépit de la difficulté présentée par le fait que le planificateur est un service sous Window NT, 2000 et XP et un processus sous 9x. Vous trouverez donc sur le Cd-Rom un source nommé starttaskscheduler.cpp, directement adapté d'un source de la MSDN, et qui devrait vous simplifier le travail. À propos des sources, ceux-ci ont été écrits et compilés sous C++Builder. Ils ont également été compilés sous Visual C++ avec le compilateur de Microsoft et le compilateur Intel C++ 7.1. Pour ces compilateurs (ou d'autres) vous devrez peut-être rajouter :

```
#include <initguid.h>
```

Cette inclusion n'étant pas nécessaire avec les compilateurs Borland. Je n'ai pas été en mesure de compiler avec MinGW Developper Studio et le compilateur MinGW (gcc version Windows) en raison de l'absence, sauf erreur de votre serviteur, de l'entête mstask.h.

Organisation interne du planificateur de tâches

Du point de vue du programmeur, le planificateur est constitué de 4 classes COM, ou si l'on préfère, de 4 agrégats d'interfaces (Fig 4).

Seule la classe principale 'Task Scheduler Object' s'obtient de manière COM classique, via l'API CoCreate Instance. Les objets 'Enumération Objet' et 'Task Object' sont instanciés, via les méthodes de l'interface ITaskScheduler. 'Trigger Object', quant à lui, est obtenu via une instance de Task Objet. On comprend donc que 'Task Scheduler' chapeaute le tout, que 'Enumération' comme son nom le suggère permet de parcourir les tâches déjà planifiées, que Task gère une tâche élémentaire et qu'une tâche est exécutée par le système, à condition qu'un 'Trigger' (déclencheur) lui soit attribué.

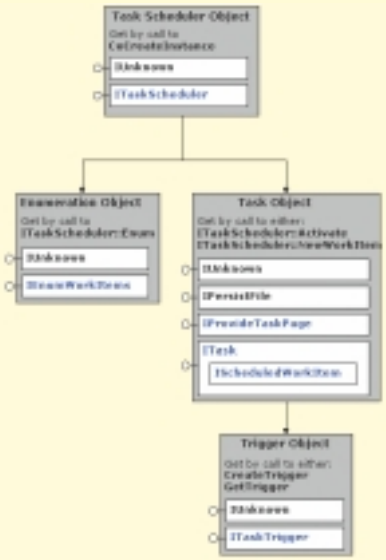


Fig. 4 : L'organisation interne du planificateur de tâches. D'après un document MSDN.

Énumérer les tâches.

Rien de plus simple que de créer une tâche, à condition que celle-ci n'existe pas déjà, auquel cas la création échoue. Nous proposons donc enumeration.cpp pour énumérer les tâches existantes. Celles-ci sont retournées sous forme d'une chaîne unicode (comme toute chaîne COM) augmentée de l'extension .job. En effet, pour chaque tâche, Windows crée un fichier dans le répertoire C:\Windows\Tasks, et c'est le nom de ce fichier qui est effectivement retourné. La méthode Next de l'objet d'énumération à un comportement très particulier. On peut lui demander de parcourir les tâches une à une, comme dans notre exemple, ou par paquet de N tâches. Dans tous les cas, le résultat est placé dans un tableau de pointeurs sur des chaînes Unicode. Toute chaîne retournée par le planificateur doit être invariablement libérée par l'appel à l'API Windows CoTaskMemFree. Le mot Task dans le nom de cette API n'est pas systématiquement associé aux objets 'Task'. Plus généralement, toute entité consommant des ressources mémoire et allouée par le planificateur doit être libérée avec cette API à tout défaire :-). Ainsi dans le cas qui nous occupe, le programmeur a la charge de libérer chaque élément du tableau, puis le tableau lui même. C'est ce qui explique les deux appels consécutifs à CotaskMemFree dans l'exemple. En outre, la structure du code de l'exemple ne fonctionne bien que parce que nous demandons les tâches une à une. En effet si les tâches étaient demandées, par exemple, trois par trois, il pourrait arriver que le tableau contienne les deux dernières tâches et qu'en même temps, l'API signale qu'il n'y a plus de tâches. En ce cas, notre code conduirait à une fuite mémoire. Pour travailler par lots, il faut passer à Next, un pointeur sur un ULONG, qui sera renseigné avec le nombre d'éléments effectivement contenus dans le tableau.

Créer une tâche

C'est tout à fait trivial (c.f createtask.cpp). On instancie un objet Task, via la méthode NewWorkItem de ITaskScheduler. Nous noterons deux

Fig. 5 : Notre tâche, crée et commentée.

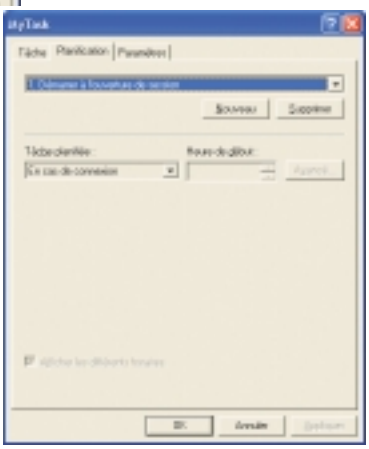


Fig. 6 : Notre tâche se déclenchera à chaque connexion au système.

choses. D'abord, nous travaillons avec COM, donc le nom de la tâche à créer doit être dans une chaîne unicode, et nous n'ajoutons pas l'extension .job. Ensuite, créer la tâche ne suffit pas en soi. Il faut sauver le fichier qui la décrit sur le disque. Pour cela, l'objet Task implémente une interface COM standard : IPersist. Tout ce que nous avons à faire est d'obtenir un pointeur sur cette interface de façon classique, c'est-à-dire via QueryInterface de notre objet Task ou (c'est une particularité de COM) via QueryInterface de n'importe quelle autre interface de l'objet Task dont nous aurions un pointeur. Une fois IPersist obtenue, on invoque sa méthode Save et le tour est joué. Nous n'avons absolument pas à nous préoccuper du contenu du fichier.

Créer un déclencheur

Sans déclencheur, une tâche ne fait rien. En revanche une tâche peut se voir attribuer plusieurs déclencheurs et donc s'exécuter dans des circonstances diverses. Pour créer un déclencheur, il faut d'abord activer la tâche, via Activate de ITaskScheduler. Le mot activation ne signifie rien d'autre que d'informer le planificateur que l'on travaille sur cette tâche précise. Ceci fait, il est obligatoire d'attribuer un compte utilisateur, soit un nom d'user et le mot de passe à la tâche. Le programme d'exemple, triggercomment.cpp ne compilera pas si vous ne fournissez pas ces informations dans le source. Ensuite, on définit l'application à lancer, soit le Notepad pour notre exemple (Fig 5), puis on pose un commentaire facultatif. Enfin, on renseigne une structure TASK_TRIGGER que l'on attribue à la tâche. Dans notre exemple, la tâche se déclenche au login (Fig 6). Vous devez donc voir surgir le Notepad à votre prochaine connexion. Bien sûr on sauve le tout avec IPersist, comme expliqué précédemment. Le principal est dit. Vous pourrez approfondir sans peine la programmation du planificateur de tâches sur la base de cet article et de la documentation de Microsoft (http://msdn.microsoft.com recherche pour Task Scheduler API). Quant à nous, n'oublions pas de planifier de nous retrouver pour d'autres aventures au coeur de Windows.

Encadré 1

Un pointeur COM intelligent

```
template <class T> class TSmartComPtr
{
public:
  TSmartComPtr::TSmartComPtr()
  {
    p = NULL;
  }

  virtual ~TSmartComPtr()
  {
    if(p)
      p->Release();
  }

  TSmartComPtr(const TSmartComPtr<T>& param)
  {
    p = param.p;
    if(param.p)
      param.p->AddRef();
  }

  T& operator*()
  {
    return *p;

  }

  T** operator&()
  {
    return &p;
  }

  T* operator->()
  {
    if(!p)
    {
      TComException ce(E_POINTER);
      throw ce;
    }
    return p;
  }

  T* operator=(const TSmartComPtr<T>& param)
  {
   if(param.p)
      param.p->AddRef();

    if(p)
      p->Release();

    p = param.p;
  }

private:
  T *p;
};
```

■ *par* **Frédéric Mazué,** *fmazue@programmez.com*

Synchronisation de données sur Pocket PC avec la méthode Remote Data Access

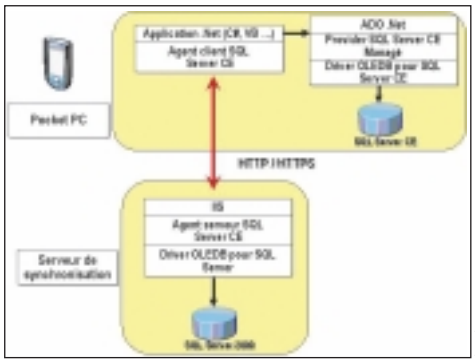
Un cas fréquent d'utilisation d'applications mobile : le commercial qui, en clientèle, saisit ses commandes en mode déconnecté sur son Pocket PC, puis synchronise ses données ultérieurement vers un serveur centralisé. Pour répondre, entre autres, à ce besoin, deux mécanismes sont proposés par le .Net Compact Framework pour le transfert de données : le Remote Data Access (RDA) et Merge Replication. Nous allons dans cet article étudier comment synchroniser des données entre une base de données embarquée SQLCE et une base de données centralisée SQLServer 2000, en utilisant la méthode RDA.

Nous parlons ici de l'architecture RDA, des transferts d'une base SQL Server 2000 vers SQL CE, soumission de données modifiées de SQL CE vers SQL Server 2000 et enfin, soumettre les ordres SQL de SQL CE vers SQL Server Server 2000.

Pour réaliser ce développement, nous avons utilisé :

- VS.NET 2003
- .NET Compact Framework
- Smart Device Extensions
- Serveur IIS
- SQL Server CE 2.0 SP3 & SQL Server 2000 SP3
- Active 3.7

L'architecture Remote Data Access (RDA)



L'architecture Remote Data Access

RDA permet la communication de périphériques mobiles exécutant SQLServer CE, avec une base de données distante SQL2000, via les protocoles Internet standard (HTTP, HTTPS). Ce fonctionnement présente plusieurs avantages :

- La possibilité de se connecter à SQL2000 avec une simple liaison Internet
- La possibilité de bénéficier des mécanismes d'authentification et d'autorisations IIS et de communiquer à travers des firewalls
- La possibilité de crypter des données en utilisant un protocole HTTPS

Le produit SQLCE comprend trois composants :

- le moteur SQL Server CE : déployé sur l'appareil mobile, il permet de gérer les données (création des tables et des index, surveillance des modifications apportées à la base de données SQLCE)
- l'agent client SQL Server CE : réside sur l'appareil mobile sous la forme d'un composant nommé " ssceca20.ll ". Son rôle est d'interagir avec la base de données SQLServer CE locale, à travers les méthodes proposées par l'objet SqlCeRemoteDataAccess, et de communiquer les modifications apportées aux données à l'agent serveur SQLServer CE via IIS et HTTP/S.
- l'agent serveur SQL Server CE : composant installé sur le serveur de synchronisation (sscesa20.dll). Il reçoit les requêtes HTTP transmises par l'agent client, puis fait appel au provider OLEDB SQLServer pour les exécuter.

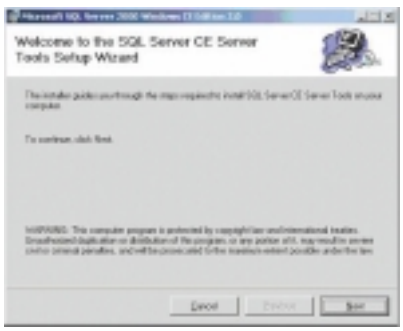
L'agent serveur combine trois rôles :

- Une extension ISAPI résidant dans un répertoire virtuel spécial IIS, à créer pour faire fonctionner l'architecture RDA
- Son nom fait partie d'une URL à travers laquelle les agents clients et serveurs peuvent communiquer.
- Un programme SQL2000 capable de se connecter et d'interagir avec une base de données SQL2000

Installation et paramétrage de SQLServer CE

Afin de mettre en œuvre l'architecture RDA, il faut d'abord installer SQL Server CE 2.0 sur le serveur hébergeant IIS. Il est nécessaire pour cela de synchroniser la version de SQLServer CE avec la version du service pack d'SQL2000. Si vous disposez de SQL2000 SP3 ou SP3a alors, il est nécessaire de télécharger le fichier sqlce20sql2ksp3a.exe à l'adresse suivante http://www.microsoft.com/sql/ce/downloads/sp3.asp.

L'installation débute par le lancement du fichier sqlce20sql2ksp3a.exe.

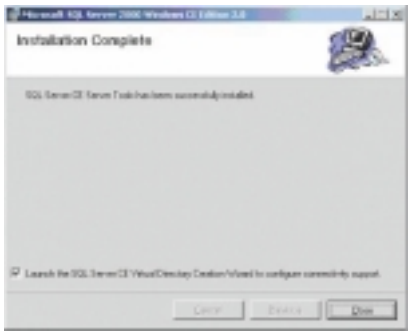


Ecran d'accueil de la procédure d'installation SQL Server CE

L'assistant nous demande où nous souhaitons installer les composants SQLServer CE.

Ecran de sélection du répertoire d'installation SQL Server CE

L'installation des composants est achevée.



Ecran de fin d'installation SQL Server CE

Passons désormais à la création du répertoire virtuel IIS et au paramétrage des permissions HTTP et NTFS. Pour ce faire, SQL Server CE nous propose un assistant.

Ecran de lancement du Wizard Sql Server CE

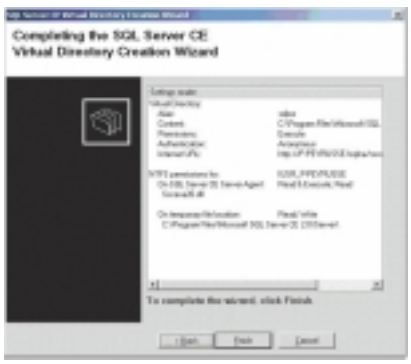
Le premier écran est chargé de la création du répertoire virtuel et nous demande de saisir le nom du répertoire virtuel à créer et le chemin associé. Ce répertoire contiendra le fichier sscesa20.dll correspondant à notre filtre ISAPI.

Ecran de paramétrage du répertoire virtuel Sql Server CE

L'écran suivant nous demande de paramétrer les permissions d'accès à notre répertoire virtuel. Nous utiliserons un accès anonyme, via le compte IIS de notre serveur de synchronisation.

Ecran de paramétrage des permissions du répertoire virtuel Sql Server CE

Le dernier écran nous affiche un récapitulatif des paramètres définis. Il est également possible de modifier ce paramétrage ultérieurement grâce à la console SQLServer Ce Connectivity Management qui se trouve dans le répertoire d'installation des composants SQLServer CE.



Ecran récapitulatif des paramètres de configuration Sql Server CE

Mise en œuvre de RDA

Le modèle objet RDA est accessible depuis la classe SqlCeRemoteDataAccess de l'espace de noms System.Data.SqlServerCe. Cette classe propose trois méthodes Pull, Push et SubmitSQL utilisées dans le processus de synchronisation que nous allons décrire plus précisément.

Elle présente également les propriétés suivantes, qu'il faut définir avant tout appel à une méthode de la classe.

Propriété	Description
InternetLogin	Login IIS pour se connecter à l'agent SQLServer CE Serveur
InternetPassword	Mot de passe IIS pour se connecter à l'agent SQLServer CE Serveur
InternetUrl	URL utilisée pour se connecter à l'agent SQLServer CE Serveur
LocalConnectionString	Chaîne de connexion locale pour se connecter à SQLServerCE

Pour la réalisation de nos tests, nous utiliserons une base de données SQL2000 nommée " Client " qui dispose d'une table " commandes ".

La méthode Pull

La méthode Pull permet de récupérer des tables, des index et des données d'une base de données distantes SQLServer 2000 et de les transférer vers une base SQLCE de notre Pocket PC. Elle permet de ne récu-

pérer qu'une seule table à la fois et ne peut donc être utilisée pour agréger des données. De ce fait, il est recommandé d'effacer toute table locale à récupérer par un appel à la méthode Pull, sous peine d'obtenir un message d'erreur. Dans le cas où de nombreuses données devraient être rapatriées sur le Pocket PC, il peut être intéressant, pour des raisons de performances, de les regrouper dans une seule et même table. L'exemple suivant met en œuvre l'utilisation de la méthode Pull. Cette méthode prend en paramètre le nom de la table à créer localement, la requête permettant la récupération des données, la chaîne de connexion distante, une option de suivi des modifications apportées à SQLCE et une table de stockage des erreurs. Précisons que nous créons également la base de données locale SQLCE (commandes.sdf) si celle-ci n'existe pas.

Attention, l'option de suivi RdaTrackOption doit être positionnée si vous souhaitez mettre à jour la table extraite sur le périphérique mobile et renvoyer par un push les enregistrements modifiés vers la table SqlServer source.

```
...
using System.Data.Common;
using System.Data.SqlServerCe;
using System.Data.SqlClient;

...
public void RecupererTableCommande()
{
SqlCeConnection Cnx = null;
   SqlCeRemoteDataAccess Rda = null;
   SqlCeEngine Engine = null;
   string sConnexionLocale = @"Data Source=\My Documents\
commandes.sdf";
   string sConnexionDistante = "Provider=sqloledb; Data Source=192.
168.0.1;Initial Catalog=Client; " +        "User Id=sa;Password=leduke";


            try
            {
                if (!File.Exists("\\My Documents\\commandes.sdf"))
                {
                    Engine = new SqlCeEngine(sConnexionLocale);

                    Engine.CreateDatabase();
                    Engine.Dispose();
                }
                else
                {
                    Cnx = new SqlCeConnection(sConnexionLocale);
                    Cnx.Open();
                    SqlCeCommand Cmd = Cnx.CreateCommand();

                    Cmd.CommandText = "DROP TABLE Commandes";
                    Cmd.ExecuteNonQuery();

                    if (Cnx.State != ConnectionState.Closed) Cnx.Close();
                }

                Rda = new SqlCeRemoteDataAccess();

                Rda.InternetLogin          = "";
                Rda.InternetPassword       = "";
                Rda.InternetUrl        = "http://192.168.0.1/sqlce/sscesa20.dll";
                Rda.LocalConnectionString  = sConnexionLocale;
                Rda.Pull("commandes",
                    "Select * from commandes",
                    sConnexionDistante,
                    RdaTrackOption.TrackingOnWithIndexes ,
                    "ErrorTable");
            }
            catch (SqlCeException sqlex)
            {
                foreach(SqlCeError sqlError in sqlex.Errors)
                {
                    MessageBox.Show(sqlError.Message + " " +sqlError.
Source + " " + sqlError.NativeError + " " + sqlError.HResult);
                    MessageBox.Show(sqlError.Message,"Erreur");
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message,"Erreur");
            }
            finally
            {
                Rda.Dispose();
            }

}
```

La photo suivante permet de visualiser le contenu de la table commandes rapatriée au sein de l'analyseur de requêtes SQLCE de notre Pocket PC.

La photo suivante permet de visualiser le contenu de la table commandes rapatriée au sein de notre application autonome Pocket PC développée.

Récupération des données dans un datagrid sur Pocket PC

La méthode Push

La méthode Push permet de renvoyer vers le serveur SQL2000 les données qui ont été modifiées localement. Elle prend en paramètre le nom de la table à renvoyer, la chaîne de connexion distante et une option de batching. Celle-ci indique si chacune des lignes doit être transmise au sein d'une transaction unitaire ou si l'ensemble des lignes l'est, au sein d'une transaction unique.

```
public void SynchroniserTableCommande()
{

SqlCeRemoteDataAccess Rda = null;

string sConnexionLocale = @"Data Source=\My Documents\commandes.sdf";
string sConnexionDistante  = "Provider=sqloledb; Data Source=192.168
.0.1;Initial Catalog=Client; " +
                "User Id=sa;Password=leduke";

            try
            {
                Rda = new SqlCeRemoteDataAccess();

                Rda.InternetLogin        = "";
                Rda.InternetPassword   = "";
                Rda.InternetUrl         = "http://192.168.0.1/sqlce/sscesa20.dll";
                Rda.LocalConnectionString  = sConnexionLocale;
                Rda.Push("commandes",sConnexionDistante,RdaBatch
Option.BatchingOn);
            }
            catch (SqlCeException sqlex)
            {
                foreach (SqlCeError sqlError in sqlex.Errors)
                {
                    MessageBox.Show(sqlError.Message,"Erreur");
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message,"Erreur");
            }
            finally
            {
                Rda.Dispose();
            }
}
```

La méthode SubmitSql

L'utilisation de la méthode SubmitSQL permet de transmettre directement des données vers un serveur SQL2000 distant. Précisons qu'elle ne permet pas de renvoyer des lignes.

```
string sRequete = "INSERT INTO Commandes (OrderId,CustomerId,
OrderDate,ProductName) values (20000,'PEYRU','01/06/2004','Sirop
derable')";
string sConnexionDistante  = "Provider=sqloledb; Data Source=192.168
.0.1;Initial Catalog=Client; " +
                "User Id=sa;Password=leduke";



...

Rda.SubmitSql(sRequete,sConnexionDistante);
...
```

Problèmes / Solutions

Deux problèmes techniques sont survenus durant la réalisation des tests. Le premier problème est un problème de connexion IIS entre le Pocket PC et le serveur de synchronisation. Le message d'erreur apparaissant est : " A request to send data to the computer running IIS has failed. Erreur 28307 -2147012867 ". L'installation d'une carte de bouclage Microsoft (MS TCP Loopback) et l'attribution d'une adresse IP fixe ont permis de résoudre ce problème.

http://support.microsoft.com/default.aspx?scid=kb;EN-US;236869 "HOW TO: Install Microsoft Loopback Adapter in Windows 2000"
http://support.microsoft.com/default.aspx?scid=kb;en-us;839013 "How to install the Microsoft Loopback adapter in Windows XP"

Le second problème concerne l'impossibilité de visualiser et de pouvoir accéder à la méthode " Fill " de l'objet SqlCeDataAdapter. En fait, il suffit de faire référence à l'espace de noms " system.data.common " pour le résoudre.

■ Christian PEYRUSSE

Tech-Ed 2004

Retour vers le futur

Déjà un an que la dernière Tech-Ed s'était tenue à Barcelone. Cette fois-ci, ce fut à Amsterdam, durant la dernière semaine du mois de juin. Des rumeurs ici ou là, des attentes, finalement, sans doute une conférence développeurs un peu en demi-teinte pour les annonces, mais avec des précisions sur plusieurs futurs produits.

Comme toujours l'ambiance est détendue, mais studieuse, le nombre des conférences donnait parfois le vertige et il fallait souvent faire des choix cruels. Plusieurs milliers de développeurs, journalistes et responsables ont répondu présents dans le grand centre des congrès d'Amsterdam, le RAI. Les fêtes furent aussi à la hauteur de la réputation.

La première journée débuta tranquillement à 9h30, la keynote démarrant un peu plus tard. En attendant, on pouvait profiter de son tambour (offert à tous les participants) en compagnie d'un groupe venu sur scène pour chauffer la salle.

La keynote était attendue. Pour patienter, on avait droit à un concert participatif de Jonathan Murray. Durant 90 minutes, Jonathan Murray de EMEA, nous a présenté en détail la stratégie cycle de vie de Microsoft. La grande annonce fut celle des déclinaisons Express d'outils, dont Visual Studio .NET, SQL Server 2005. Les versions bêta sont disponibles en téléchargement. Le but avoué est de cibler l'amateur averti, le développeur indépendant ou de petites structures ne souhaitant pas acquérir la version "pro" des outils. Il s'agit pour Microsoft d'étendre l'audience de ses outils de développement. D'autres éditeurs proposent déjà ce type de version légère. Ces versions Express sont architecturées autour des prochains outils que sont VS.NET 2005 et Yukon. Sur VS.NET 2005, on s'attendait à voir apparaître dès la Tech-Ed une nouvelle version bêta, mais ce fut finalement la version sortie lors de la Tech-Ed US que les participants ont reçue. La nouvelle préversion étant disponible quelques jours plus tard en téléchargement ! On a pu assister à quelques démos sous Visual Studio 2005 : Team System et surtout le nouvel outil de développement pour les applications mobile (autre axe important pour Microsoft).

Les éditeurs présentaient leurs solutions. Ici, Zantaz, gestion de stockage pour Exchange.

Entreprise, Mobile, Xtreme Programming

Pour la période 2006-2008, on connaît déjà quelques éléments : Longhorn et ses déclinaisons, Orcas (le futur Visual Studio), Office 12. Pour le côté entreprise, la sécurité demeure un thème récurrent de Microsoft. On aura pu constater le virage SOA de l'éditeur qui l'a mis à toutes les sauces. L'intégration fut un autre credo durant la keynote. Il fut ainsi rappelé le partenariat avec SAP. Un rappel fut fait autour du Dynamic System Initiative. Il s'agit de déployer automatiquement les applications et les ressources, et de pouvoir bénéficier d'un management dynamique. En 2005, on aura droit à la sortie de Virtual Server 2005 qui se faisait attendre depuis de très longs mois. L'accent a aussi été mis sur la productivité en entreprise et sur l'utilisateur personnel.

Autre méthode de développement mise en avant par Microsoft, l'Extreme programming. Le 64 bits et la migration du code actuel furent abordés, notamment dans des ateliers. Sur Longhorn, rien de particulier n'a été mis avant. Il était encore trop tôt pour le faire. Signe des temps, on pouvait même assister à plusieurs sessions sur Microsoft et l'Open Source ! À quand des comparatifs ou la présence de MySQL ?

Innovation, cette année, la Tech-Ed accueillait en parallèle la conférence MDC dédiée à la mobilité selon Microsoft. Il fut beaucoup question de Windows Mobile 2003, avec l'annonce de quelques nouvelles fonctions dans les futures versions du système. On retiendra notamment la présence d'un installeur unifié, ou encore le stockage persistant sur les Smartphone et la compatibilité ascendante. .Net Compact Framework 2.0 a été assez longuement abordé. Il offre une compatibilité complète avec le framework .Net au niveau des classes, de l'inter-

opérabilité et des protocoles. Il est compatible avec la version 1. On bénéficiera d'un JIT unifié ou encore d'une interopérabilité avec COM et le support de RCW. Pour les Smartphone, on aura le support de SQL Server CE ou encore du Datagrid. Pour le côté affichage, la v2 supportera le changement de résolution.Il y a eu de nombreux ajouts dans le langage, les protocoles (SOAP 1.2 et IPv6) et la sécurité, avec l'apparition de Kerberos et de NTML.

Il y eut deux vedettes : SQL Server 2005 et Visual Studio 2005 (avec ou sans Team System). Sur SQL Server, il s'agissait de démontrer les nouvelles fonctions et son intégration avec le framework .Net ou le pourquoi du comment du XML. Microsoft joue assez gros sur sa base de données et espère bien séduire les entreprises plus largement qu'aujourd'hui. Cependant, SQL Server 2005 n'a cessé d'accumuler les retards et IBM se prépare à lancer un gros pavé avec la nouvelle version de DB2. Le cycle de vie dans le développement et l'ensemble du projet dans Visual Studio a été lui aussi largement évoqué autour de plusieurs sessions sur Team System. Microsoft proposera donc l'ensemble des outils et fonctions qui manquaient cruellement à son IDE phare. Cela évite de s'appuyer sur d'autres éditeurs, comme Rational racheté par IBM.

Rendez-vous est déjà pris pour l'édition 2005 qui se tiendra de nouveau à Amsterdam du 5 au 8 juillet prochain. À vos agendas. Cette prochaine Tech-Ed risque d'être incontournable avec la cascade d'outils et de technologies prévues pour l'année prochaine. Mention spéciale au sac de cette année, spacieux, volumineux et tout orange !

■ **François Tonic**

Offensive sur le marché des développeurs

Les versions Express qui vont être disponibles sont : VB 2005, Visual C# 2005, Visual C++ 2005, Visual J# 2005, Visual Web Dev 2005, SQL Server 2005. La nouvelle déclinaison de SQL Server sera disponible gratuitement. Pour les autres déclinaisons, elles seront sans doute payantes, mais leur prix ne sera dévoilé que l'année prochaine. Actuellement en version bêta, les versions Express sortiront avec les versions " pro ". Le Visual Web Dev est dédié à la création de sites web dynamiques et de Web Services. Il se focalise sur ASP.NET 2. ASP.NET 1.1 n'est pas supporté. Il incorpore son propre serveur Web. Il s'agit avant tout de versions légères, les fonctions avancées ont été enlevées, un peu comme Borland le fait dans les versions Personal. Cependant, on peut développer sans problème. L'objectif premier est d'offrir, à un prix très réduit, des outils pour mieux appréhender le développement .NET. Dans la gamme Express, le plus impressionnant est sans doute SQL Server 2005 Express. Cette version sera donc gratuite et librement redistribuable. On y retrouve le même moteur que la déclinaison complète : T-SQL, procédure stockée, CLR, triggers, XML, Xquery, Express Manager (non inclus dans la première bêta, pour les tâches communes). Pour les limitations, cette édition se borne à un processeur, 1 Go de ram adressage, 4 Go de base de données.

Si vous étiez leader mondial*, que feriez-vous pour le rester?

HASP® HL

RÉINVENTE LA PROTECTION DES LOGICIELS ET LA GESTION DES LICENCES.

Lancer HASP HL, le système matériel de protection des logiciels de nouvelle génération permettant de protéger à la fois chiffre d'affaires et propriété intellectuelle.

Implémenter une sécurité avancée et optimale fournie par le leader incontesté, Aladdin. Protégez-les une seule fois. Protégez-les bien.

Faites-vous une opinion avec "Software Protection : 1-2-3", la démo disponible en ligne ou demandez un kit de développement GRATUIT sur le site HASPHL.com/encore.

- Solution puissante de protection de logiciels axée sur des algorithmes AES et RSA
- Modèles de gestion des licences novateurs utilisés indépendamment de la protection
- Gestion des licences simples et multiples
- Outils et intégration d'API intuitifs et faciles d'emploi
- Clé de haute fiabilité petite, compacte adaptée à toute plateforme

***Aladdin est le n°1 des clés d'authentification sur le marché de la gestion de licences logicielles en 2002 et 2003.**

– Bulletin d'IDC #31 432 de 2004 –

Tél: +33 1 41 37 70 30
Email: HASP.fr@eAladdin.com

Aladdin
SECURING THE GLOBAL VILLAGE
eAladdin.com

ToutJavaScript.com : le site français de référence sur Javascript

Fondée en 1999, cette communauté dédiée au langage Javascript se distingue par la qualité de son contenu original et par le professionnalisme de ses membres.

Rencontre avec Olivier Hondermarck, son créateur

Programmez : Peux-tu nous présenter brièvement Tout JavaScript.com ?

Tout JavaScript.com est devenu, avec le temps, une vraie communauté de développeurs web, amateurs et professionnels. Le site est bien connu dans le monde des webmasters amateurs, dans les universités et également en entreprise.

Qu'est-ce qui t'a poussé à créer cette communauté ?

J'ai lancé le site en 1999, à la fois pour me former au développement web et pour proposer une aide pédagogique, de qualité et en français. Après 5 ans, l'objectif est un peu différent. J'essaye de montrer le potentiel du JavaScript dans un développement web moderne. Le JavaScript est considéré, à tort, comme un sous-langage par beaucoup de développeurs. Pourtant, utilisé judicieusement, il peut rendre un site bien plus ergonomique. En fait, le JavaScript reste indispensable dans tous les projets web, en complément des langages serveurs.

Que penses-tu apporter aux membres ?

Le site permet de trouver une réponse précise sur une utilisation d'une fonction grâce à la référence javascript, d'apprendre le langage grâce aux tutoriaux, de trouver des bouts de code grâce aux scripts et naturellement, de demander des conseils spécifiques sur les forums. Le service rendu est vraiment utile, même aux plus expérimentés. En semaine, les connexions des entreprises doublent l'audience du site.

Quel est le point fort de Tout JavaScript.com ?

A mon avis, c'est le contenu riche, complet, et surtout original : il est développé spécifiquement pour le site et on ne le trouve pas ailleurs. Il n'y a pas d'intérêt à dupliquer des milliers de scripts recopiés chez les concurrents français ou américains. Depuis un an, le principe de communauté a été vraiment développé, avec la mise en place d'un espace membre. La convivialité, la qualité des discussions et l'efficacité des forums participent aussi au succès.

Combien êtes-vous à faire vivre Tout JavaScript.com ?

De nombreux contributeurs participent à l'enrichissement du contenu, au développement des nouvelles fonctionnalités et à l'organisation d'animations ludiques (concours basés sur de la programmation JavaScript). Les forums sont très actifs et de nombreux habitués répondent aux questions des visiteurs. Au niveau gestion du site, je m'occupe du financement et de l'administration du serveur dédié.

Des projets ?

Oui, plein, même si je manque de temps pour les mettre en œuvre ! Mon principal projet est de réaliser un outil de développement collaboratif de modules JavaScript réutilisables. Une amélioration de l'organisation des menus et des rubriques serait également utile pour mieux présenter la diversité et la richesse du contenu.

■ *Propos recueillis par* **David Thévenon**

Le site de référence pour mes développements web

J'ai découvert ToutJavaScript.com il y a deux ans, au cours d'un stage en entreprise. Afin d'améliorer l'ergonomie d'une application Intranet, j'ai dû me mettre au JavaScript ; les "tutoriels" et scripts de ce site m'ont alors beaucoup servi. Depuis, j'y retourne régulièrement, car le contenu est sans cesse enrichi et amélioré. Si bien qu'aujourd'hui, je fais partie de la communauté de développeurs née de ce site : toujours prête à venir en aide sur les forums, et partager ses connaissances.

C'est devenu un site de référence pour mes développements web !

■ **Maxime Pacary** - *Pseudo* : Frosty; *Fonction :* Ingénieur développement - *Entreprise :* WinSoft SA (activité : développement de logiciels)

Fiche communauté

Adresse : www.toutjavascript.com

Date de création : mai 1999

Thème : Développement web, et en particulier le JavaScript

Technos : PHP / MySQL / Javascript

Cible : Amateur et Professionnel

Niveau : Tout niveau, du débutant au professionnel

Ressources : Tutoriaux, scripts, références du langage, forums

Nombre de membres : 2 700

Nombre de forums : 50 000 messages

Visiteurs : 70 000 visiteurs uniques / mois

Nombre de contributeurs actifs : 10

Nombre d'inscrits à la Newsletter : 6 000

Voyage en Inde :

Cessons de nous alarmer au sujet de l'externalisation !

Par Adam Kolawa

En Europe et aux USA, nous nous alarmons de la menace de délocalisation, en particulier de l'externalisation de projets IT vers l'Inde. Cependant, après avoir été voir moi-même la situation en Inde, je suis persuadé que nous pouvons nous détendre un peu. L'Inde possède un certain nombre de sociétés de développement logiciel compétentes, et qui peuvent prendre en charge divers travaux à un coût raisonnable, mais elles ne peuvent assurer qu'une petite partie du développement dont nous avons besoin.

Le travail manuel est tellement bon marché que peu de machines sont utilisées. Les Indiens ne comprennent pas que certains travaux ne peuvent être exécutés à mains nues.

Je suis récemment parti en Inde pour discuter de qualité avec des sociétés de développement offshore. A Bangalore, Chennai, et Hyderabad, j'ai rencontré et échangé avec des responsables et développeurs de structures de développement indiennes. J'ai aussi pu observer l'externalisation en action.

On exagère la menace de l'externalisation IT

Il y a approximativement 100 000 développeurs de logiciel à Bangalore, la version indienne de la Silicon Valley américaine. Comparés aux 3 à 4 millions de postes de développeurs que l'on estime avoir aux Etats-Unis, ces 100 000 emplois sont une goutte d'eau dans l'océan. Considérez que de grandes sociétés comme IBM emploient à elles seules, près de 50 000 développeurs, rien qu'aux USA.

L'Inde ne prend actuellement en charge qu'un petit pourcentage du développement logiciel mondial. Elle ne semble pas posséder l'infrastructure nécessaire pour supporter une expansion rapide du développement offshore, et en mettre une en place risque de prendre des années. Avec le temps, le marché du développement indien pourrait grossir pour atteindre 10 à 30% du marché américain, mais l'infrastructure existante ne pourrait pas supporter le volume de développement que cela impliquerait. Ainsi, les gens ont des difficultés à se rendre sur leur lieu de travail, à cause des carences du réseau de transports – il y notamment des vaches qui provoquent des encombrements dans les rues, même devant les bâtiments de grandes entreprises de l'offshore. Une fois au travail, les employés sont confrontés à de fréquentes coupures d'électricité... Bien sûr, certaines choses peuvent être améliorées, mais, sachant que la plupart des travaux de construction et de voirie semblent être réalisés sans l'assistance de machines ou de l'automatisation, il apparaît peu probable que l'infrastructure du pays puisse être modernisée assez rapidement pour soutenir une hausse soudaine de la demande en externalisation IT.

Et même si c'était le cas, il semble déjà y avoir une pénurie de main-d'œuvre, en terme de profils adéquats. La preuve en est la facilité avec laquelle presque toute personne un tant soit peu formée aux technologies (ou avide d'apprendre) peut obtenir un emploi de développeur décent.

De plus, des différences culturelles marquées et un niveau de vie très éloigné rendent peu probable un déplacement, par exemple, de managers américains vers l'Inde pour prendre les places disponibles là-bas.
Un autre facteur qui pourrait mettre une limite à l'expansion est le fossé important et qui s'approfondit de façon dramatique entre les nantis et la couche pauvre de la société. En effet, seule une petite partie de la société indienne profite d'une croissance remarquable de leur richesse et de leur niveau de vie. Je m'attends à beaucoup de tensions sociales, voire d'agitation si ce processus se poursuit.

En Inde, il y a peu de développeurs experts, qui puissent créer et concevoir des logiciels vraiment innovants. La majorité des développeurs indiens semblent avoir une bonne compréhension des concepts du développement et

Le siège de Wipro, une société de services.

de leur application dans un projet typique, mais peu sont assez expérimentés et inventifs pour pouvoir apporter des contributions notables à leur domaine ou des solutions ingénieuses à des problèmes complexes.

Les responsables indiens ne reconnaissent pas la valeur de l'automatisation et des outils de développement

Les managers à la tête du développement en Inde sont des adeptes du coût du travail peu élevé, mais cela les empêche de se rendre compte que certaines tâches ne peuvent être exécutées de façon efficace si elles le sont manuellement. Ils sont convaincus qu'un grand nombre de développeurs, avec des salaires relativement bas, a le même impact que des outils de développement logiciel. C'est pourquoi, ils n'ont souvent pas de budget ou n'investissent pas dans une infrastructure qui permettrait d'automatiser et de rationaliser les tâches de développement critiques. Ils ne réalisent pas que, quel que soit le nombre de développeurs ou testeurs ajoutés au projet, pour vérifier les programmes critiques " à la main ", ils ne pourront pas le faire de façon aussi approfondie, homogène et efficace que le ferait un outil automatique. Pourtant, sans l'automatisation de pratiques, comme la vérification des standards de codage, la création de cas de tests, les tests de non-régression et les tests de performance...il est littéralement impossible de développer un logiciel qui fonctionne correctement, ne " plante " pas, ne se comporte pas de façon imprévisible dans des circonstances exceptionnelles, soit assez performant et résiste aux menaces en terme de sécurité.

Cette perception de l'automatisation comme un luxe, plutôt qu'une nécessité, est partagée partout à travers l'Inde. Par exemple, la plupart des travaux de construction que j'ai observés sont réalisés de façon totalement manuelle. Des groupes d'hommes travaillaient sans qu'il y ait ni grue ni machine en vue. J'ai notamment aperçu plusieurs bâtiments en construction et qui n'étaient visiblement pas droits.

En Inde, la productivité d'un développeur est basse

La productivité des développeurs indiens est extrêmement basse parce qu'ils n'ont pas les outils nécessaires à la rationalisation de leur travail. Il n'est donc pas rare de voir des sociétés de développement indiennes engager jusqu'à 500 développeurs pour chaque projet. Aux Etats-Unis, des projets comparables seraient souvent pris en charge par une équipe de seulement cinq à dix développeurs experts, disposant d'une bonne infrastructure d'outils automatiques.

De nombreuses sociétés de développement indiennes mettent en avant de hauts niveaux SEI-CMM ou SEI-CMMI, y compris le niveau 5 de certification, que tant d'entreprises rêvent seulement d'atteindre. Toutes ces sociétés certifiées doivent avoir un processus de développement bien rôdé et officialisé noir sur blanc. Mais en pratique, nombreuses sont celles qui n'ont pas les outils de développement nécessaires pour implémenter ces pratiques ou qui ont simplement un nombre limité de licences pour ces outils. Elles admettent utiliser ces outils de façon sporadique, ce qui ressort du comportement d'une société au niveau 1 mais pas au niveau 5 des CMM.

Conclusion : une demande mondiale croissante.

Il est important de comprendre que, même si la proportion de développement logiciel réalisé en Inde augmente, la quantité de code qui devra être produite partout dans le monde va augmenter également. De nos jours, on a besoin de logiciels, non seulement pour faire fonctionner des ordinateurs de bureaux, mais aussi pour améliorer et intégrer des systèmes d'entreprises, contrôler des dispositifs d'alimentation ou des systèmes de télécommunication, faire fonctionner les automobiles, lire des DVD ou des VCR etc. etc.. Avec la demande de logiciel en hausse, des développeurs seront recherchés partout, même si certains de ces postes ou des postes existants aux Etats-Unis sont déplacés en Inde, il y aura toujours un besoin de main d'œuvre ailleurs pour satisfaire cette demande.

Enfin, alors que l'Inde bénéficie ainsi de l'explosion de son industrie du logiciel, les consommateurs indiens demandent de plus en plus de biens high-tech. Certains estiment le montant des importations de l'Inde en produits high-tech depuis les USA à trois milliards de dollars pour 2004 seulement. Elles nourrissent de cette façon le besoin en software qui augmente et qui créera des emplois pour les développeurs un peu partout !

Adam Kolawa

Président et cofondateur de Parasoft. Docteur en Sciences Physiques, il est co-auteur de Bullet proofing Web Applications (Hungry Minds 2001). ak@parasoft.com