LE MAGAZINE DU DÉVELOPPEMENT

Programmez !

Mensuel • Octobre 2004 • N°68 • 5,95 €

SGBD
Les bases de données en pleine mutation

LINUX
Quelle distribution choisir ?

Carrières :
LES PROFILS LES PLUS DEMANDÉS

SÉCURITÉ
Blinder le code

BASE DE DONNÉES
Procédures stockées avec Yukon

PRATIQUE
Des PDF avec Visual C# 2005 Express

Maîtriser le shell de Windows

TECHNOLOGIE
Java en Cluster

PROJET
La méthode Macao

Printed in France - Imprimé en France
BELGIQUE 6,45 € - SUISSE 12 FS
LUXEMBOURG 6,45 € - Canada 8,95 $ CAN

M 04319 - 68 - F: 5,95 €

Choisir sa distribution Linux

Passer à GNU Linux n'est pas toujours chose facile, surtout quand on est un utilisateur assidu de Windows ou MacOSX. Comment choisir la bonne distribution pour une utilisation bureautique ou de développement ? La question n'a rien d'anodin dans la diversité des distributions actuellement sur le marché. Ici, nous n'évoquerons pas l'aspect serveur, mais uniquement le poste de travail.

Impossible en quelques pages d'évoquer toutes les distributions. Nous avons voulu nous concentrer sur les principales à savoir : Mandrakelinux avec les Discovery et PowerPack, SuSe avec SuSe Professional 9.1, Fedora Core 2 et Red Hat Enterprise Linux WS (version standard). Nos tests ont été effectués sur un portable Dell Inspiron 5100.

Comment comparer des distributions GNU Linux similaires dans les applications et les fonctions ? Voici les questions que l'on peut se poser :

- Quelle plate-forme matérielle vais-je utiliser ?
- Quelles utilisations de Linux ?
- De quelles applications ai-je besoin ?
- Suis-je totalement novice ?
- Je cherche une installation "fixe" ou "live" ?
- Distribution payante ou téléchargeable ? Et est-ce que je souhaite du support ?

Ces questions permettent de dresser un premier portrait de " ma distribution idéale ". Selon l'utilisation que l'on veut en faire, la plate-forme dont on dispose, et le processeur - Intel ou PowerPC, 32 ou 64 bits -,on n'optera pas pour la même distribution, ou tout du moins pour la même déclinaison d'un éditeur. Que l'on soit sur un desktop, un serveur, un portable, la distribution changera, ainsi que les outils / applications nécessaires.

Quelles applications

Pour un utilisateur sortant de Windows ou de MacOS X, qui souhaite utiliser aussi ou uniquement Linux, le mieux est sans doute de partir sur des distributions connues et de laisser, en tout cas dans un premier temps, les distributions plus pointues ou confidentielles. Le support et l'assistance utilisateur peuvent être un argument de choix (même si dans ce cas, l'assistance téléphonique est souvent payante). Si vous voulez tester sans installer physiquement, les distributions Linux ou sur clés USB constituent un choix intéressant : toucher sans risque. C'est une bonne solution pour apprendre Linux et les applications, et tester la compatibilité de vos documents.

Si actuellement, une large panoplie de périphériques est supportée par défaut dans les distributions (officiellement ou non), pour des matériels spécifiques, très récents ou très anciens, vérifiez avant toute installation la compatibilité Linux. Tous les constructeurs ne proposent pas les pilotes nécessaires.

N'oubliez pas non plus que certains outils (notamment) de développements Web (ou non) n'existent pas en version Linux (ex. : Macromedia Dreamweaver).

Différentes applications sont livrées en standard. Certaines distributions proposent des logiciels commerciaux en plus des outils libres. Même si des logiciels libres manquent, ce n'est pas un gros problème, il sera possible de les télécharger ultérieurement.

Les solutions Live et à clé

Si vous ne voulez pas installer un Linux, vous pouvez utiliser les solutions Live ou sur clé USB. La clé USB paraît être une solution plus souple et pratique pour utiliser Linux en Live tout en laissant libre le lecteur CD/DVD. Cependant, ces solutions sont très peu nombreuses contrairement aux CD Live, comme Knoppix ou Mandrake Move et ne comportent que le minimum vital au niveau application (mais en général, on dispose tout de même d'Open Office, Web etc.). Les performances sont tout de même moindres que sur une véritable installation. Toujours chez Mandrake l'éditeur propose depuis peu une solution intéressante, un Linux installé sur un disque dur USB que l'on emmène avec soi.

Tarification et support

Les ténors du monde GNU Linux proposent des distributions complètes et packagées vendues en boîte, ou parfois téléchargeables. Outre le

Support et assistance

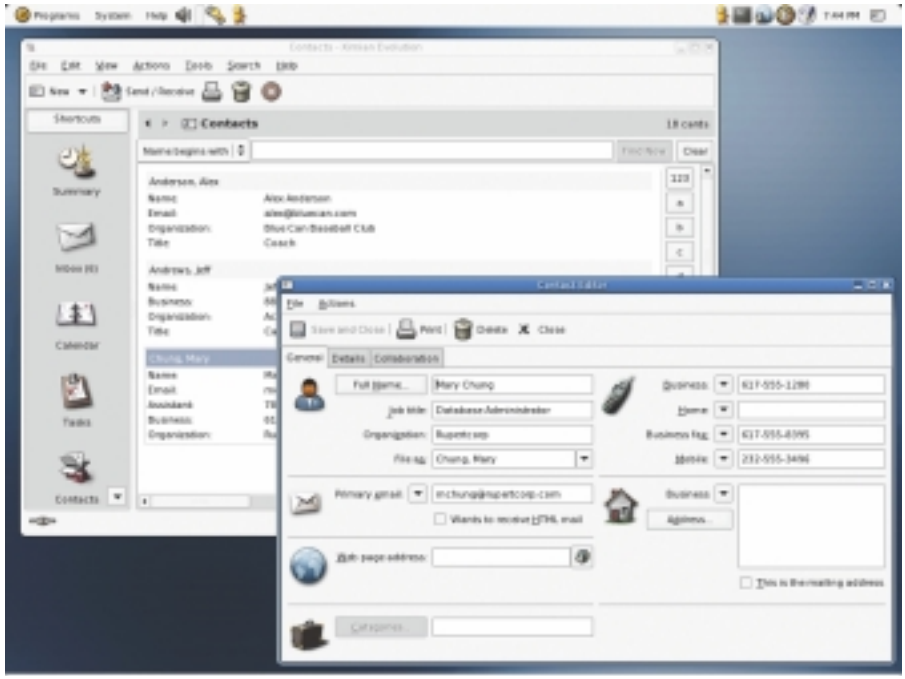
	Mandrakelinux Discovery	Mandrakelinux PowerPack	SuSe Professional 9.1	Red Hat Enterprise Linux WS Standard v3	Federa Core 2	Sun Java Desktop System
Support Web / durée	Oui / 30 jours	Oui / 60 jours	Oui / 90 jours (sur installation)	Oui 24/7 / 1 an	Non	Oui / 60 (installation – configuration)
Support téléphonique	Option	Option	Oui / 90 jours (sur installation, conditions requises)	Oui jours ouvrés / 1 an	Non	Oui / 60 (installation – configuration)
Mise à jour	1 mois gratuit offert à Mandrakeonline et Mandrakeclub	1 mois gratuit offert à Mandrakeonline et Mandrakeclub	Oui	1 an à Red Hat Network, MAJ trimestrielle	N/A	1 an
Support payant complémentaire	Oui	Oui	Oui	Oui	Non	Oui
Support en Français Web / Téléphone	Oui / Oui	Oui / Oui	Oui / Oui	Oui / Oui	N/A	Oui / Oui
Prix distribution (*)	44,90 €	79,90 €	59,95 €	313,39 €	Gratuite	100 $ US (hors promotion) pour 1 an et 1 utilisateur

(*) Attention : tarifs officiels (des éditeurs) annoncées ou pratiqués par leur boutique en ligne

prix, qui peut être un facteur de choix, il ne faut pas oublier le support et l'assistance utilisateur. Ce point peut être vital en entreprise, voire chez les utilisateurs qui maîtrisent peu l'univers du libre et les notions de recompilation du noyau, code source, etc. Le support n'est pas homogène selon les éditeurs. Un grand nombre de distributions téléchargeables ou proposées dans des livres ou magazines n'offre aucun support / assistance. Par contre, quand on achète auprès d'un éditeur, en principe, on dispose d'un support limité dans le temps et les fonctions systèmes que l'on peut par la suite étendre.

Comme vous pouvez le voir, il existe des disparités importantes. Red Hat offre un support utilisateur sans doute le plus complet (le prix d'achat est en rapport), tout comme Sun. Pour le reste, on doit se contenter du minimum. En général, un délai est nécessaire entre la question et la réponse (mais pas toujours) et le temps de chaque appel peut être limité à

quelques minutes. Pour des supports plus spécifiques ou pour étendre l'assistance, il faudra payer. Si vous décidez tout de même d'installer une distribution gratuite, vous le ferez en connaissance de cause et sans support ! Ainsi si Fedora est un projet issu de Red Hat et utilisé par Red Hat pour ses propres systèmes, cette distribution ne propose aucun support. Il est certain aussi, que les distributions Sun et Red Hat se destinent avant tout aux entreprises. Quelquefois, l'éditeur proposera des



formations (payantes) courtes pour apprendre GNU Linux. Un point important à regarder : vérifier ce que couvre exactement le support standard. Pour la SuSe, toutes les formes d'installations ne sont pas prises en compte par le support offert ! N'oubliez jamais de sauvegarder vos données avant toute installation.

Quelle distribution pour quelle utilisation ?

Si globalement, les distributions sont homogènes dans les fonctions et outils, certaines conviennent mieux à certains usages. Pour le développeur, hormis la Discovery, on peut envisager l'ensemble des distributions, ensuite c'est un souci de stabilité, performance sur son ou ses postes de travail. Durant nos essais, la SuSe, Red Hat, Fedora ou encore Sun Java Desktop System, nous ont paru stables en utilisation, alors que pour les deux versions de Mandrake, des soucis de démarrage (voire d'installation) ont été rencontrés (cependant rien de trop gênant). Pour la bureautique et un usage au quotidien, Mandrake, Fedora et SuSe constituent un bon trio. Là, on retrouve grosso modo les mêmes applications. Open Office est partout. La SuSe est peut-être un peu mieux lotie avec notamment TextMaker ou Scribus. Sun se démarque en mettant son StarOffice. La compatibilité entre un OpenOffice et Microsoft Office est souvent bonne, à condition de ne pas utiliser des fonctions trop spécifiques ou des formats spéciaux.

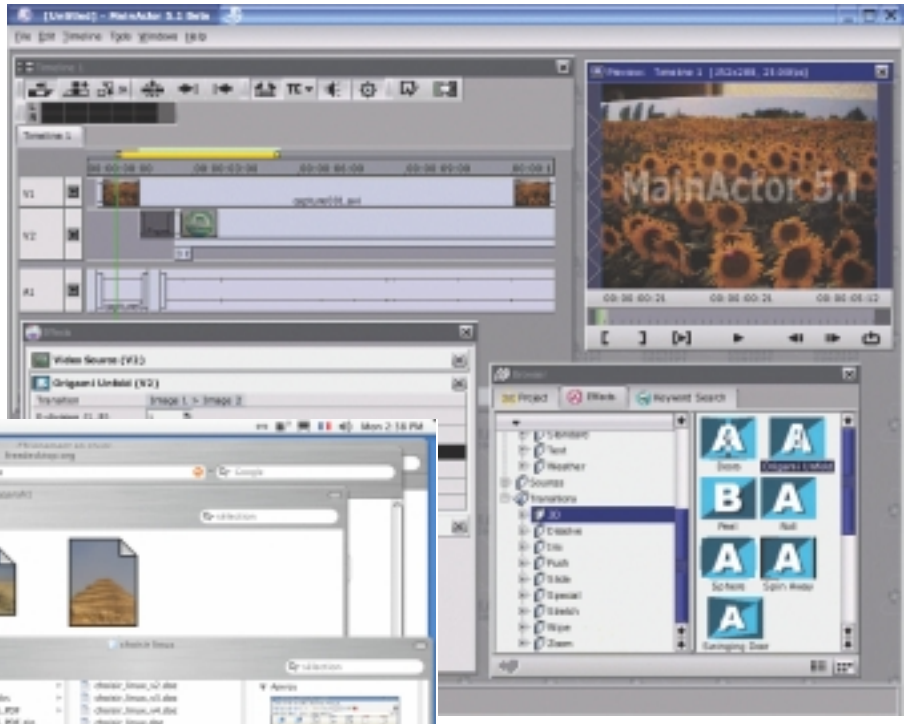
Côté développement, les outils de base sont en général partout identiques (même s'il y a une différence entre Gnome et KDE). Java n'est pas toujours disponible en standard sur les distributions. Certains ajoutent des éléments spécifiques : Java Studio Creator et NetBeans pour Sun, compilateurs Intel C++ / Revolution / Open Motif / pour la PowerPack, des couches Mono dans la SuSe... À noter que le C/C++ constitue la référence dans les distributions (QT Designer est souvent présent) alors que Java se limite à quelques distributions. À vous ensuite de rajouter les éléments dont vous avez besoin. Si vous voulez faire des sites Web, malheureusement, Linux n'a pas encore d'outils comparables à GoLive ou Dreamweaver. Si vous devez utiliser du 64 bits ou migrer sur ce type de processeur, les principales distributions possèdent ce type de versions. SuSe Professional 9.1 inclut la version 64 en standard. Vous pouvez aussi vérifier si la distribution est certifiée LSB (Linux Standard Base). Cela peut être un gage de qualité.

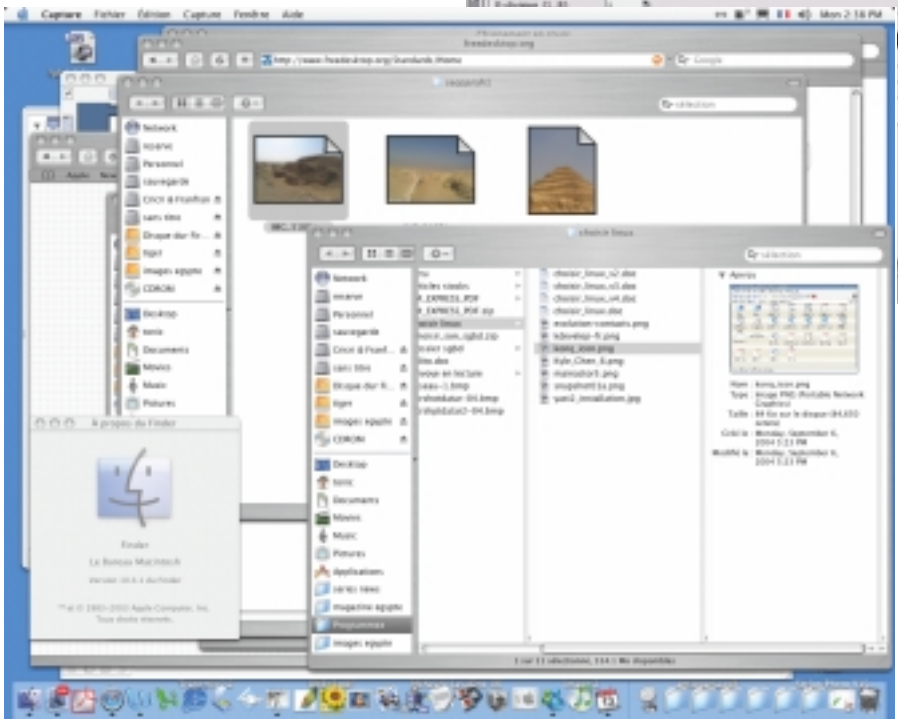
Quelle utilisation

	Poste personnel	Portable	Poste de travail	Développement	Serveur
Mandrake Discovery	*				
Mandrake PowerPack	*	*	*	*	*
SuSe Profesional 9.1	*	*	*	*	*
Red Hat Enterprise Linux WS			*	*	
Fedora	*	*	*	*	*
Sun Java Desktop System		*	*	*	*

Installation

Vous pourrez souvent opter pour le tout ligne de commande, ou pour l'interface. Si on n'atteint pas encore la simplicité d'un installeur MacOS X, Linux a fait beaucoup de progrès. Si la Mandrake fut pionnière en la matière, aujourd'hui, Suse et Fedora – Red Hat ou Sun se révèlent aussi même plus ergonomiques que les distributions Mandrake. L'installeur YaST présent dans la SuSe est efficace et masque bien les choses, tout en proposant plusieurs types d'installations selon l'usage.





L'installeur Anaconda de Red Hat / Fedora constitue une vraie bonne surprise : simple, ergonomique avec une bonne reconnaissance du matériel. La création de partition se fait souvent sans problème.

Développement

	Mandrakelinux PowerPack	SuSe Professional 9.1	Red Hat Enterprise Linux WS Standard v3	Federa Core 2	Sun Java Desktop System
Principaux outils	CH Standard Edition, Intel C++ Compiler, Open Motif Everywhere, Revolution, GCC, Kdevelop, GDB, J2SDK 1.4, J2RE, Perl, Python, QT, ODBC, outils SGBD, Glade, Postgresql, Kaffe…	Kdevelop, Umbrello, GCC, GDB, Mono, J2SDK 1.4, J2RE, jikes, jmk, QT, Eclipse, Anjuta, iAnywhere	Kdevelop, GCC, Qt, SGBD…	GCC, GDB, junit, Kdevelop, Qt, ODBC, outils SGBD	J2SE, Java Studio Standard 5, NetBeans,
Principaux langages	Perl, Python, C, C++, Java	Perl, Python, C, C++, Java, C# / .NET, Tcl	Java, perl, python, C, C++	Java, perl, python, C, C++	Java

Documentation

C'est parfois le point faible des distributions. Mandrake fait un effort dans ce domaine, même si pour la version Discovery, on doit se contenter d'un manuel assez basique. La PowerPack est plus intéressante, avec l'ajout d'un manuel de référence sur Unix et le système. La Suse bénéficie d'un excellent manuel, mais malheureusement uniquement en Anglais. La version francisée n'est pas encore disponible, mais elle est prévue (dixit l'éditeur). La Fedora ne propose rien. Il faut se contenter de l'aide disponible en ligne.

Sur l'aide disponible sur le système, la qualité est là aussi aléatoire. Mandrake et Red Hat proposent un bon contenu sur l'ensemble des applications et du système, en Français. Sous Fedora et SuSe, on navigue souvent entre Français et Anglais. Pour réellement aller plus loin dans la documentation, vous devrez aller sur le Web, ou alors acquérir les ouvrages de référence Linux. Mauvaise surprise, il apparaît que la documentation n'est pas toujours installée par défaut (ex. : dans la Red Hat, il faut installer le CD spécifique).

MacOS X et Linux sur Mac

N'oublions pas de citer MacOS X et Darwin, comme systèmes Unix performants et simples d'accès. Apple a beaucoup fait pour rendre Unix transparent aux utilisateurs et c'est plutôt une réussite, d'autant que de plus en plus d'outils fonctionnant sous Linux sont portés sous OS X. Les couches basses d'OS X, Darwin, sont disponibles librement et supportent la plate-forme PC ! De plus en plus d'outils libres sont portés sous MacOS X et grâce à Fink et ou X11 d'Apple, on accède à la plupart des logiciels du monde Linux / Unix. Il existe aussi des versions PowerPC des principales distributions. Cependant, la plus connue est la Yellow Dog.

Sun Java Desktop System

Nous n'avons pu tester la version 2 du Linux de Sun. Cette distribution est payante et se destine avant tout aux entreprises (serveur et poste de travail), même si pour une utilisation personnelle, ce Linux est impeccable. Le Sun Java Desktop System s'appuie sur la SuSe. Cette distribution est agrémentée d'applications conçues par Sun. On bénéficie ainsi de StarOffice (la version commerciale d'Open Office), de Java. Dans la version 2, Sun a ajouté des bonus pour développeurs : Sun Java Studio, NetBeans.

Les ⊕ **Basé sur SuSe, outils en standard (développement et bureautique), stabilité, support**
Les ⊖ **Plutôt réservé aux entreprises pour le moment, tarification annuelle**

Développement

La disparité est assez grande entre les distributions. SuSe propose une gamme assez complète avec, comme différence, l'implémentation de Mono et du SGBD iAnywhere. La palette d'outils est aussi vaste que celle offerte par la Mandrake PowerPack, même si dans cette dernière, on bénéficie des compilateurs C++ d'Intel et de plusieurs outils commerciaux : Revolution, Open Motif. La seule distribution réellement faite en standard pour le poste de développement est la Java Desktop System orienté Java, avec la présence de Java Studio Standard, NetBeans et sans doute rapidement de Java Studio Creator. On retrouve, hormis les outils spécifiques, toujours ou presque les mêmes outils de développement pour Gnome, KDE, C/C++, Python. En plus des outils livrés en standard, certains éditeurs proposent des packages spéciaux. Novell a concu le Novell Linux SDK incluant des librairies, IDE en tout genre et pour langage et outils tiers. Red Hat propose lui aussi son kit, la Red Hat Developer Suite (Java, C/C++...), basé sur Eclipse. Sur Fedora, on trouvera les outils standard sans plus. À vous de télécharger et d'installer.

Noyau Linux

Si fonctionnellement, les distributions se ressemblent, la version du noyau les distingue souvent. Mandrake est une des premières, ou la première " grande " distribution à implémenter les nouveaux noyaux. Cela peut affecter la stabilité du système et des applications. Si vous n'êtes pas habitués à recompiler le noyau, préférez des versions du kernel reconnues pour leur stabilité. Red Hat ou SuSe sont souvent plus lents à intégrer, mais d'un autre côté, ils jouent la carte de la stabilité et de l'utilisation au quotidien.

L'incompatibilité avec des périphériques, ou l'impossibilité d'installer la distribution peut provenir du noyau. Pour ceux qui utilisent Windows ou MacOS X, ils savent que le noyau n'est pas un élément que l'on regarde et que l'on met à jour soi-même. Sous GNU Linux, il s'agit d'un élément central. Deux versions sont très stables : la 2.4.26 et la 2.6.7. De plus, dans les distributions, il n'y a pas d'homogénéité dans les versions de noyau, chaque éditeur a sa politique. Dans le même esprit, les projets libres avancent plus vite que les distributions. Ainsi, les versions d'applications intégrées dans les distributions sont souvent en retard par rapport aux versions sorties par les développeurs. Ainsi dans la SuSe 9.1, Gnome est en v2.4 alors que la 2.6 est disponible. Cette différence s'explique par le processus d'intégration, et de production par les éditeurs des nouvelles versions, et aussi pour une question de stabilité du système.

■ **François Tonic**

SuSe : la SuSe est une excellente distribution. Mieux finie par certains aspects que la Mandrake, avec des outils de configuration maison, bien conçus. Pour un utilisateur nomade, la SuSe offre une bonne stabilité, meilleure que d'autres distributions. Elle saura être polyvalente au quotidien, à la maison ou sur son poste de travail.

Les ⊕ **Qualité du packaging, le support, stabilité sur portable, 64 bits en standard, les outils proposés, installation**
Les ⊖ **Guide papier en anglais ainsi qu'une partie de l'aide**

Mandrake Discovery / PowerPack : si la Discovery permet de démarrer en douceur dans l'univers Linux, il est certain que la version PowerPack sera bien plus intéressante pour des utilisateurs plus exigeants souhaitant exploiter Linux sans à avoir à rajouter de nombreux logiciels. Si vous maîtrisez votre système actuel, la PowerPack est le choix idéal.

Les ⊕ **Bonne tenue générale, installation, liste des outils en PowerPack, prix**
Les ⊖ **Support, quelques instabilités sur portable, Discovery assez pauvre**

Fedora Code 2 : finalement, une distribution Linux plutôt homogène et polyvalente pouvant servir à tous les usages ou presque : développement, à la maison, au bureau, en petit serveur, etc. Cependant, le manque de support officiel n'est pas favorable à une utilisation intensive au bureau ou même chez soi si on ne connaît pas suffisamment Linux.

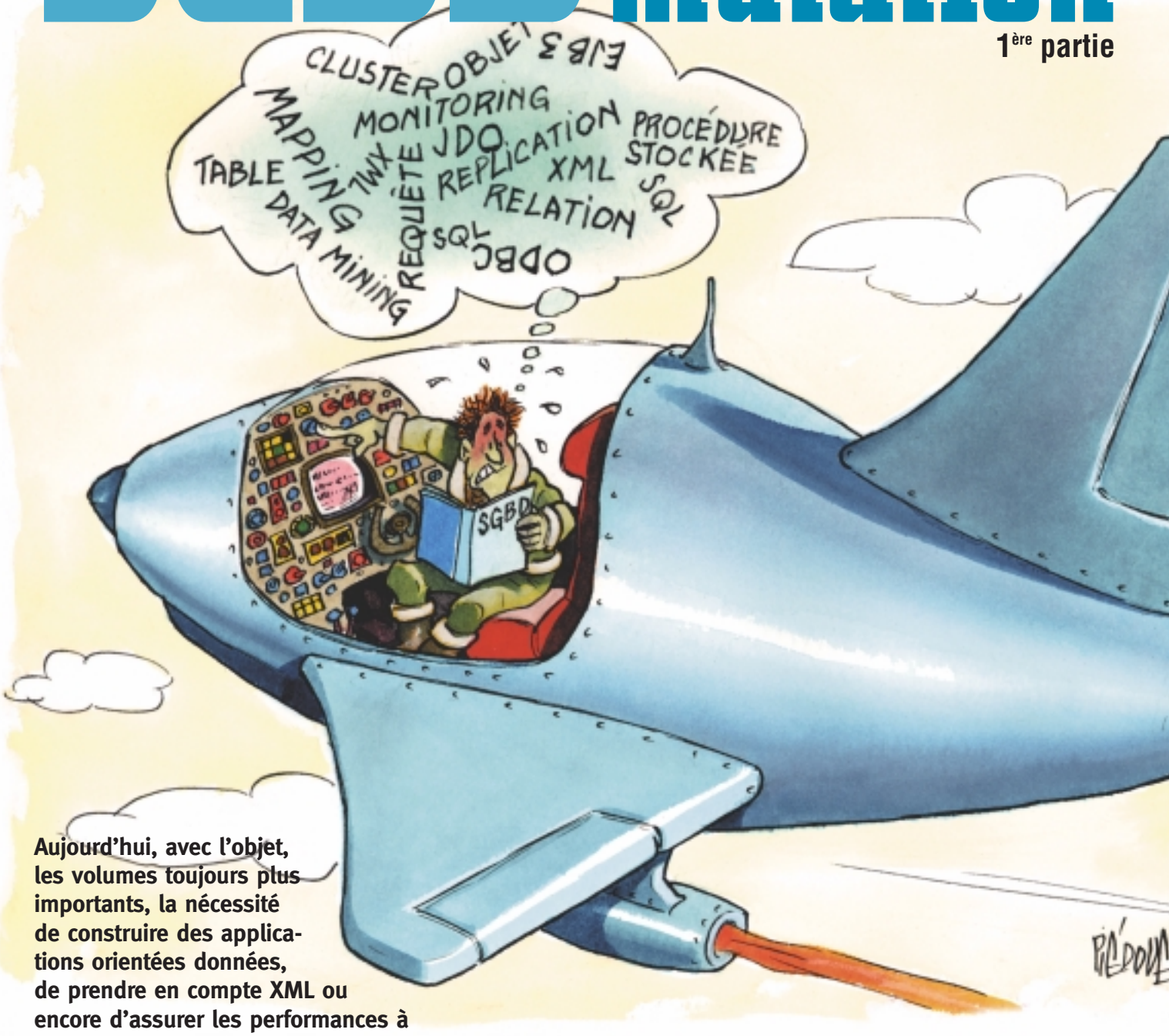
Les ⊕ **Stabilité, installation**
Les ⊖ **Support, pas de CD officiel**

Red Hat : elle est basée sur la Fedora (tout comme l'ensemble de la gamme Red Hat). Bien finie, elle convient parfaitement au poste de travail. Elle est plutôt orientée entreprise par son prix et les prestations proposées. Convient aussi à un usage plus personnel. Cependant si vous utilisez un portable, ce n'est peut-être pas le meilleur choix, à cause du manque d'outils spécifiques.

Les ⊕ **Basée sur Fedora, stabilité – installation, support, richesse fonctionnelle**
Les ⊖ **Support des portables, prix élevé, gamme Linux parfois peu claire**

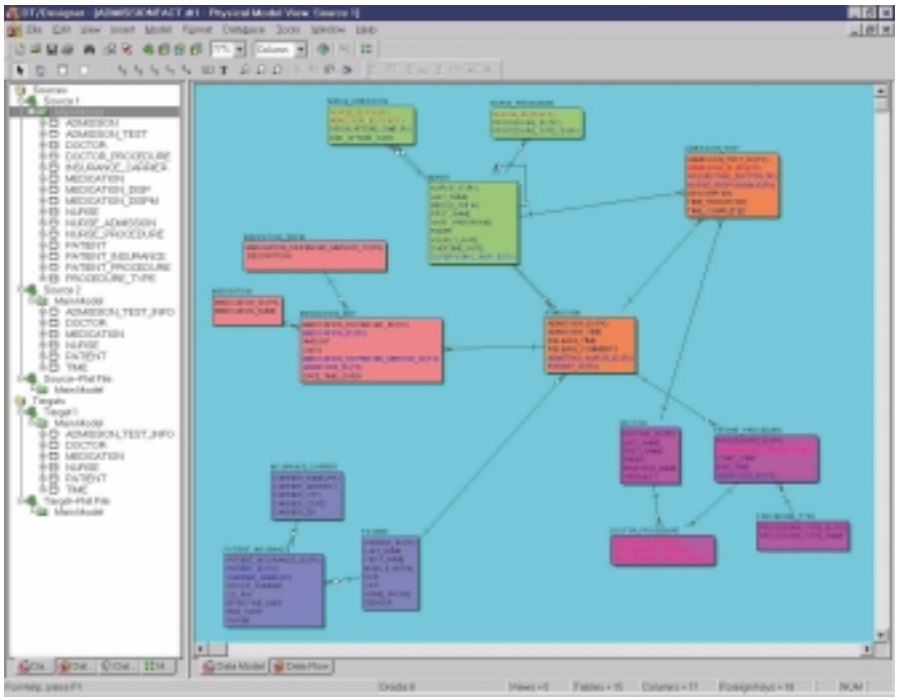
SGBD en pleine mutation

1ère partie



Aujourd'hui, avec l'objet, les volumes toujours plus importants, la nécessité de construire des applications orientées données, de prendre en compte XML ou encore d'assurer les performances à l'application, le SGBD et la donnée tiennent la vedette. Avec des outils de plus en plus complexes et lourds, les développeurs n'ont qu'à bien se tenir pour maîtriser, intégrer, gérer et manipuler bases et données. Dans ce dossier en 2 parties, nous avons voulu aborder plusieurs aspects : le mapping, les évolutions et les problématiques, le choix de son SGBD, les outils accompagnant la base.

Bases et données au cœur de l'application



On l'oublie souvent, mais le SGBD est au cœur de l'application moderne. Les données encore plus. L'explosion d'Internet et les besoins sans cesse croissants de stockage de données ont favorisé cet état de fait. Cela va du simple site, avec une gestion de contenus toute simple, aux énormes bases de données des grandes entreprises ou centres de recherches où là, les volumes dépassent l'imagination et peuvent atteindre plusieurs péta-octets ! Plus on monte dans la complexité et les volumes de stockage, plus la base de données devient capitale. On travaille dans un monde critique où la panne est impossible, où la perte de données est impensable et où la recherche et le traitement de la donnée sont vitaux. Toute proportion gardée, ces problématiques se retrouvent dans les gros sites marchands Web ou les gros projets informatiques. Application et données sont tellement imbriquées que l'on développe en parallèle le code et la base. Une erreur de conception d'une table ou d'une base et il faut tout refaire. Que la communication entre le front office (= l'application) et le back office (= base) écroule les performances et les temps de réponse et il faut alors tout re-tester, re-vérifier, comprendre d'où viennent les ralentissements. Car intégrer la donnée et la base dans une application ne va pas de soi. Bien au contraire.

La construction de la base de données se fait en général avant le code pur. On définit la structure de chaque élément de la base (table, champ, index, relation, taille, type, etc.). Pour les projets conséquents, on peut partir d'un squelette UML (ou d'un modèle non UML) pour ensuite générer l'ensemble de la base. C'est seulement ensuite que l'on peut réellement coder et concevoir interfaces et relations aux données. De plus en plus d'éditeurs proposent de la modélisation (ex. : Embarcadero ER/Studio et Describe).

Les applications mobiles n'échappent pas aux données. Outre qu'on doit utiliser des moteurs de bases de données mobiles (donc plus légers), il faut souvent mettre en place duplication, réplication, mise à jour des données entre l'application mobile et le serveur. Et là, il faut mettre en place les mécanismes d'authentification et de cryptage pour la sécurité, vérifier l'intégration des données et leur cohérence par rapport au modèle de données. Récupérer les données, les traiter, les intégrer à la base, tout cela est transparent à l'utilisateur ou presque, mais pour le développeur et les responsables réseaux, non ! Il faut coder les requêtes en E/S et prévoir les modules de traitement dans le client et le serveur.

Vive l'abstraction

Le terme sonne comme un argument. On peut citer deux grands types d'abstraction. Le premier est l'abstraction technique. Cela permet de découpler totalement l'application de la base. Pour ce faire, on passe obligatoirement par un standard (si possible ouvert ou un standard de fait).

L'API est implémentée dans l'application. Cependant, cette première couche ne peut pas faire grand-chose toute seule, car elle ne sert pas à échanger des données. Dans ce but, il faut passer par une couche métier. C'est cette couche d'abstraction qui va s'occuper des échanges, des requêtes, etc. En réalité, dans une application, il y a trois grandes couches d'abstraction : la présentation, les traitements et les données. L'abstraction permet de réaliser le découplage entre les couches. On peut ainsi, sans trop de peine, changer de couche d'accès aux données ou de SGBD. Cependant, il faudra tout de même adapter le modèle des couches pour pouvoir prendre en compte le changement. La couche " donnée " demeure plus ou moins liée au SGBD pour des raisons de communication et d'échange, même si XML ou les Web Services permettent de limiter ce lien. Mais du code spécifique ou des adaptateurs seront toujours nécessaires à part si l'on dispose d'outils de génération dynamique.

Désormais, le découplage des couches, en quelque sorte l'abstraction, devient la norme de fonctionnement. On évite de lier directement la base au code.

Migration

La migration est très à la mode, beaucoup d'utilisateurs évoquant le passage d'une base propriétaire à une libre, par exemple d'Oracle à MySQL. Pour les bases de taille réduite, ou peu complexes, la migration ne doit pas poser d'importantes difficultés. Cependant, le "couac" n'est jamais impossible. Une migra-

tion ne consiste pas à simplement déplacer les données, mais toute la logique et la structure de la base. Des outils de migration existent et fonctionnent plutôt bien. Le choix est assez vaste. On pourra citer à titre indicatif : SQLPorter de RealSoftStudio (MySQL, Oracle, IBM, Microsoft, Sybase). Plusieurs outils permettent de migrer de SQL Server à MySQL (Microsoft DTS, SQLyog, MSSQL2MYSQL). Pervasive propose aussi des outils d'intégration des données (ex. : Pervasive Data Integrator). Embarcadero, grand spécialiste du domaine, offre une panoplie d'outils autour du SGBD et des données (performances, tuning, administration...). Pour l'intégration pure, on pourra regarder du côté d'InterSystems Ensemble.

Attention : les outils de migration ne supportent pas l'ensemble des fonctions que l'on peut utiliser dans un SGBD. Les fonctions très spécifiques devront être configurées ou réécrites. Ne négligez jamais le coût de migration et sa qualité. Il faut que le nouvel SGBD s'intègre parfaitement à votre environnement. Soyez méthodique : audit, évaluation des SGBD, modélisation de la migration (base et données). Ne mettez jamais en production une base migrée sans tests. Même quand l'on migre d'une version à une autre, la prudence est de mise. Si Oracle a sorti l'an dernier la 10g, de très nombreux utilisateurs demeurent aux 8, voire 7 ! Dans le cas de très grosses applications ayant des bases conséquentes, la migration se fait en général très lentement, car, outre l'aspect stabilité de fonctionnement, le coût financier est en général important.

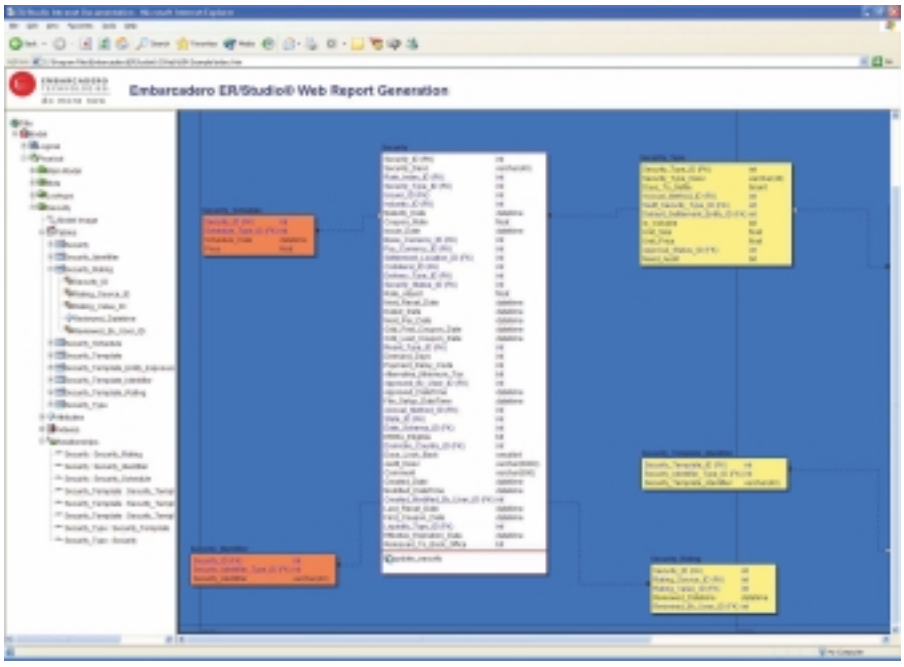
Un des enjeux dans l'avenir concerne les très nombreuses et énormes bases de données stockées sur mainframe. Va-t-on ou non voir une migration vers des solutions micro ? Sans doute y aura-t-il une migration, mais pas d'envergure, car les SGBD mainframe ont su évoluer et implémenter les technologies de la micro-informatique. Pour toutes les bases critiques des applications verticales et business (et autres), la question peut aussi se poser dans la migration vers une version récente ou vers une autre base. La réponse ne peut être globale, mais au cas par cas.

Entre cycle de vie et cohérence

Il n'y a pas que l'application à avoir droit au cycle de vie. L'architecture des données impose aujourd'hui de mettre en place une véritable politique de cycle de vie des données et de la base de données. Bien entendu, cela n'est valable que pour les gros volumes, les bases multiples et dispersées. Les projets petits et moyens ou peu complexes n'auront pas besoin d'un cycle de vie complet, on pourra donc se limiter aux aspects traditionnels.

En fait, au lieu d'avoir un cycle de vie indépendant, il faut absolument le fondre dans le cycle de vie applicatif, car la donnée en fait totalement partie. On peut aussi parfaitement modéliser dans un même environnement les squelettes de l'application et de la (ou des) bases liées (ex : Power Designer de Sybase). Ce n'est pas pour rien que des éditeurs d'outils de cycle de vie, comme Compuware ,ou Parasoft, proposent des outils d'optimisation, de test et d'analyse de bases de données.

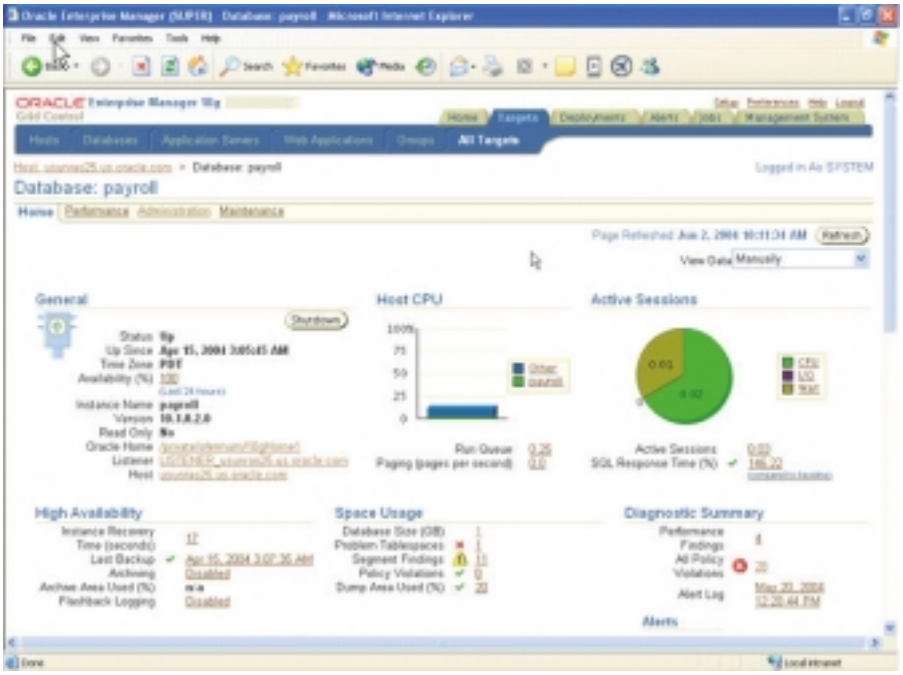
Le cycle de vie orienté données s'occupe aussi de la cohérence des données par rapport aux modèles et aux applications, ainsi que du stockage. La cohérence des données doit être une priorité du DBA et du développeur. À tous les niveaux, la cohérence est vitale. De cet élément dépend la qualité de l'analyse, du reporting et bien entendu du business, et du bon fonctionnement de votre application. Cette cohérence s'effectue du côté client, du côté serveur et entre les deux. Il faut penser à mettre des valeurs par défaut dans les champs, et définir les limites de saisies (mettre des contraintes). Vous pouvez bien entendu utiliser l'intégrité référentielle et/ou l'intégrité d'entité (validation des clés primaires). Vous pouvez aussi agir à un niveau plus haut, en implémentant dans la base des triggers, qui permettent de lancer des procédures à chaque modification. Même si vous travaillez en mode déconnecté avec des bases déportées, la base mobile (ou à défaut l'application mobile) doit s'occuper de cette cohérence. Puis, le serveur de données, à la réception des données, revérifiera cette cohérence (dans le cadre d'une réplication et/ou d'une synchronisation), surtout si les données mobiles sont transformées pour le transport. L'incohérence survient très vite et peut provoquer des dépassements de capacités et autres plantages.



Virtualisation et fédération des bases : intégration et monde hétérogène

Un autre problème survient souvent : l'hétérogénéité des données dans le format, le type, les bases de données. Car finalement, il arrive que l'on possède plusieurs bases de données, parfois homogènes, souvent hétérogènes, en local ou dispersé. Et là, il faut disposer des bons outils, des bons DBAs et des bons développeurs, surtout quand il faut attaquer les données de différentes sources. C'est pour cela que deux notions fondamentales apparaissent dans les SGBD " avancés " : la vir-

tualisation et la fédération. Pourquoi ? La raison en est simple. Une application manipule de plus en plus souvent différents types de données, parfois supportés par différentes bases de données. La fédération peut aider à éviter de tout migrer vers une source de données unique dans un premier temps, à cause des coûts et des problèmes de migration. Dans le cas d'une application Web ou Intranet, la fédération peut s'avérer utile pour fournir un unique frontal. Pour réaliser des analyses ou du reporting, la fédération s'avère aussi très utile. Encore faut-il disposer d'un SGBD capable de le faire. La virtualité poursuit un autre objectif : utiliser des données sans les déplacer (ni les bases). Dans le cas des bases géographiquement dispersées, la virtualisation permet donc de manipuler sans grand problème des données, sans pour autant les déplacer physiquement.



De plus en plus, dans le cadre d'applications lourdes et générant des trafics importants, l'éclatement de la base en plusieurs entités devient une nécessité pour des raisons de sécurité, de temps de réponse, de stabilité et de montée en charge. Il est dangereux de rester à la base unique quand son fonctionnement est intensif. Dans cette architecture, l'administration est vitale ainsi que la couche d'accès aux données.

Au-delà de la base de données

La base de données et la donnée ne constituent qu'une partie de tout ce qui sous-tend les problèmes de données. Outre le fait de disposer d'un SGBD adapté, d'une base parfaitement architecturée, il ne faut pas négliger tous les aspects connexes à la base et aux données. Cela concerne (comme on l'a vu un peu plus haut), le cycle de vie, les performances, l'optimisation, le caching, le monitoring, le tuning, le reporting, l'administration, la haute disponibilité, la réplication, etc. Que vous soyez un DBA ou un développeur, vous DEVEZ maîtriser tous les aspects de la base de données. Car, encore une fois, une base mal configurée, mal définie, mal déployée, mal " mappée ", mal optimisée, grève automatiquement et immédiatement vos applications (notamment dans les temps de réponse avec risque de déconnexion de sessions, etc.). L'environnement applicatif est global et non plus centré sur la seule application ! Le développeur doit savoir communiquer avec le DBA (et donc le comprendre) et le DBA doit comprendre le développeur. Dans les applications critiques et les applications Web à haut trafic, la gestion de la donnée est vitale. Une base indisponible pourra avoir des conséquences importantes sur le business ou tout simplement sur le bon fonctionnement de son logiciel.

Impossible en quelques pages de répertorier tous les éléments et critères à considérer. Il faudrait toutes les pages de Programmez! pour effleurer le sujet. Selon que vous êtes en .NET ou Java, si les principes demeurent identiques, la mise en œuvre ne le sera pas, et les technologies peuvent différer.

Réduire les accès : plusieurs méthodes possibles. La plus basique est d'utiliser un système de cache. On évite d'appeler constamment la base. La procédure stockée permet de réduire les accès, d'améliorer les temps de réponse et assure une meilleure sécurité dans les transferts. On stocke la procédure dans le serveur et non chez le client. Elle est précompilée. Cela évite donc de coder des requêtes SQL dans son application.

Optimiser son code : quand on doit interagir avec une base, on peut bien entendu coder en dur. Cela est valable pour des applications entièrement desktop / local ou en mode client/serveur peu exigeant. Mais dès que l'on découple application et base sur un client et un serveur, avec un volume important ou des besoins de performances, mieux vaut passer par une couche d'accès aux données qui fournit les interactions nécessaires entre son code et la base. Comme dit plus haut, les procédures stockées peuvent alléger le code et améliorer les performances, mais il ne faut pas pour autant tout mettre en procédure stockée. Vous pouvez aussi utiliser des outils de tuning et de monitoring, pour observer et détecter toute perte de puissance de l'application afin de la résoudre. Des outils de tests automatiques seront plus qu'indispensables pour les grosses applications.

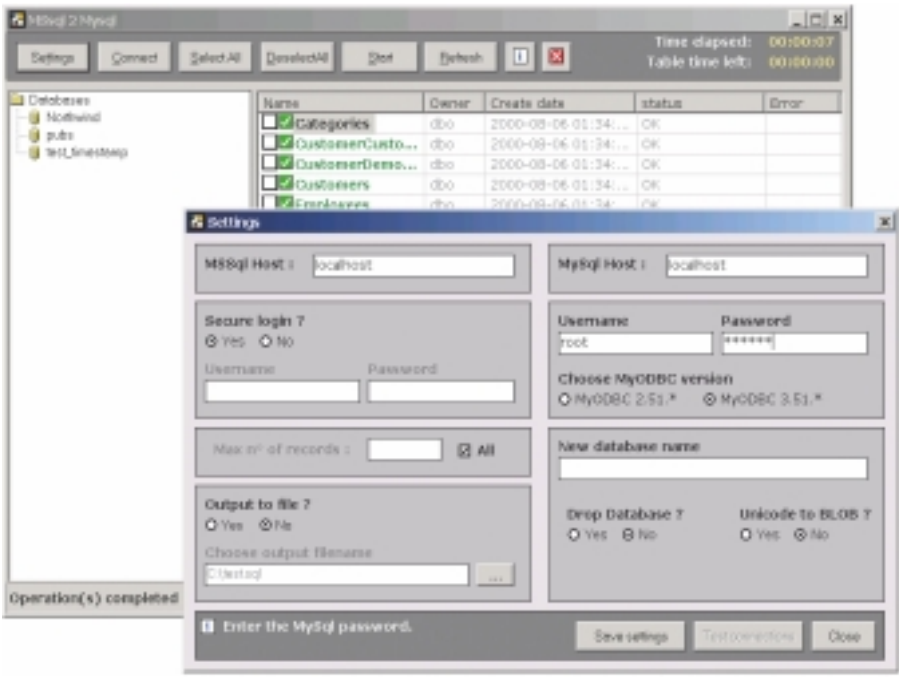
Réplication : voilà un mot qui revient tout le temps, la réplication. Elle est indispensable dans tous les cas : pour la sauvegarde de données, la reprise de charge, avec les données mobiles, etc. Les contraintes de réplication changent selon les impératifs : est-ce une réplication temps réel (une sorte de mirroring en cas de crash) ou par intervalle régulier ? Sur la totalité des données ou uniquement les données modifiées ? Qui peut l'autoriser ? Cette fonction est incontournable dans toutes les utilisations.

Haute disponibilité : c'est l'un des mots à la mode. Hier, d'un coût souvent prohibitif, aujourd'hui, elle est accessible aux petites

entreprises et projets non critiques. Quelques solutions libres existent ou existeront rapidement. Cependant, un choix conservateur semble plus judicieux. Pourquoi ? Car les acteurs traditionnels et propriétaires sont connus depuis longtemps et on connaît leurs solutions. Pour des outils libres, le recul n'est pas assez important pour juger de leur fiabilité. Dans ce domaine, il faut se montrer très prudent. Par contre, on peut réduire les coûts en prenant du Linux en système. Comme toujours, la haute disponibilité s'inscrit dans une architecture globale et surtout du matériel adapté et dédié. N'espérez aucune garantie sur des serveurs non dédiés et sans fonctions avancées (ex. : redondance, tolérance de panne...). La HD implique une architecture très spécifique pour le matériel et l'environnement applicatif (cluster, reprise de charge, répartition de charge, load balancing, réplication temps réel ou non, intégrité des données, gestion centralisée, outils de surveillance et d'analyse des charges, système de sauvegarde dédié...). On peut aussi y introduire les notions de virtualisation et de consolidation aussi bien du matériel, que des systèmes, que des données. Dans une telle architecture, le XML et les Web Services prennent une place importance. Attention : HD ne signifie pas obligatoirement cluster, le cluster n'étant qu'une des possibilités. La HD doit assurer la disponibilité des services (matériel et applications) mais aussi des données. Cela nécessite, malgré les progrès des outils d'administration et de configuration, des responsables extrêmement compétents. Si les outils libres peuvent représenter une source d'économies plus ou moins importantes, la HD revient très cher, ne serait-ce que par le matériel. Les développeurs doivent utiliser des outils adaptés pour pouvoir assurer ce type de fonctionnement. Vérifier que les serveurs de données et les autres fonctionnent en mode cluster / distribué. Des applications distribuées nécessitent des librairies et des compilateurs spécifiques. Le prix ne doit pas constituer votre unique critère de choix.

La mobilité

Avec l'explosion de l'informatique mobile, il est bien souvent nécessaire d'adapter applications et données. Là, deux logiques s'affrontent : connecté, déconnecté. Aucun de ces deux modes ne constitue une panacée technique. En connecté, on exploite les données du serveur et on manipule souvent du client



léger, voire du Web Service. Pratique, le connecté souffre des performances et des temps de réponse " réseau ". Le Wifi est une solution, mais limité à une zone de l'entreprise. Le GPRS tente de combler les soucis de performance en mobilité, mais ce n'est pas assez satisfaisant en terme de débit.

Le déconnecté présente l'avantage, en principe, de la performance, mais là, on aura tendance à installer du client lourd avec un moteur SGBD léger, voire des fichiers plats pour les données. Le principal problème du mode déconnecté est la synchronisation des données locales avec la base serveur. Tout un travail de réplication et de traitement s'avère donc nécessaire. Il faut privilégier les outils / couches de synchronisation côté serveur (MobiLink de Sybase, VoxSyn de VoxMobili, Cloudscape d'IBM, PointBase UniSync de DataMirror, etc.). Derrière cela, des API que l'on doit implémenter et coder dans l'application permettent d'intégrer les fonctions et manipulations nécessaires. La synchronisation implique naturellement l'application de règles de cohérence et d'intégrité des données (imaginez une table mobile et une table serveur ayant une différence dans la déclaration du type d'un champ, cela remet en cause l'intégrité des données). La réplication s'avère autant nécessaire que la synchronisation. Enfin, n'oubliez pas que le flux de la synchronisation (et de la réplication) peut se faire dans les deux sens, en descente et en montée. Dans ce cas, il faut aussi intégrer les mécanismes dans l'application et/ou le terminal mobile !

Les SGBD RAD ont-ils un avenir ?

La réponse est oui pour certains types d'applications. Ces solutions permettent de développer l'application à partir de la base dans un unique environnement et de déployer une seule application. On dispose aussi de la possibilité de publier, en quelques clics, des données sur le Web ou de les exposer en Web Services, voire de déployer en quelques instants sur PDA.

Gain de temps appréciable. Dans ce marché, citons notamment : WinDev / WebDev de PCSoft, FileMaker de FileMaker, 4e Dimension de 4D, Omnis de Raining Data, Visual FoxPro de Microsoft. Ce genre de solution est parfaitement adaptée aux TPE / PME / PMI, progiciels ou aux indépendants. Ils évitent de multiplier les outils, les langages et les environnements. Elles sont très présentes dans les professions libérales ou les solutions verticales. Ces solutions ne sont donc pas à oublier, surtout si vous souhaitez réaliser des applications orientées données et si vous n'avez pas besoin de tirer parti de technologies Java ou .NET.

François Tonic

Le mapping O/R : au cœur des applications

L'essor des architectures orientées objet a considérablement modifié la manière d'aborder l'accès aux bases de données relationnelles. D'abord décrié, puis délaissé au profit d'applications de niche, le mapping O/R fait aujourd'hui un tour de force remarqué au sein des applications d'entreprise, mais également des projets de moindre envergure. Quelles sont les raisons de cette soudaine notoriété ? Va-t-on vers un renouveau de ce modèle de développement, naguère réservé à un marché de niche ?

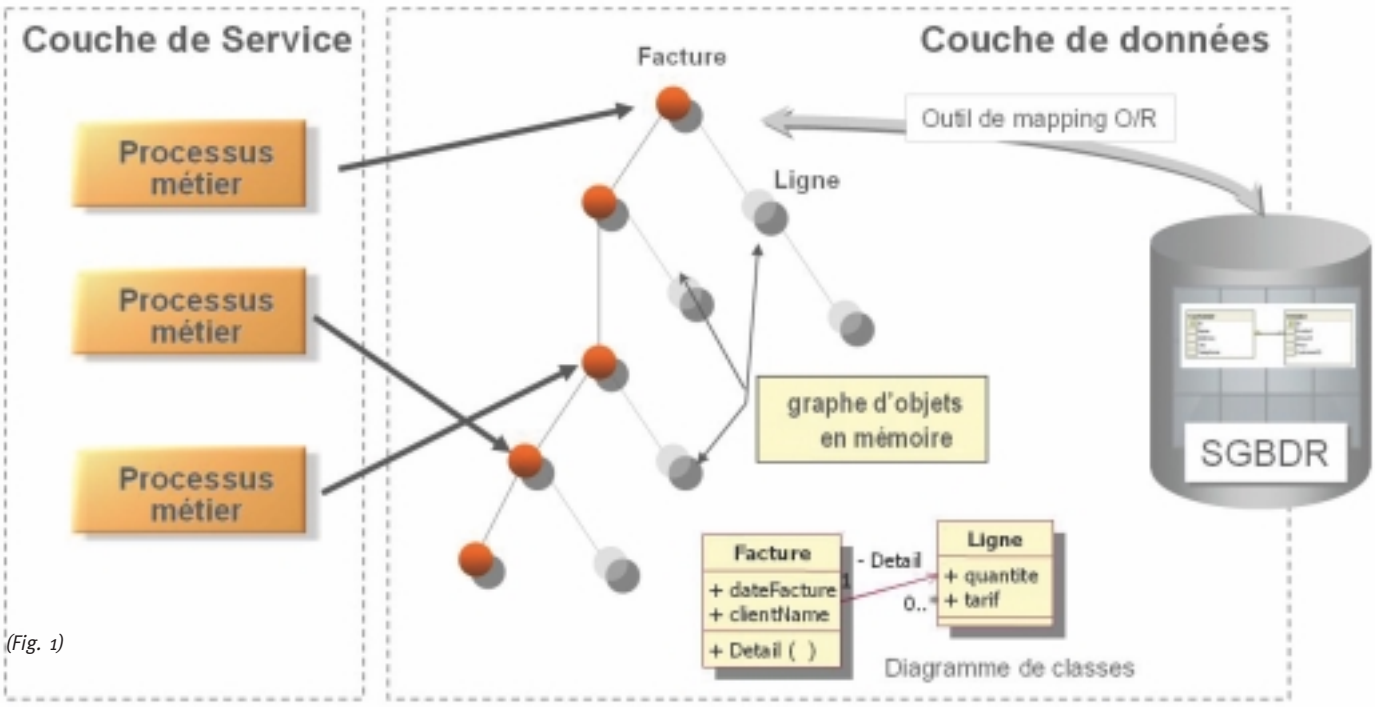
Nombre d'applications orientées objet comprennent un accès à une quelconque base relationnelle. Jusqu'à présent, pour peu que le développeur ait pris soin d'adopter une conception purement objet, une phase de conversion s'avère indispensable. Cette phase de conversion, encore appelée mapping objet/relationnel, consiste à adapter le modèle relationnel à un modèle objet, navigationnel et hiérarchique. Alors qu'un schéma purement relationnel représente les liens, via des mécanismes de clés étrangères, un modèle objet s'appuie sur des dépendances fortement typées (collections, classes, ..). Les caractéristiques intrinsèques du développement objet, telles que l'héritage ou le polymorphisme, s'accommodent difficilement d'une vue relationnelle. Ce problème a conduit, au fil des ans, les développeurs à effectuer d'incessantes pirouettes techniques, pour adapter ces deux modèles par l'utilisation de composants techniques spécifiques (pattern DAO pour Data Access Objects).

Si pour certains, la solution passe par l'acquisition d'une base de données nativement objet, pour d'autres au contraire, le problème est d'ordre applicatif. Les bases de données relationnelles ayant en effet prouvé leur maturité et leur capacité à monter en charge.

Quelle démarche ?

En J2EE ou .NET, il existe de nombreux motifs de conception, ou design pattern, permettant d'implémenter et d'optimiser manuellement (c'est-à-dire sans outil du marché) une couche d'accès aux données. Dans cette optique, une architecture multi couche s'avère souvent bénéfique, car elle tend à reléguer les phases de conversion dans cette couche spécifique appelée encore DAL (Data Access Layer) ou couche de persistance. Prendre en charge manuellement le mapping o/r nécessite de la rigueur de la part du concepteur. Car si cette démarche a l'avantage de tirer profit des compétences relationnelles existantes (SQL, procédures stockées, batch), elle devient vite fastidieuse sans un outillage minimum. En Java, cela consiste à faire appel aux API JDBC, afin d'extraire les données via des requêtes SQL, pour les restituer sous la forme de graphes d'objets. Plus le schéma relationnel est riche et complexe, plus cette tâche sera fastidieuse et répétitive. Sans compter les nombreux cas spécifiques, tels que le chargement à la demande (Lazy Loading), la gestion du cache objet ou la prise en compte de l'héritage. Qu'elle soit automatique ou manuelle, la persistance requiert la mise en place d'un modèle d'objets métier ou modèle du domaine. Ce modèle contient généralement l'ensemble des classes persistantes d'une application, indépendamment de leur structure de stockage (fichiers, sgbd, etc.).

Il constitue le pendant des tables relationnelles dans le monde objet. Nombreux sont les outils à proposer dans une approche bottom-up des générateurs de code, à partir de schémas relationnels. A l'inverse, une fonctionnalité incontournable dans le monde du mapping est la génération de schémas et de scripts SQL à partir d'un modèle objet (approche top-down).



(Fig. 1)

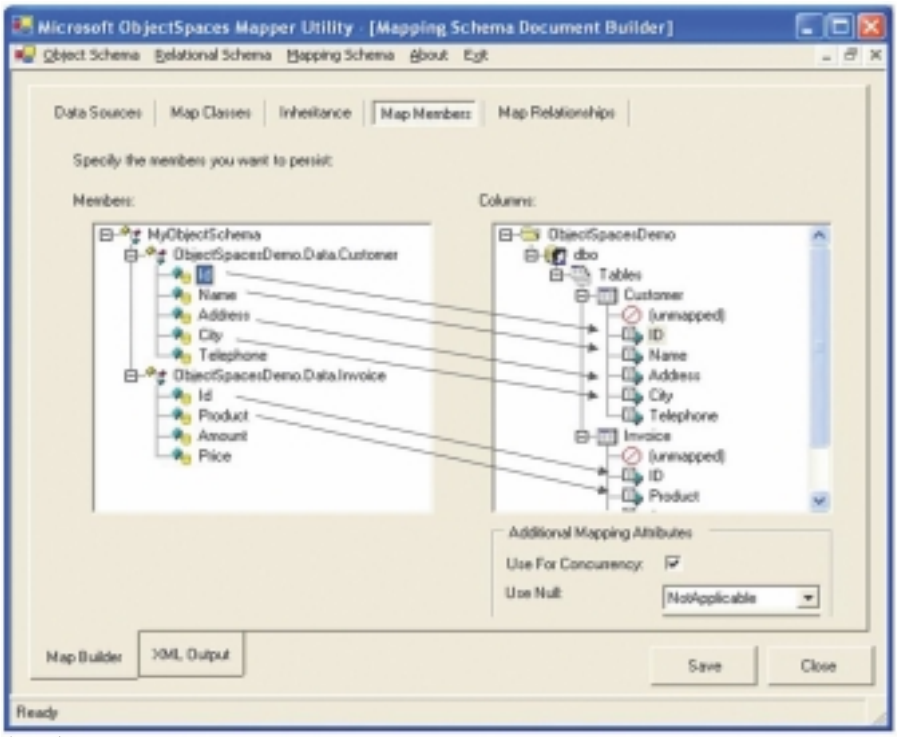
Les outils et standards J2EE et .NET

Les standards dans le domaine du mapping objet/relationnel ne sont pas légion. Seul le camp J2EE a pris l'initiative de créer une spécification : JDO. Spécification adoptée par la communauté Java, mais délaissée par les acteurs majeurs de l'industrie au profit de la future norme EJB 3 (point non encore totalement arrêté par Sun, ndlr). Pour ses détracteurs, JDO ne serait pas assez structurant. En réalité, se cachent derrière cette bataille, des enjeux financiers considérables. Contrairement aux EJB, JDO ne nécessite aucun serveur d'application particulier, seul un outil de mapping suffit à prendre en charge les différents services techniques de persistance. L'offre d'outils JDO étant déjà assez abondante et la plupart des éditeurs précédents ne fournissant aucun outil de ce type, le marché des serveurs d'application risquerait de subir le contrecoup d'une concurrence redoutable. Pour IBM "Cette spécification se chevauche avec d'autres JSR en cours de développement (ndrl EJB 3). Dans un contexte de rationalisation et de simplification des outils J2EE, il n'est pas souhaitable de voir se multiplier plusieurs modèles de développement concourant aux mêmes objectifs ". Ce qui explique le refus du géant de voter la dernière spécification JDO 2.0, l'une des plus aptes à supplanter les EJB Entité. En marge de ces travaux, et toujours côté Java, Hibernate attise toutes les convoitises. Récemment engagé aux côtés de la société JBoss et intégré au serveur d'application de même nom, Hibernate a réussi à faire l'unanimité dans la communauté, en l'espace d'un an, pour aujourd'hui représenter le socle de référence de la future spécification EJB 3. Oracle, dès 2002, s'était engagé dans le marché du mapping en rachetant la société Toplink, spécialisée dans ce type de solution.

Côté .NET, la situation est sensiblement différente. Microsoft n'a jamais caché son manque d'intérêt pour le mapping O/R, privilégiant une approche plus spécifique de l'accès aux données, avec sa solution de stockage SQL Server. L'indépendance que procurent ces outils vis-à-vis des bases de données ajoute par ailleurs au scepticisme de l'éditeur. Une stratégie qui est en train de progressivement évoluer avec le développement d'ObjectSpaces, le prochain outil de mapping Windows sur lequel viendront se greffer les fondations de Longhorn, annoncé pour 2006. ObjectSpaces fera partie de WinFS. Si Microsoft ne propose rien de convaincant pour le moment, il existe une trentaine de solutions de mapping pour .NET. Ils se nomment nhibernate (version .NET d'Hibernate), DTM (Evaluant), Pragmatier ou encore EntityBroker.

Les services techniques fournis

Le premier service rendu par un outil de persistance est la correspondance ou "mapping". Le mapping consiste à associer le graphe d'objets aux tables relationnelles. Cette opération est généralement formalisée dans un fichier au format XML, généré par une interface graphique spécifique (figure 2). Tout accès à la méthode d'un objet se traduit par l'invocation d'une requête SQL. Et, contrairement à une idée reçue, la recherche d'une collection d'objets dans une table contenant un million d'enregistrements n'implique pas le chargement d'un million d'ob-



(Fig. 2)

jets en mémoire, sauf demande explicite de l'utilisateur. Cette fonctionnalité est intimement liée à un autre service technique plus connu : le cache objet. Contrairement à une gestion manuelle du mapping, le cache objet assure que la recherche multiple de données non modifiées n'implique pas l'exécution multiple d'une même requête SQL. En terme de performances, le cache est un atout considérable dans le choix de l'adoption d'un outil. Certaines applications possèdent de nombreuses données en lecture seule, et peuvent voir leurs performances multipliées par dix, uniquement grâce à l'emploi d'un cache. Dans le jargon technique des outils de mapping, le cache s'appuie sur la session utilisateur, aussi appelée PersistenceManager.

D'autres services, tels que le chargement à la demande permettent de naviguer dans un graphe d'objets en chargeant les données, uniquement lorsqu'elles sont sollicitées par l'utilisateur. Une fonctionnalité complexe à implémenter manuellement, car nécessitant la prise en charge d'importants paramètres, tels que la profondeur et le nombre de relations entre objets. La gestion du cycle de vie, l'identité, les verrous ou la gestion des transactions apportent également une importante plus-value technique aux Frameworks de mapping. Leur évaluation passe souvent par la qualité des services qu'ils proposent et a un impact direct sur les performances, mais également sur la complexité de paramétrage de l'outil. Un aspect loin d'être négligeable.

L'administration

L'administration et le tuning sont indéniablement les points faibles des outils de mapping. Nombreux sont les DBA à regretter l'absence de fonctionnalités natives, telles que des analyseurs de plan d'exécution dédiés aux langages de requête objet. En effet, une base de données relationnelle propose l'optimisation des performances, via l'analyse préalable des requêtes SQL les plus fréquemment employées. Or, un outil de mapping génère de manière dynamique ses requêtes, au gré de la navigation au sein du modèle objet. Difficile dans ce contexte d'opérer une analyse statique. Très souvent, la base de données continue à assurer son rôle d'optimisateur de requêtes. Une problématique d'ores et déjà adressée dans de nombreux outils, mais loin de constituer la panacée. Une autre critique, souvent formulée à l'encontre du mapping, est son manque de transparence lorsqu'il s'agit d'auditer les communications entre la couche applicative et la base de données. Le but de ces outils étant de masquer la complexité du langage SQL, toute tentative d'optimisation bas niveau s'apparente souvent à un parcours du combattant. Enfin, certains produits ne permettent pas d'intégrer ou de reproduire fidèlement des fonctionnalités existantes au sein des SGBD actuels, notamment les triggers ou les procédures stockées.

Les langages de requête

Les langages de requête, tout comme l'administration, font partie des autres tendons d'Achille du mapping objet/relationnel. Chaque outil possède son propre langage et difficile de s'y retrouver dans le dédale des offres du marché. Les seules API ayant fait l'objet de spécifications sont ODMG OQL, JDOQL et EJBQL. En pratique, l'objectif premier d'un langage de requête objet est d'assurer :

- La neutralité par rapport au langage de la base : le langage sous-jacent peut être du SQL, du OQL ou n'importe quel système spécifique (mainframe, ...)
- L'optimisation de l'accès aux tables : il doit être possible de profiter de la puissance des langages sous-jacents, tels que SQL, pour étendre l'interface de requêtes (jointures, ...).
- L'adéquation avec les environnements n-tiers : il doit être possible d'effectuer des requêtes entièrement en mémoire, ou déléguer leur exécution à un système tiers.
- Le support des gros volumes de données : L'interface de requêtage doit gérer les gros volumes d'enregistrements de manière optimale, sans instancier au préalable d'objets.
- Les requêtes précompilées : il doit être possible de précompiler une requête, afin d'optimiser les appels ultérieurs.

Microsoft avec ObjectSpaces privilégie le langage OPath. Un langage maison permettant d'effectuer des recherches en utilisant un formalisme proche d'OQL. D'autres s'appuient sur une syntaxe dérivée de SQL avec des adaptations objet, c'est le cas notamment d'Hibernate et son langage HQL. Au fil du temps, le monde du mapping objet/relationnel a vu se succéder nombre de produits, tour à tour désignés comme référence. Les premiers à ouvrir le bal furent Toplink, ObjectStore (ObjectDesign) ou Versant. Aujourd'hui, c'est au tour d'Hibernate de tenir le haut de l'affiche. Une tendance qui risque fort de s'inverser prochainement, avec l'entrée en lice des poids lourds du marché engagés dans les futures spécifications J2EE. En effet, le mapping objet/relationnel fera partie des grands chantiers de cette norme et la plupart des poids lourds de l'industrie, spécialistes des bases de données, ont commencé à se positionner sur l'échiquier des outils de nouvelle génération, plus performants et moins intrusifs.

Sami Jaber

SQL Server 2005 arrive-t-il trop tard ?

Cinq années ont passé depuis la sortie de la dernière édition de SQL Server, qui a permis à Microsoft de devenir un acteur majeur sur le marché des bases de données relationnelles, et de concurrencer directement les leaders du secteur que sont Oracle ou IBM DB2, tout du moins sur Windows. Mais après cinq années de bons et loyaux services, SQL Server 2000 n'est plus en mesure de soutenir la comparaison avec ses principaux compétiteurs, et ce, dans de nombreux domaines : stockage XML, intégration de langages-objets pour les procédures stockées, ou encore performances sur plate-forme 64 bits. Une refonte importante du produit s'imposait donc. Elle arrive sous le nom SQL Server 2005 et sa sortie est attendue dans le courant de l'année prochaine.

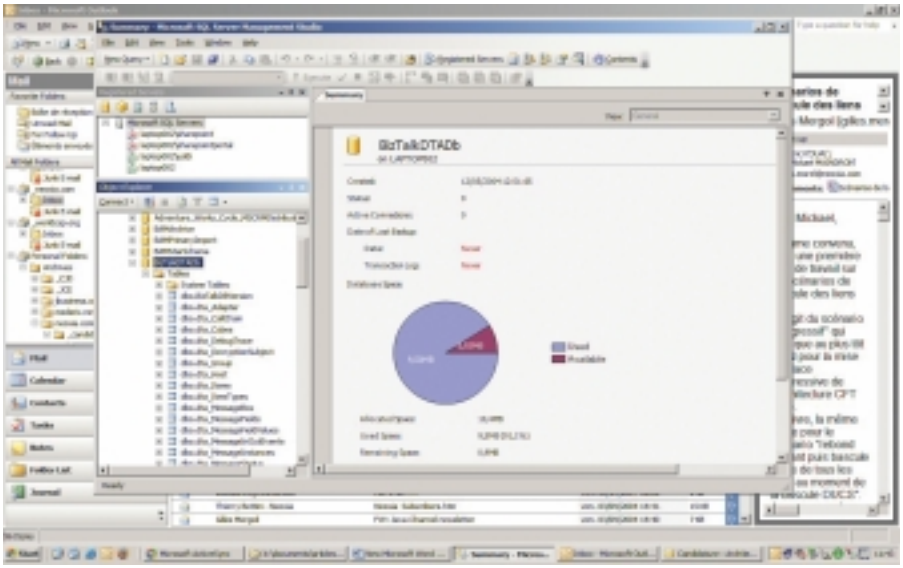


Fig 1 : l'environnement de Yukon

Convergence avec .NET

Yukon est le nom de code de la version bêta de SQL Server 2005. Cette nouvelle version est une refonte majeure du produit.

La première évolution concerne l'intégration en son sein du .NET framework 1.2, ce qui ouvre la voie à l'écriture de procédures stockées avec des langages utilisant le CLR (Visual Basic .NET, C#). SQL Server rejoint en cela Oracle qui permet déjà depuis longtemps l'intégration de Java dans son SGBDR et DB2. Le langage Transact-SQL reste néanmoins disponible et bénéficie de quelques améliorations. On dispose donc désormais de deux langages différents pour l'écriture des procédures stockées. D'une façon générale, il vaut mieux rester en Transact-SQL pour les traitements de données simples et réserver .NET pour les traitements effectuant des calculs, ou faisant intervenir des logiques métier particulièrement complexes. Par ailleurs, il est désormais possible d'exposer des procédures stockées sous la forme de Web Services pour les rendre accessibles à des applications extérieures.

La seconde évolution concerne le support de XML. SQL Server 2000 avait déjà esquissé une première intégration (très partielle) avec l'ajout des fonctionnalités FOR XML et OPENXML. Avec Yukon, Microsoft franchit un pas supplémentaire en offrant le support du standard XQuery et le stockage XML natif avec la création d'un nouveau type de données. Il devient également possible d'associer des données XML avec des schémas XSD pour en garantir l'intégrité. La convergence tant attendue entre SGBD et XML commence a devenir une réalité concrète.

Le retour de la mini informatique ?

Une autre évolution majeure de SQL Server 2005 concerne l'apparition d'un nouveau module baptisé " Service Broker " qui ouvre la voie aux traitements asynchrones à l'intérieur du serveur. C'est une évolution majeure, car pour des usages simples, l'utilisation d'un broker de messages de type MSMQ ou d'un moteur EAI comme Biztalk n'est plus indispensable. Avec le " Service Broker ", il est désormais possible de développer des applications mettant en oeuvre des traitements et des logiques métier complexes qui seront hébergés dans le SGBD. Cela transforme de facto SQL Server 2005 en serveur d'application orienté données. Cela n'est pas sans rappeler l'ère de la mini informatique et des solutions "tout intégré" sur le modèle de l'AS 400, les aspects hardware en moins.

Enfin, les outils d'administration, de développement et de supervision ont tous adopté l'ergonomie de Visual Studio .NET, ce qui n'est pas sans avantages en matière d'ergonomie et de facilité de prise en main. (Fig : 1)

Alors, SQL Server 2005 arrive-t-il trop tard pour contrer les offres d'Oracle et IBM ? Ses nouvelles fonctionnalités sont-elles de nature à changer la donne sur le marché des SGBD ?

De toute évidence, Microsoft accuse aujourd'hui un certain retard par rapport à Oracle (et à IBM avec DB2 Stinger) sur nombre de fonctionnalités, mais il convient de faire la distinction entre les fonctionnalités disponibles et celles qui sont réellement utilisées dans les développements. Les entreprises ont des comportements très conservateurs à l'égard des

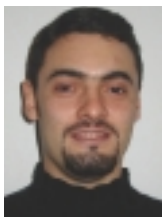
bases de données, car elles ont parfois appris à leurs dépens les coûts que peuvent occasionner la dépendance vis-à-vis des éditeurs, en particulier lors de changements de produits. Elles utilisent avec parcimonie les extensions propriétaires proposées par les éditeurs, et encore, uniquement lorsque la valeur ajoutée est importante et durable. De leur côté, les éditeurs sont sous la pression de solutions Open-Source telles que MySQL, dont les performances sont dans bien des cas comparables à celles de leurs produits. Ils sont donc contraints à une surenchère sur les fonctionnalités et à une intégration de plus en plus poussée avec les autres produits de leur gamme. SQL Server 2005 illustre parfaitement cette situation.

SQL Server 2005 est un produit complet qui offre une intégration poussée avec .NET et qui saura apporter une réelle plus-value aux entreprises qui ont fait le choix de Windows côté serveur. Son arrivée un peu tardive par rapport à DB2 et à Oracle pour le support d'XML et des langages-objets n'aura guère d'incidence, dans la mesure où, d'une part, il ne s'adresse pas exactement au même public, et, d'autre part, l'adoption réelle de ces technologies du côté des bases de données ne se fait que de façon très progressive.

Médéric Morel

Directeur technique de Neoxia, cabinet de conseil en architecture des systèmes d'information. mederic.morel@neoxia.com

Les nouveaux enjeux du rôle du DBA



On ne peut aujourd'hui aborder les nouveaux enjeux du rôle de DBA (DataBase Administrator) sans rappeler les évolutions subies par cette profession au sein des services informatiques ces dernières années. Le tableau suivant présente les étapes de cette évolution.

	Fin des années 80	Milieu des années 90	Fin des années 90
Technique révélatrice	**Client/Serveur puis Client léger**		**Data Wharehouse**
Besoins base de données	Stockage Restitution	Stockage Restitution RPC (Remote Call Procedure)	Stockage Restitution RPC (Remote Call Procedure) Optimisations avancées Fonctions OLAP
Fonctions DBA	Installation des SGBD, Administration de 1er niveau, Opérations de sécurité, Sauvegarde / restauration.	… + Connaissance des langages de développement (PL/SQL,Transac SQL) Bonne connaissance des modèles physiques	… + Tuning avancé du SGBD et des traitements. Connaissance du MCD et du fonctionnel métier.

On remarque que les évolutions majeures des systèmes de gestion de bases de données et donc du métier de DBA au cours des 15 dernières années résultent dans un premier temps de l'apparition des architectures en client léger et plus récemment de l'informatique décisionnelle.

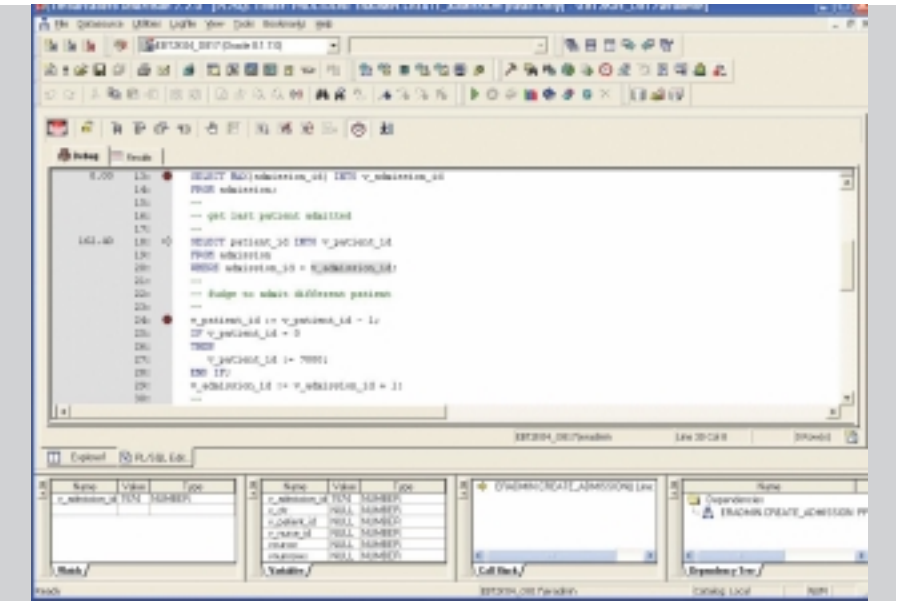
A la fin des années 80, les SGBD jouaient essentiellement un rôle de stockage de données. Afin de répondre à ces exigences, le profil du DBA était essentiellement technique et spécialisé. Le DBA s'attachait alors à des tâches d'installation et d'administration qui l'isolaient généralement des équipes projets.

L'apparition du client/serveur et par la suite des clients dits légers ont entraîné le déplacement des traitements vers les serveurs de base de données au moyen des procédures stockées. Il a donc fallu que la fonction de DBA s'agrémente de la maîtrise des langages de développement inhérents à chaque SGBD (PL/SQL, Transac SQL). De plus, les DBA ont dû acquérir de nouvelles compétences très fortement liées au développement telles que l'architecture logicielle et l'optimisation des traitements.

La fin des années 90, a vu naître les concepts de Data Wharehouse qui ont également apporté à la fonction d'administrateur de base de données son lot de tâches supplémentaires. En effet, l'abandon des modèles en 3ème forme normale et l'adoption de la redondance des données ont fait voler en éclats les volumétries jusqu'alors atteintes. Même si les principaux éditeurs de SGBD ont su s'adapter à ces nouvelles contraintes de performances, les opérations de tuning demandent aujourd'hui des connaissances de plus en plus pointues et surtout impliquent le DBA dans les équipes projets pour l'optimisation des traitements.

On peut remarquer que cette implication de plus en plus forte des DBA dans les problématiques de performances des traitements s'est vue amplifiée par l'engouement des DSI ces dernières années pour les nouvelles technologies. En effet, ces architectures très consommatrices de matière grise ont bien souvent accaparé l'intégralité des budgets et donc des ressources, reléguant la base de donnée au rôle de "parent pauvre" des projets informatiques. Vouloir conserver une telle organisation lors de projets Data Wharehouse serait bien évidemment source d'échec, on assiste donc à un revirement de situation, avec l'attribution de budgets non négligeables aux projets décisionnels. En revanche, le manque d'investissement dans la formation de ressources orientées bases de données se traduit aujourd'hui par une carence certaine de compétences dans le développement des traitements stockés; C'est alors tout naturellement que les décideurs se sont tournés vers les DBA.

Ce revirement de situation a également été rendu possible par l'évolution des SGBD qui ont intégré des outils prédictifs du comportement de la base, en vue de faciliter l'administration. On peut citer par exemple les vues du dictionnaire suffixées par ADVICE de la version 9i d'Oracle (V$DB_CACHE_ADVICE...), tendance largement confirmée dans la version 10g, incluant une interface d'aide à l'administration. Du DBA d'exploitation est donc née une nouvelle fonction, le "DBA d'étude" ou encore "DBA fonctionnel". Cet expert des systèmes de gestion de bases de données s'implique de plus en plus dans la conception et la réalisation des alimentations de données. Son rôle



consiste à modéliser et interpréter au mieux les besoins de la maîtrise d'ouvrage, afin de pouvoir assister la maîtrise d'œuvre en terme d'architecture des traitements et d'optimisation des performances.

Le dialogue avec les maîtrises d'ouvrage constitue souvent un challenge important pour cette fonction à la formation très technique et peu encline à une approche fonctionnelle. Ce tournant a eu l'avantage de rapprocher le DBA de la donnée elle-même. En effet, pour reprendre l'image d'un de nos clients qui compare les Data Warehouse à "une machine à laver la donnée", nous constatons qu'il est essentiel d'évaluer régulièrement la "dérive" des données au sein des systèmes d'information. Il est alors indispensable de posséder un minimum de connaissances métier afin d'interpréter convenablement la violation de telle contrainte ou l'absence de telle donnée, etc.

A cette tâche de contrôle vient souvent s'ajouter la tâche herculéenne d'unification des référentiels métier qui va amener le DBA à participer à de nombreuses réunions, lui permettant de découvrir les différents pôles de l'entreprise et leurs référentiels locaux. On lui demandera donc de présenter une curiosité naturelle et une forte capacité d'adaptation aux différents interlocuteurs qu'il sera amené à rencontrer. En dehors de cette approche fonctionnelle, les nouvelles tâches du DBA consistent à pouvoir faire face aux nouvelles contraintes de performances imposées par le monde décisionnel. En effet, lorsque l'on interroge ou que l'on charge un Data Warehouse, on atteint rapidement les limites des SGBD; Il est donc nécessaire de parfaitement connaître les optimisations à apporter dans ce contexte. Nombre de DBA ont une connaissance théorique des mécanismes à mettre en place sans avoir eu, pour le moment, le besoin de les mettre en pratique. On peut notamment citer les techniques de partitionnement, d'index Bitmap ou encore de parallélisation. D'autre part, les règles qui régissaient jusqu'à présent l'administration des systèmes de gestion de bases de données périclitent face à ces nouvelles contraintes. Si l'on prend l'exemple du SGBD d'Oracle, il existe une technique de sauvegarde à froid des données qui consiste à créer un fichier "dump" à l'aide de l'utilitaire d'export (EXP), on peut également utiliser cet outil pour faire de l'échange de données entre différentes instances ou différentes plates-formes. Cette méthode très largement utilisée dans le monde relationnel devient obsolète en termes de performances lorsqu'il s'agit d'exporter et surtout d'importer plusieurs terabytes de données. On peut également remarquer qu'en terme d'activité, l'informatique décisionnelle présente de nouvelles contraintes. En effet, le moteur de la base de données, va alterner entre des sollicitations fortes de requêtes Multi-Utilisateurs lors des périodes de reporting (généralement en journée!) et des périodes de chargements massifs de données lors de traitements batchs d'alimentation. En terme de paramétrage, il faut donc pouvoir gérer dans une même instance des comportements hétéroclites, ce qui amène à mettre en place une sorte de paramétrage dynamique du SGBD et des optimisations des traitements fonction du contexte d'exécution. Pour Oracle, il est notamment conseillé, lors d'un chargement massif, de supprimer certains index utiles à la restitution et de les recréer une fois le chargement terminé. Il s'agit une fois de plus d'une technique dont l'utilité est très limitée au sein de systèmes manipulant quelques centaines de mégabytes de données.

En résumé, le DBA d'aujourd'hui et de demain est très loin de l'image "du technicien solitaire adepte des salles blanches" qu'on a bien voulu lui donner ces dernières années. Il lui incombe dorénavant de supporter (au sens noble du terme) les équipes projet (MOE et MOA). On assiste donc à un "retour aux sources" qui amène le DBA à se focaliser sur la "Donnée" elle-même : quel est son sens ? D'où vient-elle ? Où va t'elle ? Comment optimiser son traitement ? ... Il doit bien sûr garder un œil de technicien averti sur les évolutions des différents SGDB, afin d'être à la pointe des méthodes d'alimentation, de stockage et de restitution. Il s'agit d'un tournant difficile à prendre pour cette fonction jusqu'à maintenant cantonnée à des tâches d'administration technique mais qui permettra, sans aucun doute, à ceux qui franchiront le pas de revaloriser leur position au sein de l'entreprise.

Sébastien Flourac - *Groupe SQLI*

Qu'est ce qu'un modèle en 3ème forme normale ?

Il découle d'une modélisation adaptée aux SGBDR et tendent, au contraire du décisionnel, à normaliser au maximum les données en évitant toute redondance.

Il existe 3 processus de normalisation :

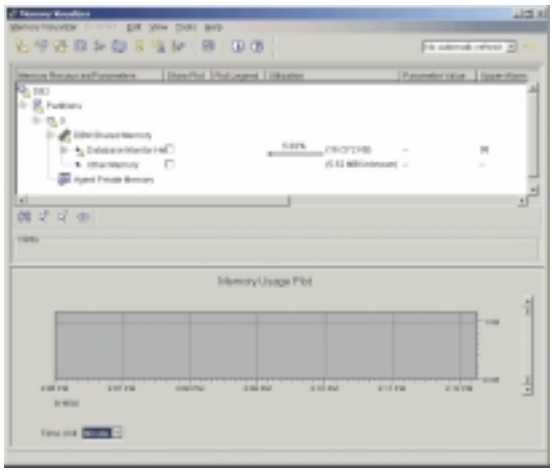
A - Un tableau est en première forme normale : s'il admet au moins un clé primaire, si toutes ses colonnes sont élémentaires (c-à-d une et une seule valeur)

B - Un tableau est en deuxième forme normale : s'il est en première forme normale, si chacun des attributs non membres de la clé primaire est en dépendance fonctionnelle totale par rapport a cette clé

C - Un tableau est en troisième forme normale : s'il est en deuxième forme normale, si chacun des attributs non membres de la clé primaire ne dépend pas transitivement de cette clé

La virtualisation : nouveau credo de DB2

IBM pousse depuis quelques mois DB2 hors du monde mainframe. L'éditeur a bien du mal à investir massivement les marchés Linux et Windows sur lesquels Oracle et Microsoft dominent. Avec le nouveau DB2, IBM espère faire mieux. Rencontre avec Isabelle Calverie-Berge, consultante IBM Data Management.



Programmez : IBM vient de sortir la nouvelle évolution de DB2. Quelles sont les nouveautés importantes à retenir ?

Isabelle Calverie-Berge : Nous avons sorti une nouvelle release de DB2, nom de code Stinger. Stinger concerne surtout le moteur de base de données. Parmi les nouveautés, il y a l'amélioration de SQL, la disparition du compilateur C quand on utilise SQL PL, la taille des procédures stockées ou encore la sécurité. Sur ce point, nous avons ouvert l'authentification. On peut rajouter des plug-ins pour avoir de nouvelles authentifications. On peut aussi maintenant crypter l'ensemble du flux de données. On supporte .NET, la CLR et C#. Nous avons toujours visé le marché Windows, nous suivons les évolutions de la plate-forme. Le support natif de .NET peut certainement aider DB2. La base de données n'est plus qu'un composant dans l'entreprise.

Programmez : quelles sont, pour IBM, les grandes tendances dans la base de données et notamment avec DB2 ? Pensez-vous que le temps réel va être un domaine de recherche du futur ?

IC-B : Nous avons 2 tendances : les très grandes bases de données et la virtualisation. Je virtualise les données pourquoi ? Je gère de manière centralisée les différentes bases. On peut gagner 20 à 30 % sur le code des applications grâce à la virtualisation. On déplace une donnée sans modifier l'application. Tout ce qui concerne la haute disponibilité et la réplication sont des thèmes forts pour IBM. Aujourd'hui, nous faisons très bien du 1 to N. Avec Masala (http://www-306.ibm.com/software/data/integration/masala.html), on fera de la réplication avec du quering. Avec le HADR, on fait de la réplication à l'identique. On peut stocker des données dans différents formats. On fera sans doute du relationnel objet. Pour le temps réel, on sait faire de l'alimentation de données en quasi temps réel. Il y a des travaux autour de cette notion.

Programmez : Les SGBD libres sont-ils des concurrents sérieux pour DB2 ?

IC-B : On rencontre les SGBD libres. Ces solutions n'ont pas les fonctions très avancées que l'on peut avoir dans DB2. On ne cible pas forcément les mêmes marchés et utilisateurs.

Programmez : Avec la complexité croissante des bases de données dans les fonctions et l'architecture, va-t-on voir apparaître de véritables développeurs estampillés SGBD, prenant place à côté du DBA ?

IC-B : C'est une bonne question. La tendance est tout de même à la simplification. On fournit des plug-ins pour optimiser le développement. On va plutôt dans le sens d'un DBA côté développement. Aujourd'hui, cela me semble indispensable. On aura peut-être des développeurs spécialisés sur certaines fonctions précises comme les procédures stockées. On soulage aussi le travail du DBA sur certaines fonctions, par contre, son rôle devient de plus en plus important dans l'optimisation, l'intégration et le développement.

Programmez : On parle beaucoup ici et là du XML, comme format de stockage. Plusieurs éditeurs le font déjà ou le feront. Est-ce viable et performant ?

IC-B : Il existe du XML s'adaptant parfaitement au stockage de données, même si XML sert avant tout à publier. On sait stocker en XML. Ce sujet va bouger dans l'avenir. Aujourd'hui, on réalise des choses simples. Dans les gros volumes, il faut des outils et fonctions complémentaires, il faut être scalable.

■ *Propos recueillis par* **François Tonic**

DB2 Stinger : du gros et du lourd !

La nouvelle release du SGBD d'IBM ne ménage pas ses efforts pour convaincre avant l'arrivée de SQL Server 2005. Les focus de cette version sont principalement : simplification du développement, limitation de l'intervention du DBA, qualité des services de la base. Stinger se positionne notamment plus fortement sur Windows / .NET, et Linux. Il s'agit de proposer une meilleure intégration de l'outil avec Java, Windows, .NET, Linux, WebSphere et les outils tiers, tout en incluant plus de mécanismes d'automitic computing. Le support .NET est assuré par l'intégration avec VS.NET 2003 et le support des procédures stockées CLR. Les plug-ins DB2 pour VS.NET doivent fournir une transparence aux développeurs et aux applications. Pour les entreprises, le business intelligence a été un axe fort de travail ainsi qu'une automatisation de la configuration et administration afin de décharger au maximum le DBA. La haute disponibilité demeure bien entendu une des priorités de Stinger avec l'architecture HADR pour les données critiques. Architecture reprise de celle d'Informix. Elle doit assurer la disponibilité des données en cas de crash ou de problèmes sur des bases. Les fonctions de réplications ont été aussi revues et corrigées en profondeur.

Éliminer la complexité !

Le marché du mapping et des SGBD bouge beaucoup et 2005 promet d'être chargé. Nous avons voulu en savoir un peu plus en interviewant le directeur technique de NET2S, Rodrigo Laurens. Attention interview sans langue de bois !

Programmez ! : on parle de plus en plus de la base de données. S'agit-il d'un mouvement de fond déjà ancien ? L'objet que l'on voit partout pose-t-il problème quand on veut l'utiliser avec des données ?

Rodrigo Laurens : Dans beaucoup d'applications Web, la base de données est déjà présente. La question est de savoir comment lier l'application à la base. Dans un dialogue technique, il n'y a pas de valeur ajoutée. Avec l'objet, par contre, c'est un nouveau problème : la continuité du modèle (ndlr : modèle relationnel, modèle objet). Il existe deux espaces très différents à concilier. Comment faire un modèle cohérent ? On utilise souvent un SGBDR. On voit l'émergence de l'objet / relationnel qui essaie de réduire la fracture.

Programmez : Faut-il développer à la main, ou passer par un framework ?

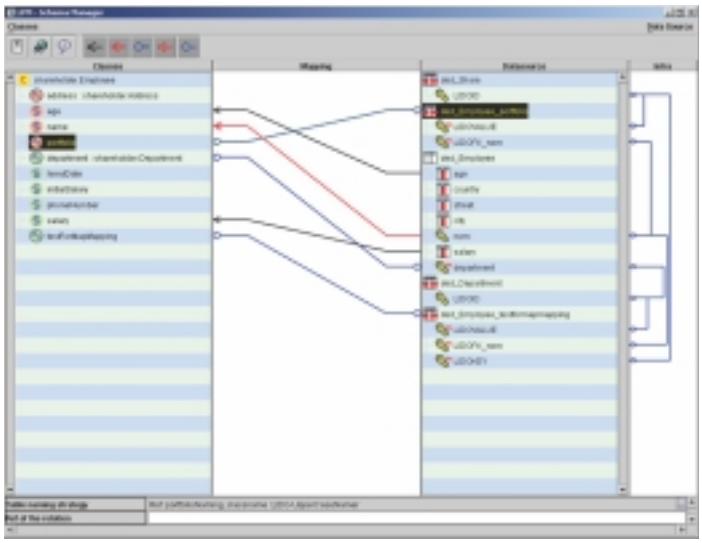
RL : ça dépend. La tendance est au développement avec framework. Il fournit un squelette, des parties basses. C'est identique avec le framework de mapping. Hibernate est un modèle de mapping alternatif à Sun. Sous .NET, la situation est là même avec ObjectSpaces encore en développement, mais le modèle Hibernate intéresse aussi .NET. Modèle assez riche, EJB 3.0 doit régler les problèmes actuels. Sun propose aussi JDO, mais dans la pratique, Hibernate est très utilisé. JDO dans ses débuts n'était pas terrible, la plupart des implémentations étaient complexes et de qualité pas optimale. Nous utilisons plus Hibernate que JDO, même si la version 1.1 améliore beaucoup les choses. La version 2.0 promet d'être encore meilleure et pas uniquement le problème du mapping. Il y aura un cache pour les performances. Cependant, il sera un peu plus complexe à utiliser.

Programmez : On parle souvent de complexité de J2EE et du mapping en général. Est-ce la réalité ? La complexité ne vient-elle pas d'une erreur d'architecture ?

RL : J2EE est complexe. Il fonctionne bien, mais il faut l'améliorer. .NET est plus simple. Les ténors de l'informatique proposent JDO, mais le développeur cherche des solutions plus simples. Si on fait de l'orienté objet, la base de données sera complexe. Bien souvent, les clients mettent les processus métiers directement dans la base ! Dans ce cas, le mapping devient encore plus complexe, or le mapping est fait pour simplifier. Ainsi dans les applications offshore, le mapping s'occupe de créer la base, la base étant uniquement un lieu de stockage.

Programmez : Dans ce cas, pourquoi ne pas créer systématiquement un découplage entre le mapping, la base et les aspects métiers, cela apporterait une architecture bien plus souple ?

RL : Idéalement c'est ça. Dans ce cas, le mapping apporte une valeur. Dans la réalité, il y a un problème de culture. L'entreprise veut mettre les aspects métiers dans la base. Cela commence tout de même à changer, mais il faudra du temps. On continue à faire de la base de données un noyau business.



Utilisez des outils dédiés pour mieux gérer la complexité

Programmez : On constate depuis longtemps que les SGBD n'en finissent plus de grossir en fonctionnalité. Tout de même, cela ne pose-t-il pas un problème de maîtrise ?

RL : Oui, l'objectif de mettre tous les éléments dans une base ne simplifie pas la vie. C'est un problème de culture et de marketing.

Programmez : un laboratoire de recherche de Microsoft a présenté un langage, C Omega, qui permettrait de manipuler du SQL, du XML et des objets. Ne serait-ce pas un avenir intéressant que de pouvoir, avec un langage, manipuler du relationnel et de l'objet ?

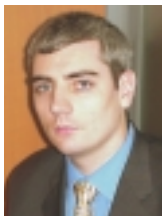
RL : Oui, peut-être, mais je suis un peu sceptique. Actuellement, le mapping est complexe. Il faut arrêter de construire des usines à gaz. Sous J2EE, Spring va dans le bon sens en s'attaquant à cette complexité en proposant un ensemble plus ouvert, plus simple. Il existe aussi un portage de Spring sous .NET.

Programmez : XML est partout. A-t-il véritablement simplifié les choses ? N'a-t-on pas détourné XML de sa tâche première ? Faut-il le généraliser ?

RL : Non je ne le pense pas. Si XML devait effectivement simplifier les choses, maintenant c'est une grosse usine à gaz. XML se transforme en tour de Babel. On ne fait qu'ajouter des couches. La faute incombe aux grands éditeurs. Le flux XML est plus complexe et les performances tombent, car le parsing XML prend beaucoup de ressources. Il faut l'utiliser uniquement quand il est nécessaire. Dans les applications Web, l'usage du XML alourdit l'architecture, mais c'est de plus en plus à la mode !

■ *Propos recueillis par* **François Tonic**

Sybase, le 4e éditeur



Sybase, un des grands éditeurs de SGBD, conserve 5% de parts de marché et des comptes importants, grâce à des solutions dédiées.
Rencontre avec Bertrand Audras, Senior Presales Engineer (Sybase France).

Programmez ! : Les cabinets d'analystes créditent Sybase de moins de 5 % de part de marché, en 4e position après Oracle, IBM et Microsoft. Cela reflète-t-il une réalité ?

Bertrand Audras : Cela reflète notre marché, même si ce sont des pourcentages globaux. Notre marché historique est la banque / finance notamment dans les salles de marché, les bases mobiles avec des technologies dédiées. Nous avons même créé une société spécifique à marché, iAnywhere. Certains métiers peuvent en tirer profit : les commerciaux itinérants notamment. Nous avons développé beaucoup de solutions métiers, comme dans la santé. On privilégie la fluidité des données.

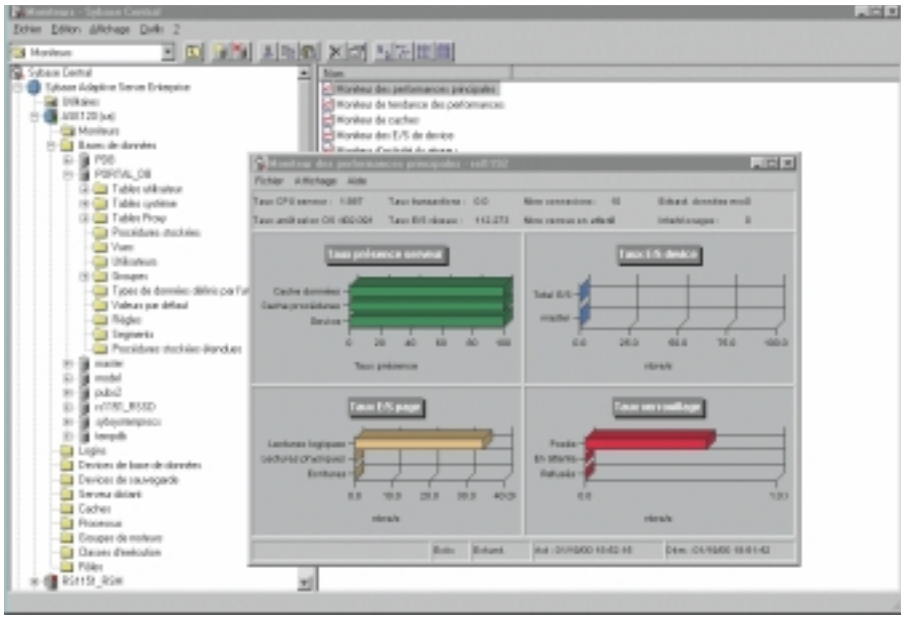
Programmez ! : quelles sont pour vous les prochaines évolutions des SGBD ? Regardez-vous la virtualisation et fédération de données comme IBM ?

BA : Aujourd'hui, j'en vois deux : XML et Web Services. XML est un excellent format d'échange. On passe maintenant d'application à application, basé sur des messages. XML est incontournable dans la fluidité des données. Il est cœur du moteur. Les Web Services sont une façon plus ouverte de communiquer et d'échanger dans un monde hétérogène, avec un standard et à moindre coût de développement. Sur la virtualisation et fédération, ce sont des thèmes pris en compte dans la future version 15 de notre SGBD. Nous voulons une vision unique de l'information, accessible de façon multiple.

Programmez ! : Les éditeurs parlent souvent de XML comme format de stockage. Est-ce une bonne idée ? Qu'apporte de plus un tel format ? Allez-vous intégrer l'objet comme d'autres le font ?

BA : Je dirais qu'il s'agit de mettre un document XML en stockage. C'est une possibilité supplémentaire pour le développeur afin d'obtenir une cohérence d'information. Cette manipulation est tout de même plus lente. Car quand on stocke un document XML, on doit le décomposer pour pouvoir y chercher les informations nécessaires. Nous sommes "relationnel". Gérer et manipuler des objets, est une démarche que nous n'avons pas. Cela reste des extensions du moteur relationnel.

BA : Clairement, ces bases ne sont pas réellement concurrentes de Sybase. Elles peuvent servir de prototypage. Si des clients les utilisent, ce ne sera pas pour des applications critiques. Les clients ont besoin d'être confortés et rassurés. Ces bases se rencontrent dans des applications non critiques, notamment dans les sites Web.



Programmez ! : On reproche parfois aux SGBD d'être de plus en plus lourdes, en ajoutant sans cesse des fonctions. Comment Sybase résout ce problème ?

BA : On ajoute des modules (fonctionnels) dans notre base. C'est très modulaire. En cas de besoin, on active le module adéquat. Si des modules ne sont pas utilisés, ils n'ont aucun impact. On augmente les fonctions internes pour être plus efficace sur l'administration et les performances.

Programmez ! : Les bases libres font beaucoup parler d'elles. Comment Sybase se positionne-t-il par rapport à ces solutions ?

Programmez ! : Le DBA et le développeur vont-ils se fondre dans leur fonction, ou au contraire se spécialiser toujours plus ?

BA : Je ne vois pas forcément une concurrence entre le DBA et le développeur. Certains projets nécessitent peut-être des développeurs plus spécialisés. Il n'y a plus de séparation entre la base et le développement. Cependant, les jeunes diplômés maîtrisent mal le SGBD et pourtant la base est incontournable, dans les équipes de développement, on fait les deux, le codage et la base. Personnellement, je trouve que c'est une faiblesse du cursus.

■ *Propos recueillis par* **François Tonic**

Dossier SGBD : 2eme partie dans le prochain numéro

Oracle mise sur les petites entreprises et la haute disponibilité

Ouvertement tancé par IBM et son DB2, Oracle doit aussi subir les assauts annoncés des SGBD libres. Souvent dénoncé pour des tarifs très élevés, aujourd'hui, Oracle mise sur plusieurs marchés très porteurs. Mise au point avec Pascal Rawsin, responsable marketing d'Oracle Database.

Programmez ! : De quelle manière Oracle perçoit-il le marché du SGBD ?

Pascal Rawsin : On distingue trois niveaux. En haut, les solutions de type grid (computing). Beaucoup de grands comptes sont dans un contexte de réorganisation de projets afin de faciliter la maîtrise de l'architecture réseau, avec la possibilité de réaliser une répartition de ressources. Au milieu, il s'agit plutôt des PME/PMI dynamiques. Elles n'hésitent pas à mettre du cluster sous Linux. Elles considèrent ce dernier comme mature. Et elles font de la haute disponibilité. Ensuite, nous avons les "petits clients". Ils n'ont pas une informatique toujours maîtrisée en interne. Ils passent par des prestataires. Oracle fait un effort sur ce marché avec la disponibilité de la SE1. Pour nous, c'est un marché nouveau. Enfin, je rajouterais un 4e niveau, avec les projets touchant au business intelligence et au reporting, OLAP.

Programmez ! : On parle beaucoup des SGBD libres, comme MySQL. Sont-ils devenus de vrais concurrents ? Oracle garde-t-il une longueur d'avance ?

PR : c'est une réalité. Dans les grands comptes, il n'y a pas de grands déploiements sur ce type d'outils. Une société a besoin de garanties. La base de données est au cœur de l'entreprise. La communauté a fait évoluer MySQL, mais nous avons encore une marge technologique. On a fait évoluer notre support XML, le mode cluster sur plusieurs systèmes, etc. Nous avons franchi le pas Linux depuis plusieurs années. MySQL Cluster correspond à ce que l'on avait dans les précédentes versions d'Oracle.

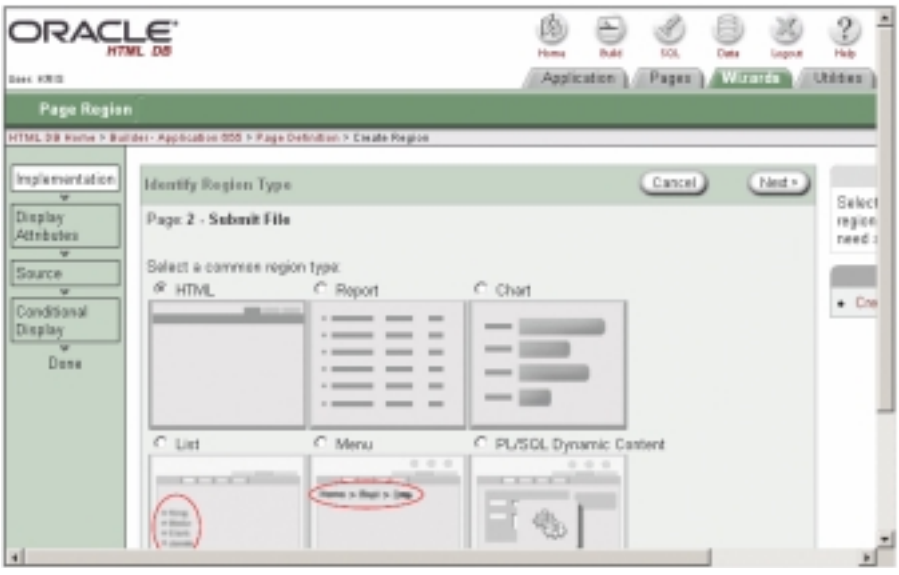
Programmez ! : Une des critiques que l'on fait habituellement aux SGBD concerne l'embonpoint constant de la base et des fonctions. Est-ce un réel problème ? Comment Oracle résout-il le problème ?

PR : Nous continuons à faire évoluer les fonctions. Le système s'est banalisé. Aujourd'hui, on ne regarde plus uniquement les performances du moteur de données, on veut d'autres fonctions. Oracle a fait de gros efforts sur la lourdeur de l'outil. Avec la 10g, le moteur est plus compact, on a aussi revu la taille à l'installation. On automatise les fonctions rébarbatives de la base. Le DBA n'a plus à s'en occuper. Il dispose d'un spectre fonctionnel plus large. Il se rapproche du développeur grâce à des outils de supervision, de performance et de développement.

Programmez ! : un débat secoue toujours le monde informatique pour savoir si XML doit être uniquement un format d'échange ou aussi un format de stockage des données. Qu'est ce que cela vous inspire ?

PR : Il y a 5-6 ans, on avait mis XML comme format de stockage dans la base. Depuis, il est devenu le format d'échange que tout le monde utilise. Dans la 10g, nous avons développé un vrai format de stockage XML que l'on peut utiliser. Le débat n'existe plus.

■ *Propos recueillis par* **François Tonic**



✔ l'actu en ligne ✔ la newsletter
www.programmez.com
✔ le forum ✔ le téléchargement ✔ les archives du magazine
Nouveau : abonnez-vous en ligne !

enquête

1e partie

Métiers du développement : *Les compétences les plus demandées*

Avec ce numéro débute une enquête en plusieurs volets sur les nouvelles tendances dans les métiers du développement. Développer semble désormais quasiment un pré-requis, et la différence se fait sur d'autres aspects : connaissance d'un métier, expertise technique, méthodologie, capacité à manager. Développer une expertise technique ou une double, voire une triple compétence est désormais nécessaire pour intéresser les employeurs.

Sur Cadremploi.fr, 23 % des internautes ayant une expérience dans la fonction informatique ont exercé une fonction de chef de projet et 19 % une fonction d'analyste-programmeur. Les statistiques ci-dessous montrent que les offres d'emploi sont largement supérieures à la demande sur la partie conseil et audit (20 % des offres, 9 % des demandes d'emploi dans la fonction informatique) et surtout des bases de données (26 % des offres, 3 % des demandes). Le ratio s'inverse pour les postes d'analyste-programmeur et de développeur : 3% des offres et 19% des demandes, ainsi que d'ingénieur d'études : 8 % des offres et 14 % des demandes.

Ces chiffres illustrent, si besoin était malgré ce que nous vous signalons chaque mois, la nécessité d'avoir aujourd'hui un profil nettement plus polyvalent que celui de simple développeur

Java et .net en pointe...

Quelles compétences techniques sont les plus recherchées ? Lorsqu'on interroge les SSII, principaux employeurs de profils de développeurs, chef de projets et autres techniciens un peu pointus à ce sujet, elles restent le plus souvent dans le flou : elles ne recherchent pas tant un développeur PHP ou un expert dans telle ou telle techno, soulignant qu'il est toujours possible d'assurer une formation technique.

Pour autant, lorsque l'on regarde sur des baromètres comme ceux de hitechpros ou de cadremploi .fr le décalage entre l'offre et la demande de compétence apparaît. Claude Curs, P-dg d'Hitechpros, qui dispose notamment d'une bourse d'inter-contrats entre SSII, relève une demande soutenue pour tout ce qui touche aux nouvelles technos (Java, J2EE notamment, ASP, Websphere), aux ERP (SAP Toujours, et dans une moindre mesure PeopleSoft et Oracle) et Base de données/décisionnel (Oracle, SQL Server). A contrario, la demande pour les mainframes s'effondre et les cobolistes devraient se hâter de se recycler. Un avis que ne partage pas Marie-Aude Firmin, consultante en recrutement chez Aedian, SSII spécialisée dans le tertiaire financier, secteur où le mainframe est encore très présent pour traiter des volumétries importantes. Cette SSII recrute donc encore des cobolistes, même s'ils ne sont pas majoritaires.

Aston Education, filiale d'Aston dédiée à la formation, observe une très forte demande sur Java : le chiffre d'affaires des formations Java sur le premier semestre a été supérieur de 22% à celui de l'ensemble de l'année 2003, indique Laurence Carlinet, directrice Aston Education. Concernant les formations Microsoft, elle indique un doublement du chiffre d'affaires sur XP + Windows Server 2003, et autant sur MS dev.

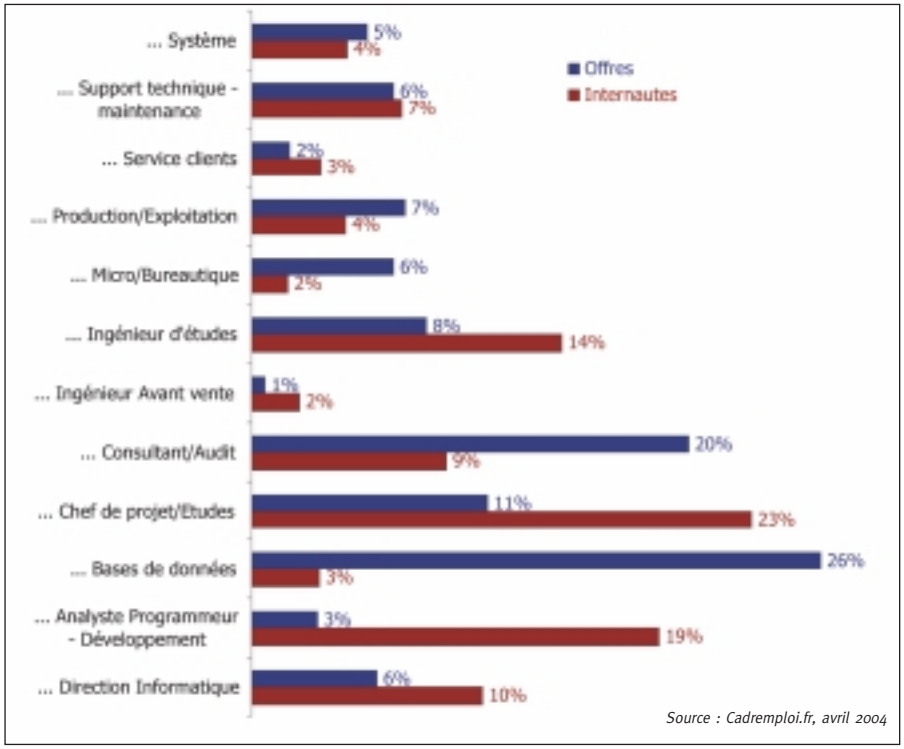
Sur les sites d'offres d'emploi comme monster, cadremploi.fr ou encore celui de l'Apec, les compétences les plus demandées demeurent Java, J2EE et .net.

Des tendances technologiques que confirme Pierre Dellis, délégué général de Syntec Informatique : une demande pour le mainframe qui a nettement diminué, diminution qui touche également le Client/serveur classique et une demande forte pour toutes les nouvelles technologies, les nouveaux langages, tout ce qui touche au réseau. Il souligne également la percée des profils Open Source.

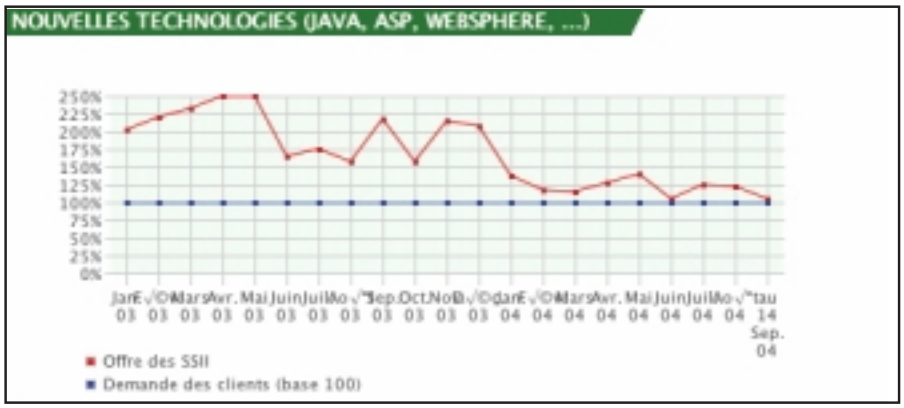
...Open Source en embuscade

Auparavant cantonnées dans des SSII spécialisées, les SSLL, elles sont aussi désormais largement recherchées par les " grandes ". Un besoin qui selon Pierre Dellis ne peut que se renforcer, du fait de la forte demande des administrations et du service public en général

pour ces technologies. Syntec Informatique prévoit d'ailleurs que cette année, comme les deux années précédentes, le secteur public sera l'un des (rares) moteurs de croissance du marché des logiciels & services en France.
Eric Hubert, directeur général de Homsys Group et fondateur de la filiale Objet Direct indique que " Dans le domaine de l'infrastructure, les outils Apache / Tomcat, et bien sûr Linux sont très répandus, ainsi que les serveurs d'application JBoss et Jonas, et le SGBD MySQL, désormais largement utilisés dans les phases de développement et de déploiement. Parmi les outils développement, on trouve notamment les CVS et l'environnement Eclipse, soutenu par IBM, aujourd'hui très utilisé, dans le monde JAVA/J2EE. Il s'impose grâce à son ouverture, qui permet le développement de nombreux plug-in (outils complémentaires) disponibles en open source."



Répartition de la fonction Informatique par sous-fonctions



Soigner la méthodologie

Coté technique, les aspects méthodologie et qualité prennent de plus en plus d'importance " UML devient indispensable, aujourd'hui c'est devenu un véritable standard", souligne Eric Hubert. Chez Homsys, les nouveaux recrutés suivent une formation systématique de cinq jours sur UML.
Au-delà d'UML, toutes les sociétés, SSII comme client final bien sûr, ont renforcé l'approche qualitative des projets informatiques, et donc l'aspect méthodologique. Les démarches CMM se multiplient, parallèlement au développement de centres de production informatique, sortes d'usines à développement, qui ne sont plus basés chez les clients. Cette nouvelle organisation du travail ne signifie pas à proprement parler l'émergence de nouveaux métiers, mais un renforcement de certains aspects, comme tout ce qui touche aux tests. Selon Syntec Informatique, cette approche qualité manque encore dans les formations initiales, où cela demeure très théorique et ne donne lieu qu'à quelques heures de cours, et ne s'acquiert que par l'expérience.

Compétences métier et comportementales

Voici pour les aspects purement techniques. Lors d'une récente réunion de Syntec Informatique, Alain Donzeaud citait comme caractéristique de ces métiers " le développement des compétences techniques et comportementales ". Or s'il paraît relativement aisé d'apprendre un nouveau langage, le sens du service ou la capacité à travailler en équipe est plus difficile à transmettre.
Par exemple, SQLI ne recherche pas des spécialistes d'une techno particulière, mais des "potentiels", capables de progresser, et dont les valeurs rejoignent celles de l'entreprise. Yahya El Mir explique être particulièrement attentif à ce dernier point, ayant constaté que la plupart des échecs de recrutement trouvaient là leur raison d'être. Être passionné et avoir une forte capacité d'adaptation d'une part, une réelle curiosité et ouverture d'esprit d'autre part sont donc des pré-requis., Didier Neyrat directeur général de Cadextan, SSII spécialisée dans le métier de la finance souligne également que " le potentiel et la personnalité son plus importants que les compétences immédiates. ".
Un constat que confirme Jacques Benamor, DRH de CSC, qui souligne que " ce qui fera la richesse d'un profil est la capacité à avoir une vision d'ensemble du système d'information du client ".

La double compétence devient incontournable

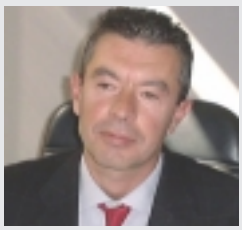
Outre ses compétences techniques, un informaticien, au sens large, doit désormais connaître un métier particulier (la banque est très recherchée) ou une fonction de l'entreprise, par exemple le contrôle de gestion. Une demande des clients, les entreprises utilisatrices, qui s'est accrue avec la période de crise

2001/2004. Selon Pierre Dellis, Syntec informatique, le défaut actuel des ingénieurs est de "ne plus mettre le pied dans une usine", en faisant dès leurs études l'essentiel de leurs stages au sein de la DSI. Découvrir le terrain et la réalité du métier de l'entreprise, ainsi que ses différents rouages que sont par exemple la direction commerciale ou financière, manquerait quelque peu aux ingénieurs actuels...
Certaines SSII se plaignent que les clients demandent des " moutons à cinq pattes ". D'autres, positionnées sur un ou deux métiers, ou un ou deux domaines fonctionnels, s'en félicitent plutôt. " Nous sommes convaincus depuis douze ans (date de création de Business & Décision, ndlr) qu'avoir une ouverture d'esprit et un intérêt pour le métier de ses clients quand on est informaticien est primordial ", insiste Patrick Bensabat, P-Dg de Business & Décision. Selon lui, la double compétence informatique/métier ou informatique/ fonctionnelle tend à devenir triple compétence informatique/métiers/fonctionnelle : par exemple, sur la fonction gestion des risques, ou contrôle de gestion dans le secteur bancaire. Chez Cadextan, Didier Neyrat, rapporte des demandes très précises sur les métiers de gestion d'actif (Asset management), par exemple, outre la connaissance générale du métier de la finance, celle nettement plus pointue des Swap ou des contraintes liées aux projets Bâle II. Parmi les secteurs les plus demandés, la banque, qui domine assez nettement, mais aussi les services en général, le domaine de

"Il y un vrai problème de migration des compétences"

Interview d'Eric Hubert, D.G. de Homsys Group

Quels profils d'ingénieurs en développement recrutez-vous actuellement ?



Eric Hubert, D.G. de Homsys Group et fondateur de la filiale Object Direct

Nous recrutons actuellement plusieurs types de profils : En premier lieu, des ingénieurs expérimentés sur les technologies JAVA et J2EE, connaissant les serveurs d'application, les technologies Web (client léger) les bases de données, et ayant une pratique d'UML dans le cadre de la conception. La demande dans ces technologies continuera à augmenter durant les prochaines années, étant donné la généralisation des architectures Web 3 Tier, et la domination, au moins pour l'instant, de l'architecture J2EE.
Nous recrutons également, avec la montée en puissance de la plate-forme Microsoft .Net, des ingénieurs ayant une bonne connaissance de C# .Net, et une véritable pratique de la conception et du développement objet, dont UML. Ce dernier point est très important, car on sous-estime généralement la difficulté du passage des environnements Microsoft classiques (VB...) au nouvel environnement.Net C# qui est réellement orienté objet, et exige une expérience dans ce domaine. En fait, il est plus facile de passer à C# .net quand on vient du monde JAVA/UML, d'où beaucoup de désillusions aujourd'hui. Nous recherchons également des compétences sur C++ et PHP, mais en complément d'une expérience sur un des deux domaines précédents. Pour ces profils, nous attendons une expérience minimale de 3 à 4 ans, avec une formation Bac + 5 spécialisée en informatique. Nous recherchons également des profils plus expérimentés, à même d'intervenir sur des missions d'architecture technique (J2EE ou .Net), de conseil en méthodologie (mise en place de processus UML/UP), de chef de projet maîtrise d'oeuvre, ou encore d'urbanisation du système d'information. Pour ces profils, l'expérience minimale est de 6 à 8 ans, et doit être diversifiée sur les différentes architectures, avec une dominante technique.

Les compétences recherchées par Homsys

• Compétences principales :
- Java
- J2EE/Web
- .net(C#)
- UML/UP
- SGBDR

• Compétences complémentaires :
- Outillage de conception et développement
- C++
- PHP
- Compétences Open Source : Outils d'EAI etc.

Quels sont les profils que vous avez du mal à recruter ?

Nous manquons de compétences confirmées sur ce type de techno, sur toute la chaîne, de l'aspect interface homme/machine, le web jusqu'à l'expertise sur les bases de données et serveur d'applications. Sur. Net, s'agissant d'une plate-forme récente, les compétences confirmées manquent également. Beaucoup de gens connaissent les plates-formes classiques comme VB ou ASP, mais l'abord de cette nouvelle plate-forme, en particulier avec C#, demande une compétence dans l'approche objet, très différente. Il y a un vrai problème de migration des compétences : nous recherchons des gens à la fois expérimentés sur les technologies Microsoft en général et ayant une bonne pratique de l'approche objet

Recherchez vous des profils double compétence ?

Notre métier est davantage technique que fonctionnel, toutefois, nous recrutons spécifiquement des ingénieurs ayant une expérience télécom ou banque/finance par exemple, en plus d'une bonne connaissance des architectures techniques.

Avez-vous identifié de nouveaux métiers ou fonctions dans le domaine du développement ? lesquels ?

Chaque nouvelle technologie suscite naturellement l'apparition de nouvelles spécialités techniques. Au-delà, on peut observer par exemple, que le développement des approches à base de composants (J2EE/EJB), entraîne une séparation plus nette entre les activités de développement et d'intégration, d'où l'apparition de spécialistes sur les domaines de l'intégration et du test. La montée en charge très rapide des applications Web grand public entraîne aussi des spécialités dans le domaine du test de charge. La généralisation de l'utilisation de UML/UP provoque l'apparition d'architectes méthodes, responsable de la définition et du déploiement du processus méthodologique.

la santé, assez spécifique, ou encore celui de l'aéronautique, des télécoms, de la chimie et de l'énergie. Liste non exhaustive bien sûr...
La nécessité de connaître le métier de l'utilisateur va également de pair avec l'adoption des progiciels de gestion intégrée ; les aspects de paramétrage supposent davantage de connaissance métiers que de connaissances informatiques, si ce n'est bien sûr sur les progiciels en question.

L'informatique... en plus

Point n'est besoin d'être diplômé en informatique pour commencer dans le développement... ce serait presque le contraire. Pierre Dellis Souligne le succès des Mastères qui forment à l'informatique en un an, des diplômés en mécanique, chimie, biologie ou encore finance (ex-DESS double compétences). Les profils issus de cette filière sont largement aussi recherchés que des ingénieurs. "Ils ont un profil rêvé, et une efficacité très supérieure à celle des formations classiques ", précise-t-il. Business & Décision pratique le recrutement de non informaticiens... plus ou moins important selon les périodes. Parmi les profils recrutés durant les années 1998/2000 : ceux de statisticiens, d'économistes. "On a formé une excellente génération de consultants", souligne Patrick Bensabat. Durant la période de crise ces trois dernières années, les clients demandant des profils plus confirmés, la société a perdu un peu de latitude en la matière, les clients refusant le temps nécessaire à la formation de ces non professionnels de l'informatique. Depuis un an, la société explique être revenue à une logique de recrutement plus variée : titulaires de Miages ou de DESS, diplômés d'écoles d'ingénieur, mais également titulaires du DECF ou DESCF pour des projets de mise en œuvre d'outils de consolidation.

Pierre Dellis

Préparer sa reconversion

Selon Pierre Dellis " l'informatique, c'est très bien d'y débuter, mais il faut se poser la question d'y rester au bout de dix ans ". À l'instar d'un sportif de haut niveau, le développeur-chef de projet-expert devrait donc préparer sa reconversion. En cause, une capacité d'adaptation qui tend à s'éroder avec l'âge.

Des chefs de projet de plus en plus polyvalents

Tout développeur n'a pas les qualités nécessaires pour des postes d'encadrement. Une approche originale a été adoptée par SQLI dans la cadre de la formation de ses managers : une formation théâtrale, qui doit leur permettre de " créer des liens et de développer des techniques de communication différentes ". " On ne fabrique pas des managers, on ne peut qu'accélérer le processus de maturation ", souligne Yayha El Mir, président du directoire de SQLI. Pour les postes de chefs de projets, les qualités déterminantes sont la maturité relationnelle et le savoir - être. " Une envie ou une prédisposition " selon Yahya El Mir, qui souligne que " certains développeurs ne deviendront jamais chefs de projets ".
La plupart des SSII forment leurs chefs de projet aux aspects d'encadrement et d'animation d'équipe. Une des tendances actuelles est de les former aussi aux aspects financiers et juridiques des projets. Ainsi, CS Communication & système a achevé en juin la formation de 280 personnes au management de contrat : Une formation lourde, quatre semaines, donnant lieu à une certification, et qui a pour but d'aborder les aspects juridiques, contractuels et même de trésorerie.
De son côté, CSC insiste sur la nécessité pour les chefs de projet d'avoir un excellent contact avec les clients, et ce, pour plusieurs raisons, explique Jacques Benamor : être à même d'interpréter un cahier des charges, développer une intimité avec le client et s'imposer comme un véritable " relais d'information ".

Des architectes très demandés

Ce sera un des aspects que nous développerons dans un prochain volet de cette enquête, mais il ressort que les besoins d'architectes capables d'avoir une vision d'ensemble du système d'information et de conseiller les entreprises en la matière sont très importants.
Pour Jacques Benamor, un des profils un peu rares est celui d'un " architecte ayant une vue généraliste et une bonne maîtrise des différentes couches informatiques existant chez le client ". Il souligne que CSC valorise les fonctions techniques dans l'évolution de carrière de ses ingénieurs d'étude. Outre la classique évolution vers des postes d'encadrement, la SSII affirme qu'il est possible en son sein de "réussir sa carrière en étant un pur technicien".

Décisionnel toujours porteur

Compétences bases de données et outils décisionnels font partie des tops de l'année : les classiques Hyperion, Business Objects et SAS sont rejoints par Informatica, Microstrategy ou encore Datastage d'Ascential. Chez Aston, Laurence Carlinet observe une forte demande de formation sur les outils décisionnels, notamment Business Objects, sur le premier semestre. Sur la partie base de données, la forte demande serait à mettre en relation avec, là aussi, l'adoption accrue des progiciels, qui suppose une mise en forme des données alimentant le PGI.

Carole Pitras

Demandez des formations !

Claude Curs

Pierre Dellis souligne que les SSII qui réussissent le mieux actuellement sont celles où la formation est une dimension très forte de leur politique de gestion des ressources humaines. Citons notamment parmi celles-ci Unilog, Sopra, Accenture ou encore Atos Origin et IBM. " La formation a été négligée " observe Pierre Dellis. " c'est dommage, au niveau des hommes comme des sociétés ". Trop de sociétés miseraient encore sur des technologies trop anciennes. À noter parmi les initiatives récentes en la matière, celles de CS Communication et Systèmes qui forme actuellement et durant quatre semaines certains de ses collaborateurs, pour les adapter " à la demande du marché ". Au total a souligné Gérard Jousset, directeur général de CS, ce programme d'adaptation concernera 200 personnes.
Claude Curs indique que Hitechpros a mené une enquête auprès des SSII qui confirmerait que les ingénieurs n'ont pas tendance à demander des formations, alors même que les directions sont à la recherche de collaborateurs proactifs ; Donc, n'hésitez pas, réclamez !

isen : une école multi-sites

L'institut Supérieur d'Électronique et du Numérique (ISEN) a été créé presque en même temps que l'Isep, que Programmez vous a fait découvrir il y a quelques mois. Le fondateur est d'ailleurs le même : Norbert Segard.

Au départ, l'Isen était l'institut supérieur d'électronique du nord. Ce nom a été récemment changé pour prendre en compte les évolutions géographiques de l'école, désormais aussi présente à Toulon et à Brest, et son orientation vers les TIC : L'objectif de l'école est de former " un ingénieur généraliste dans le domaine du traitement de l'information ". Leon Carrez, directeur général de l'Isen, explique que l'un des axes importants de l'enseignement est La formation au management et à la conduite de projet notamment. Le développement de l'Isen s'est fait en plusieurs étapes : Créée en 1956, l'école comptait à la fin des années 80 des promotions de 120 à 130 étudiants. Souhaitant conserver un lien étroit entre enseignant et élève, une augmentation du nombre d'étudiants n'était pas envisagée, précise Léon Carrez. Décision a donc été prise de créer un autre site. Ce fut Toulon, car il y avait déjà des relations entre l'Isen et la DCN concernant la recherche sur l'acoustique sous-marine. Les collectivités locales souhaitaient par ailleurs l'implantation d'une école d'ingénieurs. L'entité toulonnaise a été créée en 1991. La même approche a présidé à l'ouverture d'un site à Brest en 1994. L'école, sur ses trois sites, compte désormais 300 diplômés par an. L'enseignement de base y est identique, seules les options de dernière année diffèrent. Un étudiant qui a commencé à Lille peut donc aller faire sa dernière année à Toulon ou Brest, en fonction de l'option choisie. Des partenariats avec d'autres écoles permettent également une fin de cycle à l'Isep (cf. Programmez numéro 63), à Angers ou à Lyon. Au total, une trentaine d'options sont donc proposées. L'Isen a par ailleurs développé l'accessibilité aux titulaires de DUT et de BTS. Léon Carrez souhaite " diversifier le bassin de candidatures " et des opérations de sensibilisation sont depuis peu menées dans les lycées à cette fin. Côté international, s'il n'est pas obligatoire dans le cadre de l'enseignement, il est possible et même largement répandu, puisque 50 % des élèves font des séjours de longue durée à l'étranger, dans le cadre de leurs stages ou d'échanges avec les universités partenaires de l'école. Un des points forts dans ce domaine, précise Léon Carrez, est le lien étroit avec les universités canadiennes, et notamment québécoises.

Un " campus numérique "

Dans le cadre de la formation initiale, une des récentes innovations de l'école est la mise en place de son " campus numérique ", qui utilise largement le e-learning, et ce, depuis septembre 2003. Chaque étudiant, quel que soit son site de rattachement, a ainsi accès à une même base d'enseignement et de groupe de travail. Les outils mis en place permettent d'accéder à différentes applications (logiciels de simulation, langages de développement). L'objectif est de permettre aux élèves de disposer, via intranet de présentations de cours et d'exercices, et de " développer les échanges interactifs au travers de logiciels d'évaluation des connaissances ou de mise en place de problèmes ". Un autre volet de cette initiative est d'équiper tous les étudiants de portables, fonctionnant à la fois sous Windows et Linux.

Plusieurs options informatiques

L'informatique à l'Isen fait partie de l'enseignement commun des deux premières années du cycle ingénieur. En première année, l'enseignement est scientifique et technique (données, logiques, algorithme, données complexes, architecture élémentaire), complété par un TP de 32 heures (programmation en C) et un projet informatique de 72 heures, qui vise à réaliser une application complète, fiable et documentée ; en deuxième année, sont abordés les systèmes d'exploitation, les réseaux, les bases de données, ainsi que la théorie des langages et de la compilation, Unix, la cryptographie.

En troisième année, trois options sont dédiées à l'informatique, une sur chaque site : Informatique à Lille (aspects techniques et organisation des SI), télécommunication, informatique et réseaux à Brest, et enfin informatique temps réel à Toulon.

Parallèlement, l'informatique fait partie des axes de recherches de l'école, via son campus numérique, dans les domaines des logiciels libres, de la sécurisation, des certificats électroniques. Des recherches sont également menées dans le domaine de l'intelligence artificielle. L'école propose aussi sur son site de Toulon une formation en alternance débouchant sur un diplôme d'ingénieur ITII, spécialité électronique et informatique industrielle. Une formation sur trois ans ouverte aux étudiants et aux salariés dans le cadre de la formation continue.

Une enquête a été réalisée en 2003 par l'association des ingénieurs Isen : 868 des 4500 ingénieurs diplômés y ont répondu : l'informatique est le deuxième secteur d'activité dans l'entreprise (33 % des répondants), derrière la R&D. En outre, 23 % d'entre eux travaillent au sein de SSII et 5,7 % dans la conception informatique.

Parmi les partenaires entreprises de l'école on compte de grandes SSII comme Sopra Group, Coframi, Atos Origin, Steria, ainsi qu'IBM et Bull.

■ Carole Pitras

Développer des applications “blindées”

Développer des applications sécurisées demande de la part du développeur un état d’esprit qui n’est pas forcément naturel au départ. On a tendance par commencer, en définissant la logique et les algorithmes corrects, pour ensuite démarrer le développement de l’application proprement dit. On s’intéresse avant tout à la conformité des fonctionnalités. Une fois que le développement est effectué et qu’on a corrigé les principaux défauts, on cherche à optimiser la performance du code et l’ergonomie pour les utilisateurs.

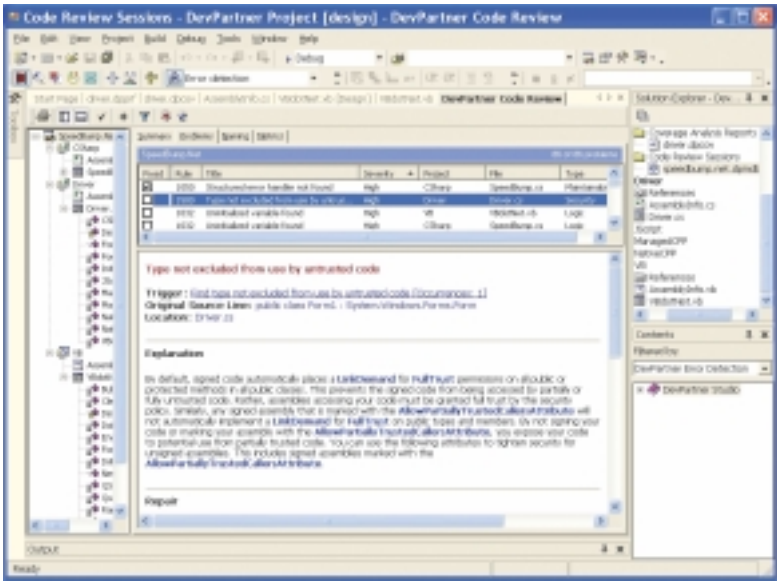


Figure 1. L'identification automatique des failles de sécurité dans le code source aide les développeurs à renforcer la sécurité des applications.

Enfin seulement, on s’intéresse à la sécurité du code. Malheureusement, arrivé à ce stade, on ne peut plus faire grand-chose pour l’améliorer. On a développé un état d’esprit qui ne nous encourage pas à identifier les faiblesses du code. Les développeurs veulent que leur application fonctionne et inconsciemment, se refusent à creuser les parties plus difficiles et obscures du problème.

La sécurité, après tout, n’excite pas tant les foules qu’une belle interface utilisateur, ou une fonction à laquelle personne n’avait pensé jusque-là. " Mieux vaut se concentrer sur ce qui se voit, puisque c’est avec ça qu’on va impressionner le client " résume sans doute cet état d’esprit.

Quand on s’intéresse alors à la sécurité, on applique typiquement la maxime " De la sécurité dans l’obscurité ", en ne résolvant pas le problème au fond, mais plutôt en le dissimulant dans des parties du code inattendues, ou en l’entourant de code abscons. Ce n’est pas intentionnel, mais c’est souvent ce que nous avons de mieux à offrir pour la sécurité du code. C’est ce qui fait que les développeurs continuent de piétiner sur le sujet. C’est un piège dans lequel il est facile de tomber, parce qu’on a du mal à imaginer que quelqu’un puisse prendre le temps qu’il faut pour pirater une application. Tous les développeurs ont des vies bien remplies, avec des échéances professionnelles, des activités et loisirs personnels et ils ont besoin de temps pour augmenter leur savoir et développer leurs compétences. Il est difficile d’imaginer que quelqu’un ferait l’effort fastidieux de passer des jours et des jours à tenter de trouver des failles dans une application, puis à écrire un code pour les exploiter.

Mais il ne faut pas nier qu’il existe des pirates qui ont du talent, de l’obstination et de la patience, disposant des équipements et des informations qu’il faut pour casser une application, motivés par le vol, ou n’ayant d’autre raison que de démontrer leur savoir-faire. Qu’ils s’intéressent ou non à votre application n’est qu’une affaire de temps et de chance, mais il est probable que cela arrivera, un jour ou l’autre. La seule défense est de leur rendre la tâche aussi difficile que possible. Si c’est facile, les pirates peuvent causer des dommages importants à votre application et, vraisemblablement à l’entreprise qui l’utilise. Si par contre ils se cassent les dents sur votre application, ils s’intéresseront à d’autres, sur d’autres serveurs, plus faciles à pirater. S’ ils persistent, cela leur prendra si longtemps que leurs tentatives de piratage finiront par être détectées par ceux qui surveillent l’application en production. Rendre une application invulnérable est impossible, quoi que vous fassiez. L’application invulnérable n’existe pas. Même si l’application elle-même est sûre, elle doit s’exécuter dans un environnement donné, constitué d’un système d’exploitation, du matériel, des réseaux et surtout, impliquant des utilisateurs. Les pirates les plus déterminés et talentueux peuvent pirater n’importe quelle application de gestion, simplement parce cette application est vivante et rend un service réel à ses utilisateurs. Une autre considération à prendre en compte est que la sécurité d’une application est inversement proportionnelle à son utilité. La seule application sûre est celle qui tourne sur un ordinateur installé dans un coffre fort et ne disposant d’aucune entrée sortie. Néanmoins, une telle application n’est pas très utile. L’application la plus simple à utiliser est celle qui ne nécessite aucune authentification et fournit à l’utilisateur un accès illimité aux informations qu’elle gère. Cette application, potentiellement très utile et simple à utiliser, est ouverte à toutes les tentatives d’intrusion. Le développeur de l’application doit donc trouver un juste milieu, mettant en oeuvre certaines restrictions d’accès à l’application, tout en la rendant simple et facile d’emploi. Cet équilibre doit prendre en compte la probabilité de la menace, le poids des conséquences liées à une intrusion, et le sérieux et l’aisance de l’utilisateur avec l’informatique.

La définition des frontières et le travail nécessaire pour assurer cet équilibre dépendent largement de la conception de l'application. Créer une application qui gère les différents niveaux de sécurité est plus difficile. Le principal problème est qu'un code correct n'est pas synonyme de code sûr. Obtenir un code sûr exige un effort supplémentaire, une connaissance détaillée des vulnérabilités de Windows, et de savoir coder pour se protéger de celles-ci. Malheureusement, la grande majorité des développeurs n'a pas les connaissances requises pour identifier précisément et corriger ces failles.

Idéalement, il serait possible d'identifier et de gérer automatiquement ces vulnérabilités, qui sont connues, mais souvent absconses et difficiles à détecter et corriger. La solution idoine "silver bullet" pour le faire n'existe pas encore, mais il est possible de faciliter le processus lors de l'écriture et de l'analyse du code.

La sécurité peut être vérifiée au cours de différentes phases du processus de développement. La plus évidente et la plus importante est sans doute celle de l'écriture du code. Il y a deux raisons à cela : la première est que c'est l'étape la moins coûteuse pour s'assurer de la sécurité de l'application, la deuxième est que c'est durant celle-ci que le développeur est plus apte à identifier et corriger les défauts de sécurité.

Ecrire du code en pensant à la sécurité est un véritable défi. Le problème est que de nombreuses constructions ou algorithmes ont des vulnérabilités, et peuvent être utilisés de telle façon que ces failles apparaissent ou pas. Un code parfaitement correct et acceptable peut subir une intrusion dans des circonstances données. Les développeurs ne sont pas toujours conscients des nuances et effets de bords que créent leurs constructions.

Une première difficulté consiste à trouver les "back doors" ou portes détournées qui sont rendues possibles par le langage et la plate-forme. Par exemple, si vous déclarez FileIOPermissionAttribute comme ceci :

FileIOPermissionAttribute(SecurityAction.Deny, Read=@"c:\passwords")

Vous allez bloquer l'accès au répertoire des mots de passe, seulement si on y accède de cette façon. Vous pouvez continuer à utiliser le chemin UNC (\\machine_name\passwords), ou d'autres chemins vers le même répertoire, à moins de pouvoir identifier et protéger toutes les autres façons d'y accéder.

Il est difficile de se rappeler de tout cela, ainsi que des nombreuses autres règles, tout en sachant quand les appliquer. Une façon de surmonter cet obstacle est d'utiliser la fonctionnalité d'analyse de code de Compuware DevPartner Studio, qui scanne le code Visual Basic et .Net pour détecter deux douzaines d'erreurs typiques concernant la sécurité de code. Vous pouvez également définir vos propres modèles de recherche de failles si vous pouvez les définir dans une expression régulière. La Figure 1 montre une erreur de sécurité particulièrement difficile qui intervient quand on utilise un code signé dans une application .Net. Par défaut, les assemblages signés donnent une permission " FullTrust " aux méthodes publiques et privées des classes publiques, de façon à ce qu'ils ne puissent être exécutés que par du code auquel on a explicitement donné les permissions. En omettant de signer votre code, ou en marquant votre assemblage avec l'attribut AllowPartially TrustedCallers, vous exposez votre code à être utilisé par du code non autorisé. Cela laisse donc la porte ouverte au détournement de votre code par un code pirate inséré ou en provenance d'un autre programme. DevPartner Studio identifie les occurrences de ce type de problème (ainsi que d'autres failles de sécurité) avec son automate de revue de code, ce qui épargne aux développeurs et aux équipes de développement le temps et l'énergie qu'il faut pour appliquer ces règles manuellement.

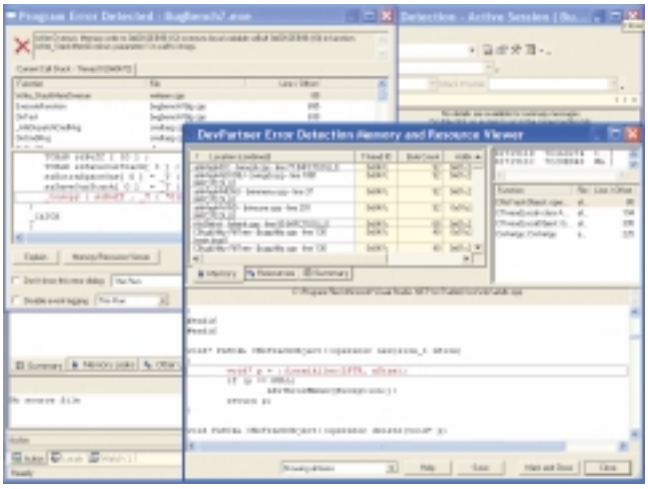


Figure 2. La détection des failles de sécurité à l'exécution permet aux développeurs de les corriger alors qu'ils sont encore dans la phase de codage.

Analyse de la sécurité à l'exécution

Une fois le code écrit, et l'application développée, il faut trouver des failles de sécurité à l'exécution. Le problème est toutefois mieux compris. Ainsi, la plate-forme .NET offre une approche de sécurité de type bac à sable ("sandbox"), qui fait que les problèmes relatifs à une application ne se propagent pas à d'autres. Cela ne résout pas entièrement le problème, bien évidemment, mais cela en atténue les effets. Des problèmes plus graves surviennent lorsqu'une application .NET appelle un code natif Windows, via un accès P/Invoke, un "wrapper" COM, ou par un autre moyen.

Le flux de contrôle de l'application quitte alors le bac à sable et devient vulnérable à n'importe quelle corruption du code qui s'exécute hors du framework .NET. Ce type de faiblesse est maintenant assez bien connu, d'autant qu'il a été identifié et exploité par les pirates et les concepteurs de virus depuis des années. Cependant, localiser les failles de ce type dans le code, en particulier quand il a été écrit il y a des années, est fastidieux et génère des erreurs. Les développeurs qui écrivent de nouvelles applications .NET, souvent, ne réalisent pas qu'ils introduisent des failles de sécurité en appelant un code natif existant.

La plus " infâme " de toutes ces failles de sécurité à l'exécution, c'est-à-dire la plus couramment utilisée par les pirates, est sans doute le dépassement de mémoire tampon (buffer overflow). En langage C et C++, il n'y a pas de vérification interne de cohérence qui assure qu'une valeur pourra être stockée dans un espace mémoire donné. C'est au développeur d'écrire ces vérifications dans le code, mais la plupart ne s'en donne pas la peine. N'importe quel utilisateur peut alors, à partir d'une ligne de commande ou d'une page web, surcharger intentionnellement la mémoire et tenter de prendre ainsi le contrôle de l'application. Si c'est une mémoire locale, elle écrase la pile et " plante " l'application ou l'un de ses composants et dans certains cas, cela permet à un code malveillant de s'exécuter une fois la pile effacée.

D'autres failles du code natif Windows sont de même nature, détournant et utilisant la mémoire allouée à l'exécution. On trouve par exemple la surcharge d'un tableau, l'utilisation de mémoire non initialisée ou un pointeur pointant

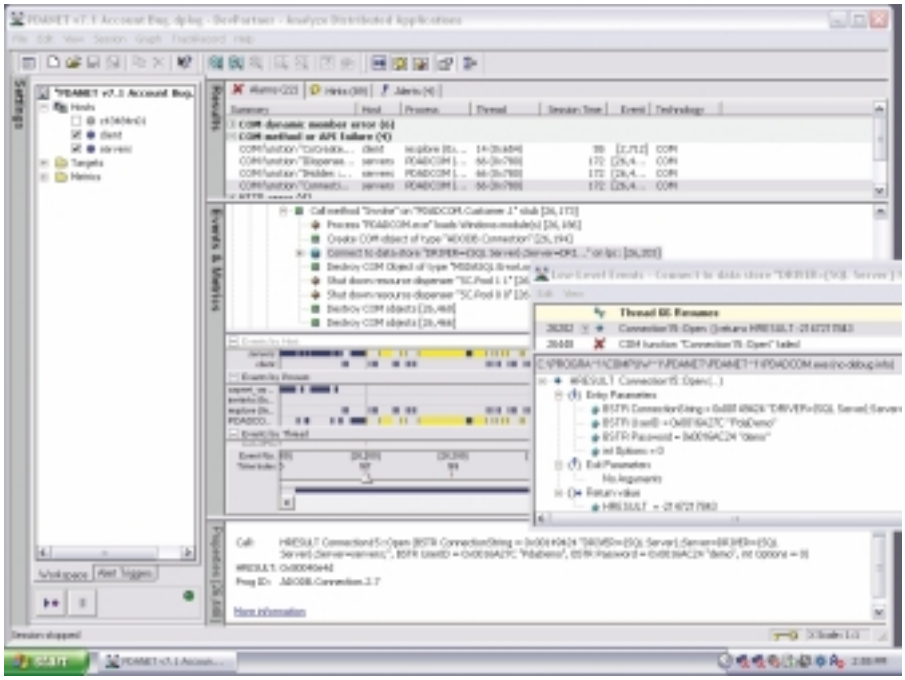


Figure 3. La possibilité de voir le nom d'utilisateur de la base de données et le mot de passe non protégés montre aux développeurs combien il est essentiel de crypter de telles données critiques.

vers une adresse hors limite. Une fuite de mémoire sur une interface COM peut devenir une faille, en permettant à un code pirate de prendre son contrôle et de renvoyer des données corrompues dans l'application .NET appelante. Identifier ces failles implique une revue détaillée de chacune des lignes de code, la création d'une liste des variables déclarées, la quantification de la mémoire allouée, et la mise en place d'un code détecteur d'erreurs placé aux endroits sensibles. Au-delà du temps et des coûts engagés, les éventuelles failles sont difficiles à identifier dans une application qui effectue des millions d'assignations et de désaffectations au cours de sa vie. L'identification des failles de sécurité liées à la gestion de la mémoire peut être automatisée, grâce au système de détection d'erreur automatique de DevPartner. Le simple fait d'exécuter la détection d'erreur permet d'identifier le potentiel de surcharge des mémoires tampons, d'écrasement des valeurs de piles et autres problèmes de ce genre. Par exemple, la figure 2 illustre une erreur d'écrasement de pile, semblable à ce que pourrait accomplir un pirate pour corrompre une application. Bien entendu, comme cela se passe quand l'application s'exécute, vous ne la trouverez que si l'erreur est injectée dans l'application. Cela veut dire qu'il faudrait simuler une attaque pour être efficace et gérer ce genre de faille.

Les données en clair

La troisième phase, où la détection de failles est cruciale, se passe quand on transfère des données entre les composants d'une application distribuée. De nombreux développeurs croient que les données circulant dans l'application sont en sûreté, simplement parce que l'utilisateur n'y a pas accès. Pourtant, un virus ou un programme hostile peut observer l'application et soit déterminer ses failles lors d'une attaque, soit corrompre les données utilisées dans l'application.

Par exemple, un programme hostile peut observer des données critiques en clair et transférées d'un composant à l'autre. Un mot de passe en mode texte simple saisi sur une page web peut être capturé et permet alors un accès au compte de l'utilisateur. Il est également courant que les développeurs incluent le mot de passe de la base de données dans un script plutôt que dans l'application. Certains développeurs utilisent même l'identifiant et mot de passe de l'administrateur système pour exécuter des requêtes vers la base de données par des utilisateurs web occasionnels. N'importe quel programme hostile qui peut capturer ce mot de passe aurait un contrôle total de la base de données en production.

Malheureusement, la configuration d'un accès à la base de données ainsi que le transfert de données entre les différents niveaux d'une application distribuée sont souvent au cœur des décisions que les développeurs doivent prendre lors de la conception détaillée de l'application. Ce niveau de mise en oeuvre n'est pas planifié à l'avance. Ainsi, à moins que les développeurs ne soient à la fois formés sur les aspects de sécurité et vigilants lors de la mise en oeuvre, il est très probable que des informations critiques dans une application soient vulnérables.

Identifier ces éléments vulnérables une fois que l'application est entièrement codée est un véritable défi, parce que vous devez être capable d'observer le transfert de données à l'intérieur de l'application. L'analyseur distribué DevPartner Distributed Analysis permet d'afficher les données en clair qui sont transférées entre les composants de l'application pour faciliter l'identification de failles. Le 3e schéma montre le passage en clair du " login " et " password " d'un utilisateur, d'un composant COM vers une base de données. Ceci est indéniablement une faille de sécurité qui doit être corrigée dans une application en production. (Fig.3)

La partie visible de l'iceberg

Il y a un nombre infini de possibilités d'écrire du code susceptible de contenir des failles de sécurité similaires à celles que je vous ai décrites. À moins de vous spécialiser dans la détection et la correction de failles de sécurité, il est peu probable qu'un seul individu puisse connaître, ne serait-ce qu'un nombre significatif d'entre elles. Cela veut dire que votre application présentera des failles de sécurité. Certaines d'entre elles seront très sérieuses. Si vous ne faites rien, ces failles seront toujours présentes dans l'application en production.

Que votre application ait des failles ne signifie pas nécessairement qu'elle sera attaquée. Vous pouvez toujours vous fier à la providence pour que votre application n'attire pas l'attention des pirates, ou vous dire que votre code est vraiment trop complexe pour qu'un pirate s'y attaque. Mais ces deux défenses ont leurs limites. Pourquoi donc lui faciliteriez-vous le travail ? Automatiser la détection et le diagnostic de failles de sécurité vous permet de compter un peu moins sur votre bonne étoile et vous aider, en passant, à écrire un code de meilleure qualité.

Peter Varhol
Compuware Corporation

Applications Java en cluster avec Tomcat 5

Afin de répondre aux exigences en termes de fiabilité et de disponibilité, de plus en plus fortes, les éditeurs de serveurs d'applications commerciaux ont très rapidement introduit des mécanismes de clustering, permettant d'assurer ces fonctionnalités de haute disponibilité. À côté de cela, les solutions open source, bien que reconnues pour leur qualité d'implémentation, se sont toujours positionnées en retrait et n'avaient donc pas les moyens de se confronter aux serveurs d'applications commerciaux sur ce domaine. Mais les choses sont en train de changer.

Tomcat propose désormais en natif dans sa version 5, des solutions pour répondre aux problématiques de clustering, tout comme JBoss, qui fournit de tels services depuis quelque temps déjà. Contre toute attente, les mécanismes de clustering ne font toujours pas l'objet d'une spécification de la part de Sun, et leur implémentation, reste à la charge des éditeurs et développeurs, ce qui explique souvent les différences de mise en oeuvre que l'on peut retrouver d'un produit à l'autre.

Rappels sur le clustering / les attentes

Le clustering au sein d'un serveur d'application J2EE permet de garantir un niveau de qualité et de continuité de service minimum des applications qu'il exécute.

Cette qualité de service peut se décliner à des niveaux plus ou moins fins, et peut être mise en oeuvre selon plusieurs approches technologiques.

Ainsi, une solution de clustering répond généralement à trois besoins distincts :

- la montée en charge (scalabilité),
- la haute disponibilité (failover),
- la répartition de charge (ou load-balancing).

Les principes techniques

La montée en charge (Scalabilité)

Lorsqu'une application J2EE monte en charge, une des plus importantes contraintes en terme de qualité de service est d'assurer un temps minimum et garanti pour la réponse à chacune des requêtes émises par les utilisateurs.

Ainsi, pour une application bien conçue, si une requête unitaire prend un temps " t " pour s'exécuter, il faut qu'elle soit capable de répondre à un nombre concurrent de requêtes dans un temps toujours très proche de " t ".

Pour conserver ce temps de traitement toujours au plus proche de la valeur plancher unitaire " t " lorsque l'application est soumise à une forte charge, il faut être capable de rajouter des moyens matériels.

Cette notion est appelée scalabilité. La scalabilité peut être abordée sous deux angles :

- la scalabilité verticale, qui consiste à ajouter des ressources sur le serveur (mémoire, CPU, etc.),
- la scalabilité horizontale, en augmentant le nombre de serveurs au sein du cluster.

Dans chacun des deux cas cités ci-dessus, l'applicatif doit être conçu de façon à ce que les gains apportés sur les performances soient toujours linéaires.

Haute disponibilité (failover)

L'objectif de la haute disponibilité est de fournir un moyen permettant de garantir de la façon la plus transparente pour l'utilisateur, la continuité du service si une défaillance venait à survenir (arrêt impromptu, crash disque, etc.) sur un des serveurs qui compose le cluster.

Dans le cas d'un serveur d'applications J2EE, on peut distinguer deux niveaux de " qualité " pour gérer la haute disponibilité en cas d'avarie technique :

- La haute disponibilité au niveau des requêtes (request failover),
- La haute disponibilité au niveau des sessions (session failover).

Request failover

Dans cette configuration, si l'un des serveurs du cluster tombe, les requêtes suivantes sont redirigées sur les serveurs restant actifs dans le cluster. Concrètement, ce mécanisme repose souvent sur un système de " heartbeat " (battement de coeur) qui s'assure périodiquement de l'état de chacun des serveurs du cluster, ce qui évite ainsi d'envoyer des requêtes sur des serveurs qui ne peuvent pas y répondre.

Session failover

En mode session failover, le cluster doit être capable de maintenir les sessions des clients si l'un des serveurs du cluster était amené à tomber. Ainsi, les autres serveurs du cluster récupèrent et garantissent la continuité des sessions utilisateur, avec une perte minimum d'information et de continuité dans le service. De façon plus concrète, cela consiste à répliquer les données de session au travers de l'ensemble des machines qui composent le cluster.

Répartition de charge (load balancing)

L'objectif du load balancing est de permettre la distribution de la charge entre les différentes machines du cluster afin de fournir le meilleur temps de réponse possible aux utilisateurs du site.

Ce mécanisme s'opère au travers d'un algorithme de distribution de charge, soit de type round-robin, soit plus élaboré, qui est chargé de distribuer les requêtes aux serveurs du cluster en gardant trace de la charge et des ressources disponibles sur chacun des ser-

veurs. Le modèle d'implémentation d'une telle solution repose très souvent sur le principe du client/serveur.

Les critères de choix pour la mise en place d'un cluster J2EE

De nombreux éléments sont à prendre en compte lorsque l'on souhaite mettre en place un cluster J2EE. Pour cela, il convient donc de répondre préalablement à de nombreuses questions.

En effet, il faut pouvoir déterminer quels sont les besoins et les attentes que la solution doit apporter.

- Le clustering implémenté doit-il être vertical ? Horizontal ?
- Le clustering doit-il porter sur tous les tiers de l'architecture (SGBD, frontaux Web, serveurs d'application) ?
- Quel algorithme de répartition de charge retenir (round-robin, aléatoire, weight-based, etc.) ?
- De quelle façon sont considérées et traitées les anomalies sur le cluster ?
- Quelle doit être la fréquence de réplication des sessions ?
- etc.

Toutes ces réponses pourront bien évidemment varier en fonction des exigences induites par le projet.

Le clustering avec Tomcat 5

La dernière version de Tomcat fournit des fonctionnalités permettant la mise en cluster et la répartition de charge. Ces apports sont très importants pour Tomcat car ils lui permettent désormais de concurrencer WebLogic et WebSphere sur le terrain du déploiement et de la robustesse des applications Web.

Le load-balancer fourni avec Tomcat 5 s'articule autour d'une application basée sur un système de règles. Par défaut, Tomcat propose deux règles simples de répartition de la charge entrante, basées sur :

- un algorithme round-robin,
- un algorithme de répartition aléatoire

Tomcat n'assurant qu'un rôle de conteneur de servlet (il n'offre pas l'ensemble des fonctionnalités J2EE : EJB, JMS, etc.), il ne peut assurer qu'un clustering autour des servlets et des JSP.

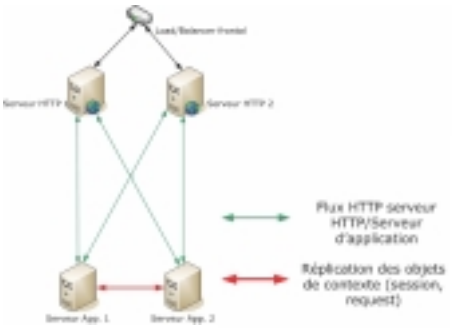
Clustering

Une architecture clusterisée impose à tous les serveurs contenus dans le cluster de devoir communiquer et échanger les informations relatives aux contextes et sessions utilisateurs. Cette communication interprocessus peut être effectuée selon deux modes différents :

- par une communication point à point en RMI (Remote Method Invocation),
- par IP multicast.

La plupart des serveurs d'applications (ATG Dynamo, BEA Weblogic, IBM Websphere, etc.) utilisent le mode basé sur IP Multicast pour communiquer et s'échanger les informations relatives à l'état des différentes instances, les mises à jour des informations de contextes et de sessions, etc.

Le principe retenu pour Tomcat est similaire à celui-ci, il est donc basé sur un modèle déjà bien éprouvé. Tous les membres du cluster s'échangent des données par le biais de messages multicast. Chaque instance de Tomcat envoie un message contenant son adresse IP, et son port d'écoute TCP pour la réplication des sessions. Si une instance n'a pas reçu le message dans un laps de temps donné, le serveur est considéré comme étant en panne.



Les développeurs de Jakarta prévoient, dans une prochaine version, d'introduire la notion de " farming " dans Tomcat. Le " farming " permet le déploiement à chaud d'applications web sur une ferme de serveurs. Dans cette configuration, il suffit de déployer le fichier WAR sur un seul et unique serveur ; ensuite, c'est le mécanisme de farming qui permet de gérer le déploiement des applications parmi l'ensemble des machines du cluster. De la même façon, le fait d'enlever le fichier WAR sur le serveur va avoir pour effet de désinstaller l'application web sur toutes les machines du cluster.

Load Balancing

Dans les précédentes versions de Tomcat, le load balancing n'était pas fourni en standard. Dès lors, la solution la plus répandue pour répondre à ce besoin consistait à intégrer Tomcat avec Apache 2 pour répartir plus facilement la charge. Dans cette configuration, une instance de Tomcat appelée " Tomcat Worker " est configurée pour assurer le load-balancing.

Tomcat 5 fournit désormais trois fonctions de load-balancing de façon native selon 3 modes :

- avec le " mod_proxy " d'Apache en utilisant le connecteur natif JK de Tomcat,
- avec le " mod_rewrite " d'Apache en utilisant le connecteur natif JK de Tomcat,
- avec une webapp nommée " balancer ".

L'application " balancer " est une webapp basée sur un système de règles et qui utilise un mécanisme de " Servlet Filter " pour rediriger les requêtes entrantes sur le premier membre disponible du cluster. L'application "balancer" n'est pas destinée à remplacer les solutions de répartition de charge matérielles, cependant, c'est une solution simple et efficace pour rediriger le trafic parmi un ensemble de serveurs (sans utiliser un boîtier dédié).

Les règles sont centralisées dans un fichier de configuration rules.xml, qui contient les différentes règles et les URL de redirection. Le filtre de l'application " balancer " vérifie l'enchaînement des règles, pour déterminer où il doit effectuer la redirection, en se basant sur les règles contenues dans le fichier, et s'arrête dès que la règle correspond aux critères.

Exemple de fichier rules.xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<rules>
  <!– If the URL contains News (case-sensitive),
go to CNN.com –>
    <rule  className="org.apache.webapp.
balancer.rules.URLStringMatchRule"
    targetString="News"
    redirectUrl="http://www.cnn.com" />

  <!– If the request contains a parameter named
paramName whose value
      is paramValue, go to Yahoo.com. –>
    <rule  className="org.apache.webapp
balancer.rules.RequestParameterRule"
    paramName="paramName"
    paramValue="paramValue"
    redirectUrl="http://www.yahoo.com" />

  <!– Redirect all requests to jakarta.apache.org. –>
    <rule  className="org.apache.webapp
balancer.rules.AcceptEverythingRule"
    redirectUrl="http://jakarta.apache.org" />
</rules>
```

Tolérance aux pannes/Réplication de sessions.

La tolérance aux pannes est la capacité du système à reprendre un traitement ayant échoué, et à la poursuivre sur un des autres serveurs du cluster de la façon la plus transparente possible pour l'utilisateur. Le cas idéal de tolérance aux pannes se produit lorsque le cluster détecte qu'une des instances de serveurs n'est plus disponible et qu'il redirige de façon complètement transparente toutes les requêtes à destination de cette instance vers une instance disponible.

Dans la version 5 de Tomcat, la réplication des sessions s'opère de façon globale. La totalité des données et des objets de chacune des sessions sont répliquées sur tous les membres du cluster, et ce, de façon permanente. Cette solution est efficace dans le cas d'un cluster de petite taille, mais peut vite se révéler pénalisante quand le cluster devient plus conséquent. Pour les clusters importants, la topologie retenue s'oriente généralement autour d'un serveur de réplication primaire, et d'un serveur secondaire. Cette approche devrait être implémentée dans une version ultérieure de Tomcat.

	Scalabilité	Failover (request FO, session FO)	Load Balancing	Farming
Tomcat 5	☑	☑	☑	
ATG Dynamo	☑	☑	☑	
BEA Weblogic	☑	☑	☑	☑
IBM Websphere	☑	☑	☑	

Matrice comparative des fonctions de clustering des serveurs d'application

Les sessions peuvent être répliquées selon 3 modes :

- en utilisant la réplication en mémoire (SimpleTcpCluster),
- en utilisant la persistance de session (JDBCStore), c'est-à-dire en stockant les données dans un SGBD,
- en sauvegardant les sessions sur un système de fichiers partagé (FileStore).

Conclusion

Les problématiques d'architectures induites par la haute disponibilité et la tolérance aux pannes des applications Java sont d'une complexité assez importante. Les alternatives Open Source, bien que très réputées pour leur grande qualité, n'ont jamais, jusqu'à présent, apporté une réponse crédible face aux éditeurs reconnus (ATG, BEA, IBM, etc.) de serveurs d'application J2EE, sur le terrain très sensible du clustering.

Cependant, avec sa version 5, Tomcat n'a plus grand-chose à envier aux autres serveurs d'application sur ce terrain, et est à même de répondre aux besoins les plus fréquents. Les différents modules proposés sont conçus selon des modèles éprouvés et simples à mettre en oeuvre.

■ **Mickael MAINDRON,**
architecte chez Neoxia
mickael.maindron@neoxia.com

Les containers légers basés sur l'Inversion de Contrôle face aux EJB

La richesse technique de la norme EJB n'est pas toujours évidente à mettre en œuvre sur des projets relativement modestes. Un grand nombre de fonctionnalités disponibles de facto à travers les EJB sont souvent délaissées par les équipes, soit par manque de compétences, soit par manque de besoins.

Aujourd'hui, plusieurs solutions logicielles à base de containers légers reposant sur le 'design pattern' d'inversion de contrôle existent et proposent des fonctionnalités réduites, mais dans la plupart des cas suffisantes pour les projets. À travers cet article, nous allons rapidement présenter la notion d'Inversion de Contrôle, puis les principaux containers se basant sur ce principe, à savoir :

- Avalon, avec le projet Merlin développé par la fondation Apache
- Spring qui inclut notamment l'utilisation de la programmation orientée aspect
- Pico qui utilise une approche minimaliste et simpliste

L'Inversion de Control (IoC)

L'Inversion de Contrôle (IoC) est un 'design pattern' qui est utilisé par les containers pour gérer les dépendances entre composants, le cycle de vie et la configuration des composants.

Le terme d'Inversion de Contrôle étant jugé trop ambigu, plusieurs synonymes existent pour définir le même concept : DIP (Dependency Injection Principle, présenté dans l'article de Martin Fowler et 'The Hollywood Principle' (Don't call us we'll call you).

Un composant conçu selon les règles de l'Inversion de Contrôle, ne se charge pas d'aller chercher et de créer les composants nécessaires à sa tâche. Il déclare, à travers des dépendances, les composants dont il a besoin, laissant le container se charger de fournir les éléments requis d'où le nom 'IoC/DIP/ Hollywood Principle'. Le composant est tributaire du container qui va se charger de faire respecter la dépendance en préparant les composants hiérarchiquement.

Le cycle de vie des composants entre également en ligne, dans la mesure où le container se charge également de fournir les composants dans un état donné. La plupart du temps, le cycle de vie va être extrêmement simple (start, stop, dispose), mais pourra le cas échéant être agrémenté d'états intermédiaires.

La configuration des composants est une des facettes de l'IoC, ces derniers nécessitent souvent un paramétrage (issu de fichier de configuration par exemple). Les containers ont pour rôle d'initialiser les composants avec les configurations adéquates.

De par son approche faiblement couplée, l'IoC facilite, de fait, la notion de 'Separation of Concern' (SoC), ou les implémentations sont découplées des interfaces manipulées par les composants. Le container est en charge de fournir des implémentations aux composants, ces derniers manipulant généralement des interfaces.

Différents containers

Avalon Merlin (interface injection)

Le projet Apache Avalon s'est constitué à l'origine pour définir un framework d'Inversion de Contrôle, indépendant de toute implémentation de container. Fort de son framework, la communauté rassemblée autour du projet, a développé plusieurs containers, tels que Phoenix (reconverti dans le projet Loom), Excalibur et plus récemment, Merlin.

L'inversion de contrôle proposé par Avalon Merlin va reposer sur la définition de dépendance sur une interface dans une classe implémentant l'interface 'Servicable'. Une classe va pouvoir déclarer une dépendance à un service en déclarant l'interface requise pour son traitement. Le 'service manager' mis à disposition dans la méthode Service (apporté par l'interface Servicable) fournira l'implémentation de l'interface qui sera définie dans un fichier XML de configuration.

Merlin propose l'utilisation de déclaration de type 'doclet' pour simplifier l'écriture des dépendances à même le code. Les écritures de type @avalon.component définissent un composant, alors que les @avalon.dependency se chargeront de spécifier la dépendance du composant. Ces déclarations seront traduites en fichiers de configuration XML lors de la phase de précompilation du source.

```
/**
 * @avalon.component name="Pere.name" lifestyle="transient"
 */
public class Pere implements Serviceable
{
   IFils fils;
   /**
    * @avalon.dependency type="IFils" key="fils"
    */
    public service( ServiceManager manager ) throws ServiceException
    {
        this.fils = (IFils) manager.lookup( "fils" );
    }
}


/**
 * @avalon.component name="Fils.name" lifestyle="transient"
 * @avalon.service type="IFils"
 */
public class Fils implements IFils
{
          public void sousSurveillance(){
                    System.out.println( "Sous surveillance" );
}
}
```

```
public interface IFils
{
        public void sousSurveillance();
}
```

Ici, l'interface IFils est implémentée par la classe Fils. La classe Pere implémente 'Servicable' et à travers sa méthode 'Service' récupère une instance d'une implémentation de IFils.

```
<component name="Fils.name" class="Fils"/>
```

À travers cette déclaration dans le fichier de configuration XML, la classe d'implémentation Fils est associée l'interface IFils, ce sera donc une instance de Fils qui sera transmise à la classe Pere (conformément à la dépendance définie dans la classe Pere).
La particularité un peu déroutante de Merlin (par rapport aux deux autres containers présentés ci-dessous) réside dans son absence de point d'entrée pour le démarrage de l'application (équivalent du main). À l'exécution de Merlin, celui-ci se charge de lancer tous les services qui doivent être activés au démarrage. Par exemple, si la classe Pere est un point d'entrée du service, il faudra indiquer à travers le fichier de configuration une activation au démarrage comme ceci :

```
<component name="Pere.name" class="Pere" activation="startup"/>
```

En plus de la gestion de la dépendance, chaque classe pourra implémenter des interfaces qui définiront un cycle de vie élaboré. Parmi les interfaces définies dans le framework Avalon, le développeur pourra choisir les états : LogEnabled, Contextualizable, Parameterizable, Servicable, Configurable, Initializable, Startable ou Executable. Le container se chargera d'appeler les méthodes définies dans les interfaces, avant et après consommation du service.
Sur notre classe Pere cela se traduira, par exemple, de la façon suivante :

```
public class Pere implements Serviceable, Initializable
{
    // ...
    public void initialize() throws Exception
    {
        //...
    }
}
```

Merlin va au-delà de sa fonction de container et se présente plus comme un 'micro noyau' (A System of Patterns) capable d'assembler des briques de composants indépendants, en préservant leur intégrité à l'aide de 'classloaders' dédiés. A cette fonctionnalité s'ajoute la possibilité de gérer les dépendances et de télécharger des librairies dans un 'repository' (à l'instar de Maven). À travers le fichier de configuration, une application pourra spécifier les composants, librairies et point d'entrée nécessaire pour l'exécution du programme : Merlin se chargera automatiquement de récupérer et assembler les différentes briques. Ces fonctionnalités s'avèrent extrêmement pratiques lorsqu'il est question de déployer massivement des applications complexes (composées d'un vaste ensemble de briques logicielles hétérogènes) sur un réseau.
Par ailleurs, Merlin bénéficie des composants développés dans le cadre du projet Avalon, regroupés sous l'appellation 'Cornerstone' et conçus à l'origine pour les containers de génération antérieure (Phoenix, Excalibur). Ces composants facilitent énormément le développement d'applications serveurs (avec par exemple gestion des connexions, des sockets, des threads, le stockage, l'ordonnancement, etc.).
Pour finir, Merlin va bénéficier des briques apportées par le framework Avalon telles que la prise en compte des Logger, la gestion des paramètres de configuration à travers des fichiers XML, la possibilité de définir de façon fine le style de vie des composants (singleton, un par thread, une instance nouvelle à chaque demande).

Spring (setter injection)

Spring propose un framework 'tout-en-un ' articulé autour d'un noyau basé sur l'inversion de contrôle, autour duquel différents services sont proposés. Spring se base notamment sur la programmation orientée aspect, pour proposer certains services, comme la gestion transactionnelle. La modularité de Spring avec la simplicité de configuration dans le choix des différentes implémentations fonctionnelles, illustre la simplicité et l'efficacité de l'approche se basant sur l'Inversion de Contrôle. Concernant l'inversion de contrôle, Spring va définir les dépendances dans un fichier de configuration à travers les paramètres des méthodes 'setter' de la classe.

Par exemple :

```
public class Pere {
        Fils fils;
        public setFils( Fils fils ){
                this.fils = fils;
        }
        public void  surveille(){
                fils.sousSurveillance();
        }
}

public class Fils {
        public void sousSurveillance(){
                System.out.println( "Sous surveillance" );
}
}
```

```
<beans>
   <bean id="myPere" class="Pere">
        <property name="fils">
       <ref bean="myFils"/>
        </property>
   </bean>

   <bean id="myFils" class="Fils">
   </bean>
</beans>
```

Dans ce fichier de configuration, la dépendance entre le fils et le père est déclarée à travers la propriété 'fils', le container se chargeant d'injecter une instance de fils via le 'setter' setFils de la classe Pere.
Le container va mettre à disposition une instance de 'Pere' de la façon suivante :

```
InputStream is = new FileInputStream( new File( "config.xml" );
```

```
XmlBeanFactory bf = new XmlBeanFactory( is );
Pere pere = (Pere) bf.getBean("myPere");
```

Spring propose d'intervenir sur le cycle de vie des objets de deux façons : Soit, via la configuration :

```
<bean id="myPere" class="Pere" init-method="init"/>

public class Pere {
    public void init() {
        // ...
    }
}
```

Soit, via l'implémentation d'une interface :

```
<bean id="myPere" class="Pere"/>

public class Pere implements InitializingBean {
    public void afterPropertiesSet() {
        // ...
    }
}
```

Dans les deux cas, le fonctionnement est le même, le container se chargera d'appeler les méthodes concernées, après injection des valeurs concernées par la dépendance.
Spring s'appuie sur le container et la programmation orientée aspect, pour proposer une liste de services non exhaustive, tels la persistance, la gestion des transactions, un framework web MVC, etc.

PicoContainer (constructor injection)

PicoContainer utilise une approche minimaliste en termes de fonctionnalités, laissant les applications de plus haut niveau reposer sur le container. NanoContainer est un exemple de container s'appuyant sur PicoContainer.
Concernant l'Inversion de Contrôle, PicoContainer se base sur les paramètres des constructeurs pour établir un graphe de dépendances entre les objets. Par exemple :

```
public class Pere {
        Fils fils;
        public Pere( Fils fils ){
                this.fils = fils;
        }
        public void  surveille(){
                fils.sousSurveillance();
        }
}

public class Fils {
        public void sousSurveillance(){
                System.out.println( "Sous surveillance" );
}
}
```

Ici, le fonctionnement de la classe 'Pere' dépend de celui de la classe fils. La dépendance est définie dans le constructeur de la classe 'Pere', à travers le paramètre 'Fils'.
Le container se chargera d'instancier les classes nécessaires pour le bon fonctionnement et la mise à disposition de la classe 'Pere'.
Les classes devant être gérées par le container sont enregistrées, pour pouvoir être instanciées successivement.

```
MutablePicoContainer pico = new DefaultPicoContainer();
pico.registerComponentImplementation(Pere.class);
pico.registerComponentImplementation(Fils.class);
Pere pere = (Pere) pico.getComponentInstance(Pere.class);
pere.surveilleFils();
```

Le cycle de vie des objets sera déterminé, via les interfaces implémentées. Par exemple, dans le cas de composants 'Startable' (supportant un démarrage et un arrêt)

```
public class Pere implements Startable {
    //...

    public void start() {
        //...
    }

    public void stop() {
        //...
    }
}
```

Dans ce cas, la classe Pere pourra être démarrée avant mise à disposition.

```
pico.start();
Pere pere = (Pere) pico.getComponentInstance(Pere.class);
//...
pico.stop();
```

Ici toutes les classes enregistrées dans le container sont 'start' avant récupération des instances, puis 'stop' après opérations.
La simplicité de PicoContainer a très vite séduit, malgré les fonctionnalités réduites, le principe étant de laisser les implémentations de plus haut niveau prendre en charge les spécificités diverses : NanoContainer s'appuie par exemple sur PicoContainer. Au final, PicoContainer s'avère être une solution très allégée et rapide à mettre œuvre dans tous types de projets.

Autres containers

La liste des containers open source de qualité, proposant des fonctionnalités intéressantes ne cesse de croître. À titre indicatif nous citerons :
- Enterprise Object Broker (EOB): qui repose sur Apache Avalon Phoenix et Apache AltRMI, et se pose comme une alternative aux EJB
- Keel Framework: se présente comme un framework de haut niveau, intégrant un ensemble assez vaste de composants (Avalon, Cocoon, Struts, Hibernate, Velocity, WebWork2, Axis, Maverick, JBoss, OpenJMS, Turbine, Lucene, BSF, Jelly, JFreeChart, Quartz, etc.)

- Hivemind : autre projet de la fondation Apache qui se positionne comme un micro noyau de base, relativement proche de Merlin sur les objectifs, mais conceptuellement différent

Positionnement par rapport aux EJB

L'assemblage des composants avec la gestion de leur dépendance n'est pas pris en compte par les EJB. Les composants EJB doivent se composer 'manuellement', en effectuant les requêtes au service de nommage JNDI.

En revanche, les différentes solutions basées sur l'IoC auront une approche minimaliste, proposant un cœur de fonctionnalités réduit et pas aussi complet et riche que les EJB. Certains frameworks tels que Spring proposeront un ensemble de briques pouvant s'articuler autour du container, mais peu (au jour d'aujourd'hui) proposent la totalité des services disponibles dans les containers EJB avec la maturité qui les caractérise.

Récapitulatif

	EJB 2.0	Avalon Merlin 3.2	Spring 1.0	Pico 1.0
Lookup	JNDI	Sur une String via le service manager	Sur une String via le bean factory	Sur une classe via le container
Type IoC	-	Interface	Setter	Constructor
Cycle de vie	Statique	A la carte	À la carte avec deux approches possibles	À la carte
Accès Remote	Oui	-	RMI Exporter	En cours (via Nanocontainer)
Gestion dépendance des librairies	-	Oui	-	-
Gestion fine des logs	-	Oui	-	-
Gestion de la sécurité des accès	Oui	-	-	-
Contexte transactionnel	Oui	-	Oui	-
Contexte utilisateur	Oui	-	-	-
Gestion de la persistance	Oui (CMP)	-	À la carte (Hibernate, JDO...)	-
AOP		-	Oui	-
Management JMX	(Oui)	(Oui)	Prevu pour 1.1	En cours

Conclusion

Les différentes solutions introduites à travers cet article mettent en évidence la volonté, de la part de la communauté des développeurs, de disposer d'une alternative crédible et simple aux containers EJB. Même si la plupart des approches restent encore loin d'égaler en termes de fonctionnalités les EJB, il faut reconnaître que pour un grand nombre de projets de petite et moyenne envergure, elles s'avèrent particulièrement appropriées, chacune avec ses domaines de prédilection :

- Merlin pour son excellente gestion des dépendances entre des briques logicielles et son approche 'micro noyau'
- Spring Framework pour la panoplie de composants connexes tels que l'intégration avec Hibernate et JDO, la gestion transactionnelle ou les frameworks MVC
- PicoContainer pour son côté 'small is beautiful', minimaliste et léger.

1 Références

Inversion of Control Containers and the Dependency Injection pattern
Martin Fowler
http://www.martinfowler.com/articles/injection.html

Pattern-Oriented Software Architecture, Volume 1: A System of Patterns
de Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal, Peter Sommerlad, Michael Stal
Publié chez John Wiley & Sons; 1 edition (August 8, 1996)
ISBN : 0471958697

Apache Avalon Framework
http://avalon.apache.org/framework/
Excalibur
http://excalibur.apache.org/
Merlin
http://avalon.apache.org/merlin/
EOB
http://www.enterpriseobjectbroker.org/
Keel Framework
http://www.keelframework.org/
Hivemind
http://jakarta.apache.org/hivemind/index.html
Loom
http://loom.codehaus.org/
Spring Framework
http://www.springframework.org/
PicoContainer
http://picocontainer.codehaus.org/
NanoContainer
http://nanocontainer.codehaus.org/

Alexis Agahi *Expert XML/J2EE (Objet Direct, filiale de Homsys Group)*

À propos de Homsys Group :

Créé en 1991, Homsys Group accompagne les entreprises dans leurs innovations et la valorisation de leurs informations à des fins de productivité et de compétitivité. Homsys Group est spécialisé autour de trois grands pôles : la Business Intelligence (Homsys), les Architectures Systèmes et Réseaux (Kedros) et les technologies Objet et Internet (Objet Direct). Homsys Group est implanté à Bordeaux, Grenoble, Lyon, Marseille, Paris, Rennes et Toulouse. www.homsysgroup.com

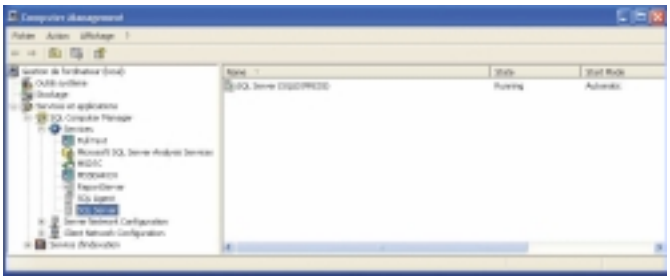
L'actualité en ligne
www.programmez.com

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

SQL Server 2005 (Yukon) :
comment créer une procédure stockée en code managé ?

Avec le prochain SQL Server 2005, un développeur pourra écrire une procédure stockée en VB.NET ou C#. Microsoft propose de tester cette nouvelle technologie dès maintenant, et gratuitement, à l'aide de SQL Server Express.

Pour la petite histoire, chaque nom de code d'une nouvelle version de SQL Server s'inspire du nom d'un parc national américain (le parc national Yukon est situé à la frontière de l'Alaska et du Canada). SQL Server Express est le nouveau nom du moteur MSDE. Il s'agit d'un sous-ensemble de SQL Server 2005 alias Yukon, mais comportant certaines limitations. La taille de la base de données ne peut excéder 4 gigas octets, la mémoire vive gérée ne peut dépasser le giga-octet, et enfin ce moteur de base de données est mono processeur.
A l'heure où nous rédigeons ces lignes, la bêta 2 est publique et librement téléchargeable. Il n'y a pas encore d'interface GUI, mis à part l'outil de gestion "SQL Computer Manager", qui permet par exemple d'arrêter ou de démarrer le service SQL Server Express. En fait, il remplace le "SQL service Manager", l'outil de gestion réseau et l'outil client réseau, anciennement livrés avec SQL Server 2000. (Fig. 1)



(Fig. 1)

Le but des concepteurs de SQL Server Express est d'arriver à réaliser un package le moins encombrant possible (pour l'instant 35 Mo, mais ils visent 25 Mo pour la bêta 3). Par conséquent, certaines fonctionnalités de SQL Server 2005, comme l'indexation et la recherche sur l'ensemble d'un champ (Full-Text Indexing) ne sont pas incluses (sinon cela doublerait la taille du package..).
Un outil baptisé "Express Manager" est néanmoins prévu avec la version bêta 3. Il s'agit d'un IDE qui fait appel au même Framework que Visual Studio et qui remplacerait l'"entreprise manager" et l'analyseur graphique de requête.
Enfin, la nouvelle base de données d'essai est maintenant baptisée "AdventureWorks". Elle est téléchargeable séparément de SQL Server Express (car elle pèse environ 100 Mo) mais il est aussi tout à fait concevable de réinjecter la base de données Northwind pour les nombreux développeurs qui ont pris l'habitude de la manipuler pour leurs essais. C'est d'ailleurs ce que nous allons faire.

Une installation presque transparente

L'installation de SQL Server Express est très simple, car un assistant nous guide pas à pas. En fait, SQL Server Express est prévu pour s'installer au besoin de manière "silencieuse". Un prérequis logiciel est l'installation du framework .net 2.0. L'assistant de SQL Server Express vous signalera d'éventuels problèmes, au niveau des ressources matérielles ou logicielles, en effectuant une douzaine de tests de compatibilité (* 1).
L'assistant d'installation vous demande ensuite de spécifier un nom d'instance. Vous pouvez laisser celui proposé par défaut ("SQLExpress") mais en tout cas, vous devez noter ce nom quelque part, car nous allons en avoir besoin. (Fig.2)

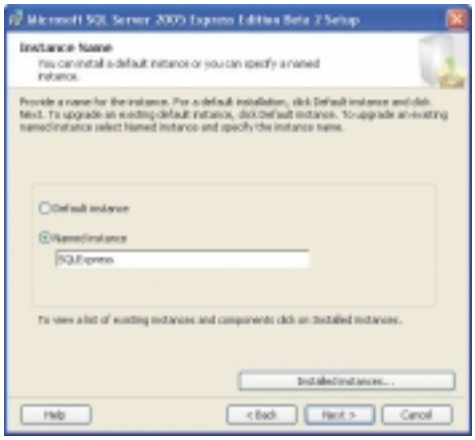
Comme nous n'avons pas à notre disposition d'outil graphique, nous allons devoir jouer avec la ligne de commande (* 2).
Avec SQLCMD, commande intégrée à SQL Server Express, nous allons ouvrir l'instance de SQL Express :

```
C:\Program Files\Microsoft SQL Server\90\Tools\binn>SQLCMD -E -S
 DEIMOS\sqlexpress
1> select name from sys.databases
2> go
name
------------------------------

master
tempdb
model
msdb

(4 rows affected)
```

Chargement d'une base de données d'exemples

Comme nous pouvons le constater, il n'y a aucune base de données d'exemples par défaut. Sur le site de Microsoft consacré à SQL Server Express, vous pouvez télécharger la base de données "Northwind and pubs Sample Databases" (package "SQL2000SampleDb.msi" (* 3)).
Celle-ci s'installera en théorie sous l'arborescence "C:\Program Files\Microsoft SQL Server 2000 Sample Database Scripts". Or, SQL Server Express n'apprécie pas les espaces dans les noms de répertoires. Nous l'avons renommée, ce qui a permis de lancer facilement le script d'installation sql :

(1) Extrait du rapport de compatibilité avant l'installation proprement dite*
(2) En fait, ce n'est pas tout à fait exact : vous pouvez également créer une nouvelle base de données à l'aide de Visual Studio ou Visual Basic Express.*
(3) L'adresse exacte de téléchargement est : http://www.microsoft.com/downloads/details.aspx?FamilyID=06616212-0356-46a0-8da2-eebc53a68034&DisplayLang=en*



(Fig. 2)

```
C:\Program Files\Microsoft SQL Server\90\Tools\binn>SQLCMD -E -S
DEIMOS\sqlexpress -i "D:\MicrosoftSQLServer2000SampleDatabase
Scripts\instpubs.sql"
Changed database context to 'master'.
Beginning InstPubs.SQL at 02 Sep 2004 11:30:26:070 ....
Creating pubs database....
Changed database context to 'pubs'.
Now at the create table section ....
Now at the create trigger section ...
Now at the inserts to authors ....
Now at the inserts to publishers ....
Now at the inserts to pub_info ....
Now at the inserts to titles ....
Now at the inserts to titleauthor ....
Now at the inserts to stores ....
Now at the inserts to sales ....
Now at the inserts to roysched ....
Now at the inserts to discounts ....
Now at the inserts to jobs ....
Now at the inserts to employee ....
Now at the create index section ....
Now at the create view section ....
Now at the create procedure section ....
Changed database context to 'master'.
Ending InstPubs.SQL at 02 Sep 2004 11:30:32:480 ....
```

Et aussi :

```
C:\Program Files\Microsoft SQL Server\90\Tools\binn>SQLCMD -E -S
DEIMOS\sqlexpress -i "D:\MicrosoftSQLServer2000SampleDatabase
Scripts\instnwnd.sql"
...
```

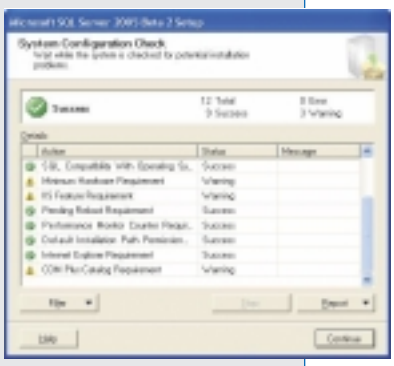
Ce qui donnera au final :

```
1> select name from sys.databases
2> go
name
_____
master
tempdb
model
msdb
Northwind
pubs

(6 rows affected)
```

Nous pouvons maintenant tenter une requête :

```
1> Use pubs
1> select au_fname,phone from authors
2> go
au_fname        phone
_____           _____
Johnson         408 496-7223
Marjorie        415 986-7020
...
(23 rows affected)
```

System Configuration Check



Neuf tests ont été passés avec succès, mais il y a trois messages d'avertissement (non fatals).

WMI Service Requirement (Success)
- WMI Service Requirement
- Check Passed

MSXML Requirement (Success)
- MSXML Requirement
- Check Passed

Operating System Minimum Level Requirement (Success)
- Operating System Minimum Level Requirement
- Check Passed

Operating System Service Pack Level Requirement. (Success)
- Operating System Service Pack Level Requirement.
- Check Passed

SQL Compatibility With Operating System (Success)
- SQL Compatibility With Operating System
- Check Passed

Minimum Hardware Requirement (Warning)
- Minimum Hardware Requirement
- The current system does not meet the minimum hardware requirements for this SQL Server release. For detailed hardware and software requirements, see the readme file or SQL Server Books Online.
- The current operating system does not meet Service Pack level requirements for this SQL Server release. Install the most recent Service Pack from the Microsoft Download Center at http://www.microsoft.com/downloads/, and then run SQL Server Setup again. For system requirements, see Hardware and Software Requirements, in the readme file or in SQL Server Books Online. ...

Création d'une procédure stockée managée en VB

Une procédure stockée traditionnelle se rédige traditionnellement en T-SQL. Microsoft ne va pas arrêter l'évolution de ce langage, car il est le plus rapide en exécution, vu qu'il est natif. En effet, le code managé va appeler une API, qui elle-même sollicitera du code T-SQL. Remarquez que vous pouviez déjà écrire sous SQL Server 2000 une procédure externe en C++ (XP : Extended Stored Procedure), mais une telle architecture n'est pas aussi robuste et sécuritaire que du code managé (sans compter d'autres avantages comme le choix du langage ou la facilité de débogage). Microsoft recommande d'ailleurs la migration des procédures XP vers du code managé.

Pour que SQL Server 2005 ou SQL Server Express accepte une procédure stockée managée, il suffit de la construire comme un assemblage qui sera lié à la bibliothèque sqlaccess.

Ouvrons VB Express ou Visual Studio, ou encore le wordpad, et encodons-y ceci :

```
Imports System.Data.SqlServer

Public Class MaProcedureStockee
        Public Shared Sub EchoSQL(nom As String)
                Dim pipe AS System.Data.SqlServer.SqlPipe = Sql
Context.GetPipe()
                pipe.Send("Echo : "  & Nom)
        End Sub
End Class
```

Le Framework .net 2.x est livré avec le compilateur vb VBC. Exécutons-le pour créer l'assemblage :

```
C:\>C:\WINDOWS\Microsoft.NET\Framework\v2.0.40607\vbc.exe
c:\echosql.vb /t:library /r:"C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Binn\s
qlaccess.dll"  /out:maproc.dll
Microsoft (R) Visual Basic .NET Compiler version 8.0.40607.42
for Microsoft (R) .NET Framework version 2.0.40607.42
Copyright (C) Microsoft Corporation 1987-2003. All rights reserved.
```

Maintenant, il suffit d'installer l'assemblage sous SQL Server Express.

```
C:\Program Files\Microsoft SQL Server\90\Tools\binn>SQLCMD -E -S
DEIMOS\sqlexpress
1> CREATE ASSEMBLY sproc from 'C:\maproc.dll'
2> go
1> CREATE PROCEDURE EchoSQL(@nom AS NVARCHAR(max)) AS
EXTERNAL NAME sproc.MaProcedureStockee.EchoSQL
2> go
1> EXEC EchoSQL 'Cette procédure stockée est managée par dot net !'
2> go
Echo : 'Cette procédure stockée est managée par dot net !'
1>
```

Attention, nous devons attirer votre attention sur deux points. Premièrement, si vous désirez compiler la DLL sous VB, vous devez ajouter l'espace de nom System.Data.SqlServer aux références connues (Project/add reference/browse/ puis il faut pointer vers sqlaccess.dll). Deuxièmement, la syntaxe peut varier d'une version de sqlaccessà l'autre. Il s'agit bien ici de System.Data.SqlServer.SqlPipe et non de System.Data.SqlServer.Pipe (ce qui se rapporte à la version 9.00.852 de la bibliothèque).

SQL Server 2005/Express stocke les informations se rapportant aux procédures stockées externes dans plusieurs tables systèmes : sys.assemblies, sys.assembly_files et sys.assembly_references.

```
1> select * FROM sys.assembly_files
2> go
assembly_id name      file_id     content
____________________________________
      65536 C:\maproc.dll
1 0x4D5A90000300000004000000FFFF
…

(1 rows affected)
1>
```

Il est non seulement possible d'installer une procédure stockée sous la forme d'un assemblage avec CREATE ASSEMBLY, mais aussi une fonction, une procédure, un déclencheur, ou un TYPE (CREATE ASSEMBLY, CREATE FUNCTION, CREATE PROCEDURE, CREATE TRIGGER ou CREATE TYPE).

Notez enfin que si vous utilisez Visual Studio il sera facile de créer un point de débogage pour l'assemblage (breakpoint).

Création d'une procédure stockée managée en C#

Vous avez sans doute compris le principe :

1) nous éditons un fichier C:\essai.cs qui contient le code source de la classe C# ;

```
using System.Data.SqlServer;

public class NET_C_SHARP
{

    public static void MaProc()
     {
         SqlPipe myPipe = SqlContext.GetPipe();

         myPipe.Send("Procédure stockée managée en C# !");

         SqlCommand cmd = SqlContext.GetCommand();

         cmd.CommandText ="select au_fname,phone from authors";

         myPipe.Send(cmd.ExecuteReader());

     }
}
```

2) nous compilons cette classe avec le compilateur C# (du Framework .net 2.0) pour obtenir un assemblage ;

```
C:\>C:\WINDOWS\Microsoft.NET\Framework\v2.0.40607\csc.exe
```

```
/t:library /r:C:\sqlaccess.dll /out:essai.dll C:\essai.cs
Microsoft (R) Visual C# .NET Compiler version 8.00.40607.42
for Microsoft (R) Windows (R) .NET Framework version 2.0.40607
Copyright (C) Microsoft Corporation 2001-2003. All rights reserved.
```

3) nous sélectionnons la base de données pubs et y associons l'assemblage ;

```
1> use pubs
2> go
Changed database context to 'pubs'.
1> CREATE ASSEMBLY test_c_sharp from 'C:\essai.dll'
2> go
```

4) nous créons une procédure (EXTERNAL NAME)

```
1> create procedure proc_c_sharp as EXTERNAL NAME test_c_sharp
.NET_C_SHARP.MaProc
2> go
```

5) nous appelons notre procédure :

```
1> proc_c_sharp
2> go
Procédure stockée managée en C# !
au_fname            phone
——————————— ————————————
Johnson             408 496-7223
Marjorie            415 986-7020
…
 (23 rows affected)
1>
```

Le résultat obtenu est correct, étant donné que nous obtenons la même sortie que précédemment (lors de la création de la base de données Northwind), mis à part qu'ici, c'est la procédure stockée managée qui se charge de tout...

Voici les principales classes de l'espace de nom System.Data.SqlServer :

```
SqlContext
GetCommand()
GetConnection()
GetPipe()
GetReturnResultSet()
GetTransaction()
GetTriggerContext()
```

```
SqlDefinition
CommandText
CommandType
ParameterCount
GetParameterDirection()
GetSqlMetadata()
SqlDefinition()
```

```
SqlExecutionContext
ISqlExecutionContext
IDBExectionContext
IDisposable
ISqlGetTypedData
IGetTypedData
ISqlSetTypedData
ISetTypedData
```

```
SqlPipe
SendingResults
Execute()
Send()
SendResultsEnd()
SendResultsRow()
SendResultsStart()
```

```
SqlTriggerContext
ColumnsUpdated
EventData
TriggerAction
```

En conclusion

Avec SQL Server Express ou/et SQL Server 2005, le développeur peut créer ses procédures stockées en code .net. Il s'agit d'une avancée extraordinaire, mais encore faut-il bien peser son utilité réelle : d'abord T-SQL s'avère évidemment le plus rapide, et le code managé ne sera intéressant que lorsqu'il faudra effectuer des calculs purs, par exemple le cryptage du numéro d'une carte de crédit dans une table ; ensuite, il ne faut pas oublier que vous avez toujours la possibilité de faire exécuter du code à un serveur d'applications intermédiaire. De nombreux paramètres sont alors à prendre en compte : le choix de votre architecture, la puissance de vos serveurs, les ressources matérielles et logicielles de vos clients, etc.

Xavier Leclercq - *Xavier.Leclercq@programmez.com*

NIVEAU

La programmation du shell de Windows

Personnaliser l'interface utilisateur graphique de Windows par la programmation du Shell est une activité gratifiante. Mais qui requiert l'écriture de serveurs COM. Découvrons cela ensemble.

SOURCES DE L'ARTICLE
WWW.PROGRAMMEZ.COM

Que le terme de Shell ne nous trompe pas. Alors que sous Unix, le mot désigne l'interface en ligne de commande, sous Windows, il s'agit de l'interface graphique. Autant dire qu'il s'agit là d'un vaste monde dont font partie le bureau, la barre des tâches, les menus contextuels, les boîtes de dialogues de configuration, l'explorateur, pour ne citer que ces éléments. Pour sa plus grande satisfaction, le programmeur à la possibilité d'agir sur, ou d'interagir avec ces éléments qui sont des composants logiciels, basés sur l'architecture COM pour l'immense majorité. Ainsi le mois dernier, nous avons travaillé avec le planificateur de tâches de Windows. Nous avons vu qu'il était lui aussi un serveur COM et nous l'avons programmé côté client. Ceci nous a donné l'occasion d'exposer les bases de l'architecture COM (n'hésitez surtout pas à vous reportez à Programmez! 67) et de nous faire la main, car la programmation COM côté client est plus facile que côté serveur. Nous nous proposons aujourd'hui d'aller plus loin. Est-ce à dire, si nous avons l'intention d'écrire un serveur COM, que nous allons enrichir l'interface Windows d'un nouveau logiciel ? Non (mais un jour qui sait ? ;). En fait, il se trouve que les composants du Shell, en plus d'être des composants COM, s'attendent à collaborer avec d'autres composants COM. Voici comment les choses se présentent. Quand un élément de Windows offre une possibilité d'extension, il va rechercher dans la base de registre si un tel composant d'extension y est inscrit. Ce composant doit être lui aussi un serveur COM qui implémente une ou plusieurs interfaces prédéfinies, ce qui permet au composant Windows d'invoquer les méthodes de l'extension. Autrement dit, nous pouvons considérer que Windows est une architecture à base de plug-ins, un peu à la manière de celle d'Eclipse que j'ai eu l'honneur de vous présenter dans Programmez! 63, 64 et 65. Mais pour écrire un "plug-in Windows", nous n'écrivons pas une classe Java, mais un serveur COM et nous n'utilisons en aucun cas Java, mais au contraire un langage supportant les pointeurs. COM étant a priori indépendant du langage, n'importe quel langage à pointeurs convient : C, Delphi, etc. Cependant, parce que c'est le langage à pointeurs préféré de votre serviteur, et, aussi et surtout, parce qu'il simplifie énormément le travail, nous emploierons C++. Notre but est d'ajouter une entrée au menu contextuel de l'explorateur, lorsqu'un fichier d'extension .prog est sélectionné. Cette entrée sera opérationnelle et invoquera, si activée, une application pour ouvrir le fichier.prog. Bien entendu, on pourrait obtenir ceci par un simple ajout de clés dans la base de registres, depuis le panneau de configuration, mais si nous voulions un comportement plus riche au moment de la sélection du menu, ou encore si nous voulons que le texte du menu s'adapte spécifiquement au fichier sélectionné, au lieu d'être identique pour tout fichier .prog, alors écrire une extension au Shell devient nécessaire. Pour parvenir à notre but, nous devons travailler en deux temps, étant donné la complexité de la question. D'abord, dans cet article, nous allons comprendre le minimum vital à propos de COM côté serveur et nous familiariser avec les outils de programmation pour écrire un petit serveur rudimentaire ne se greffant pas dans Windows, puis le mois prochain nous réaliserons effectivement notre but. Si cet apprentissage est long et parfois un peu ardu, une fois le savoir - faire acquis, les possibilités d'application dans Windows sont très nombreuses et amusantes ce qui nous récompensera largement ainsi que nous le découvrirons dans d'autres articles à venir.

Les outils

Même avec la puissance de C++, écrire un serveur COM de A à Z est une tâche lourde, pour ne pas dire périlleuse. C'est pourquoi, nous allons nous aider d'une librairie, afin de ne pas partir de zéro. Les MFC de Microsoft permettent de faire du COM, mais nous les trouvons malcommodes. Nous leur préférons une autre bibliothèque Microsoft: ATL. ATL signifie Active Template Library, le mot Active, suggérant que la librairie convient bien à l'écriture de composants COM particuliers, les contrôles ActiveX. ActiveX dépasse notre propos, mais, qui peut le plus peut le moins, et ATL permet d'écrire du code léger et efficace pour du simple COM, ce qui est bien agréable. ATL consiste essentiellement en un jeu de classes patron C++ agrémenté de quelques-unes de ces macros étourdissantes dont Microsoft a le secret. Nous devons donc utiliser un outil supportant ATL. Nous choisissons Visual C++ 6.0 et C++Builder 6.0. Chacun d'eux a ses propres particularités que nous allons découvrir. Dans les deux cas, ATL nous décharge du gros de la gestion de COM. Ainsi, les méthodes QueryInterface, AddRef et Release des interfaces IUnknown ou IDispatch, (le lecteur voudra bien se reporter à l'article du mois précédent) sont implémentées automatiquement. Grâce à ATL, nous voyons notre serveur COM pour ce qu'il est dans le plus essentiel : un agrégat d'interfaces. Et dans le contexte C++, une simple classe implémentant les méthodes des interfaces. Notre serveur d'exemple ne comprendra qu'une interface et cette dernière ne comprendra que 2 méthodes. La première, Get2, reçoit un pointeur sur un entier dont elle double la valeur. La seconde, GetString, reçoit un tampon et recopie une chaîne dedans. Vous trouverez sur le Cd-Rom, ou, si ce n'est pas le cas, sur le site officiel www.programmez.com, les projets complets (serveur et client d'essai) pour Visual C++ et C++Builder. Les projets venant sous l'arborescence C:\DemoCom, vous pourrez les utiliser directement après extraction de l'archive. Dans les deux cas, compilez d'abord le projet serveur, puis le projet client. Avant de passer à la pratique, nous ne saurions trop vous recommander de faire une sauvegarde de votre base de registres, ce qui revient à sauvegarder les fichiers c:\windows\system.dat et c:\windows\user.dat sous Windows 95, à utiliser l'utilitaire

prévu à cet effet sous les Windows ultérieurs, ou encore à créer un point de restauration, si vous êtes l'heureux détenteur d'un Windows XP.

Visual C++

Commençons donc par créer un espace de travail sous le nom 'Visual'. Dans cet espace, nous ajoutons un projet de type 'ATL COM AppWizard', de nom DemoComVisual. (Fig.1) Ce nom sera le nom de la librairie de types générée par le wizard. Une librairie de types est un ensemble d'informations sur les serveurs COM ou Automation qu'elle contient. Cela n'intéresse principalement que les langages interprétés et ne sert à rien ici, mais ne gêne pas non plus. En revanche, puisque nous acquérons un savoir faire pour la programmation Shell, nous devons faire en sorte que notre serveur vienne dans une librairie partagée (dll) et ne soit pas une application autonome. Nous devons donc accepter l'option par défaut proposée par la boîte de dialogue suivante.(Fig.2). Le wizard nous a généré quatre fichiers pour l'instant assez désertiques. Nous remarquons un fichier .idl au contenu étrange dont nous reparlerons, et dans le fichier .cpp nous voyons des fonctions globales, dont les noms donnent à penser qu'elles enregistrent quelque chose. Nous pouvons déjà compiler notre projet, et effectivement, à la fin de l'opération nous voyons

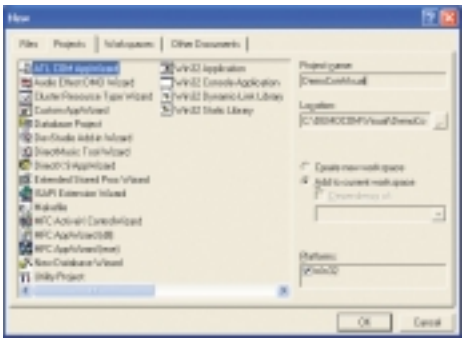


Fig.1 : C'est parti, nous créons notre projet ATL.

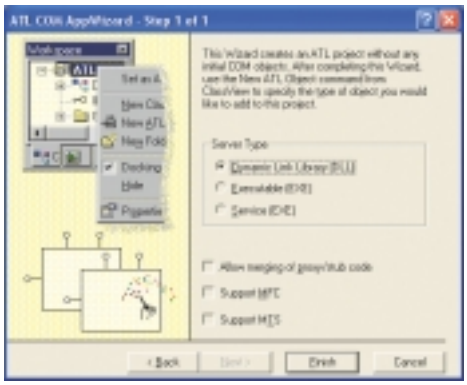


Fig.2 : Notre serveur COM devra résider dans une dll.

```
performing registration
```

De quoi s'agit-il ? Nous savons déjà qu'un serveur COM doit être inscrit dans la base de registre. Il en va de même pour une librairie de types. L'enregistrement est fait sous la clé \HKEY_CLASSES_ROOT\TypeLib. A la fin de la compilation Visual lance regsvr32.exe, un utilitaire Microsoft qui réside sous c:\windows\system32. Celui ci invoque les fonction globales mentionnées plus haut, et ATL aidant, la librairie, ainsi que tous les composants qu'elle contient, sont enregistrés. Ceci est bien pour la phase de développement, mais si vous déplacez la dll, ou si vous voulez l'installer sur une autre machine, alors vous devez taper dans la console :

```
regsvr32 librairie.dll
```

éventuellement, avec le commutateur /s, si vous voulez que l'opération se déroule en silence. Enfin, si vous voulez effacer une librairie de types ou un objet COM du registre (ce que Visual ne fait pas tout seul), saisissez :

```
regsvr32 /u /s librairie.dll
```

Il est maintenant temps d'incorporer notre composant COM à notre projet. Bizarrement, cela se fait depuis le menu de Visual, à Insert|New ATL Onject. Sélectionnez 'Simple Object' dans la boîte de dialogue qui surgit (Fig.3) Ensuite, un dialogue vous demande un nom de serveur. Donnez par exemple MonServeurVisual (Fig.4) et surtout faites attention aux attributs sous le deuxième onglet. Cochez 'Apartment', 'Dual' et 'Yes'. Apartment nous évite tous problèmes de thread, Dual fait générer une interface à deux facettes, Automation et COM. Là non plus, nous n'en avons pas besoin de tant, mais cela ne gêne pas et nous dispense de finasser.

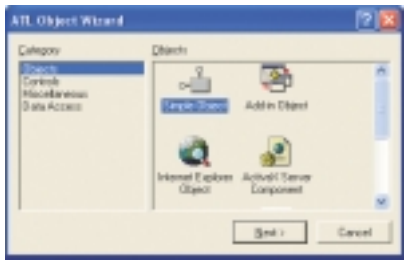


Fig.3 : Nous insérons notre composant COM à notre projet.

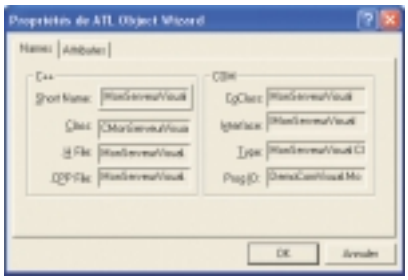


Fig.4 : Baptisons notre classe et faisons attention à ses attributs.

Un script dans le registre.

Ceci fait, notre projet s'est enrichi de quelques fichiers et le fichier .idl a lui-même été étoffé. Dispensez-vous d'aller visiter les curiosités sous 'External Dependancies', Visual se charge de tout et c'est très bien ainsi. Par contre, remarquez, sous 'Resources Files', l'apparition d'un fichier .rgs. Ce fichier contient un script ATL, destiné à l'écriture de clés dans le registre. A la compilation, le script est converti en une ressource, ainsi que vous pouvez le vérifier dans votre fichier.rc et regsvr32 crée les clefs lorsqu'il est appelé. Le mois prochain nous devrons enrichir manuellement ce script. Sa syntaxe est toute simple.

Seuls deux mots clés méritent notre attention : ForceRemove et NoRemove. Le premier fait qu'une clé est effacée avant d'être écrite lors de l'inscription dans le registre, l'autre fait que la clé ne sera pas enlevée lors de la désinscription. On emploie NoRemove lorsque la clé est commune à plusieurs applications ou configurations et qu'il ne faut pas l'enlever.

Avoir une IDL avec le bon type

Parlons maintenant du fichier .idl. Le contenu de ce fichier n'est que déclarations, selon une syntaxe établie par Microsoft. A la construction du projet, MIDL le compilateur idl, génère des fichiers .h et .cpp avec lesquels nous pouvons travailler. Nous l'avons dit, le fichier .idl est maintenant enrichi. Notamment, nous y voyons une déclaration :

```
interface IMonServeurVisual : IDispatch
{
};
```

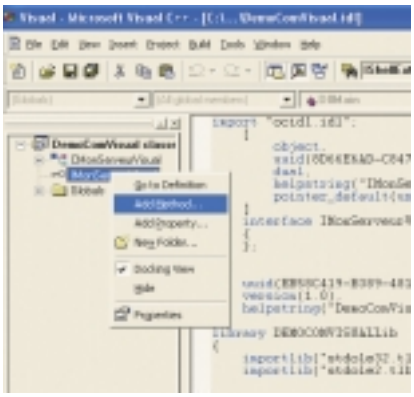


Fig.5 : Insertion d'une méthode dans notre interface.

Fig.6 : Déclaration de notre méthode via l'IDE.

Nous comprenons que le wizard a inclus une interface dans notre composant et que cette interface est maintenant vide. Nous pourrions déclarer les méthodes de notre interface manuellement directement dans le fichier. Ce n'est pas très difficile de faire de l'IDL direct et c'est parfois incontournable, par exemple, si l'on veut ajouter une interface au composant. Pour l'instant, nous passons par le ClassViewer de l'IDE (Fig.5 et 6), mais il est très intéressant de suivre ce que fait Visual dans le fichier pour apprendre comment travailler en IDL manuellement. Nous déclarons donc notre méthode Get2. Celle-ci reçoit un paramètre et nous le déclarons comme ceci :

```
[in, out]int *operande
```

Les mots in et out sont très importants. C'est leur présence qui assure que la plomberie COM fonctionnera dans les deux sens. Ceci fait, la déclaration de la méthode apparaît dans le fichier .idl mais surtout dans le fichier .h et nous n'avons plus qu'à compléter l'implémentation dans le fichier .cpp.

```
STDMETHODIMP CMonServeurVisual::Get2(int *operande)
{
        *operande *= 2;

        return S_OK;
}
```

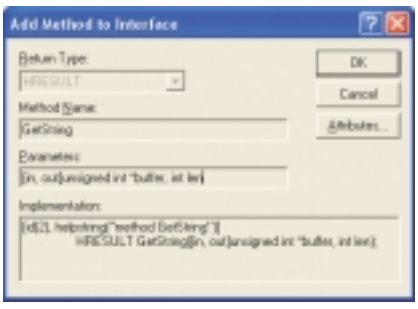


Fig.7 : ATL et COM ne connaissent pas le char, alors nous passons byte**

Passons à la deuxième méthode. Nous rencontrons une difficulté. Nous souhaiterions travailler avec un pointeur de char, mais MIDL ne comprend pas ça. En effet, en C++, que char soit signé ou non dépend de l'implémentation, ce qui peut engendrer des problèmes subtils dans le contexte de COM. Nous travaillerons donc avec un type de taille équivalente et nous passerons un pointeur de byte à notre méthode (Fig.7) et nous travaillerons au chausse-pied :-); Nous déclarons donc les paramètres en IDL comme ceci :

```
[in,out]byte* buffer, [in]int len
```

Voici maintenant l'implémentation de la méthode ;

```
STDMETHODIMP CMonServeurVisual::GetString
(byte *buffer, int len)
{
        char *texte = "Programmez!";

        if(len < 1)
                return E_INVALIDARG;

        ::StrCpyN(reinterpret_cast<char *>(buffer),
texte, len);

        return S_OK;
}
```

Vous pouvez en apprendre plus sur les types académiques reconnus par MIDL en faisant une recherche à http://msdn.microsoft.com. Pour notre exemple, ce que nous avons fait suffit et fonctionne, mais si un jour vous faites du COM destiné à un réseau comprenant des machines hétéroclites, la plus grande rigueur quant aux types COM s'imposera. Notre serveur COM est terminé. Nous n'avons plus qu'à le compiler et à le tester avec le client fourni sur le site. Ce client est écrit selon les principes et avec les outils donnés dans l'article du mois dernier. Nous n'y revenons pas.

C++ Builder

Nous passons à C++Builder. Bien sûr, les principes généraux sont les mêmes, bien que Borland ait apporté quelques modifications à ATL. Globalement, le maniement de C++Builder est beaucoup moins lourd qu'avec Visual. Nous fournissons sur le site un groupe de projets contenant le serveur et son client. Vous devrez commencer par créer un projet de type 'ActiveX Library'. Ceci fait, toujours depuis le Wizard principal, ajoutez un 'COM Object' à la librairie. Bien évidement, choisissez 'Apartment' comme model de thread et gardez cochée la case 'Mark interface OleAutomation'. (Fig.8). Cette opération faite, vous arrivez sur le visionneur de la librairie de type à partir duquel vous pouvez tout faire : ajouter (ou retirer) des méthodes et même des interfaces, ou de nouvelles CoClasse. Pour la méthode

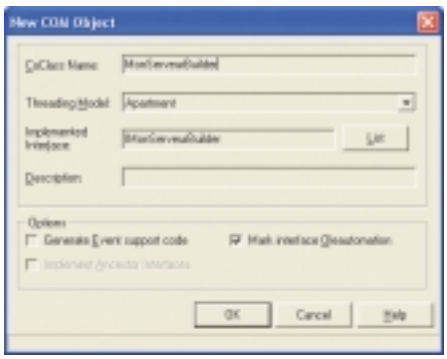


Fig.8 : Création du composant sous C++Builder

GetString, saisissez manuellement le type byte* pour le paramètre buffer et donnez [in, out] comme 'modifier' (cliquer dessus fait surgir une boîte de dialogue) (Fig.9). Compilez une première fois le projet pour générer le fichier .cpp d'implémementation. Puis écrivez le code des méthodes, qui est évidemment le même que pour Visual. Remarquez quand même que le byte* est devenu un signed_char*, à l'initiative de Builder, ce qui n'est pas mal. Nous voyons que C++Builder masque totalement le fichier IDL et nous n'avons jamais à travailler avec IDL directement. Contrairement à Visual, C++Builder n'enregistre pas automatiquement les composants après la compilation. Vous devez faire cette opération depuis le menu 'Run|Register ActiveX Server'. En revanche, vous pouvez désinscrire les composants depuis l'IDE. Celui-ci travaille avec l'utilitaire tregsvr.exe qui vient avec l'IDE. Autre différence importante, C++Builder ne génère pas de script à destination du registre. Vous ne trouverez donc pas de fichier .rgs dans votre projet. Pour ajouter des clés supplémentaires dans le registre, ce qui sera nécessaire la prochaine fois, nous devrons modifier la méthode en ligne UpdateRegistry que vous trouverez dans le fichier en-tête MonServeurBuilderImpl.h. Le client est évidement le même, aux inclusions et noms de composants près, que celui de Visual. Nous savons maintenant programmer un composant

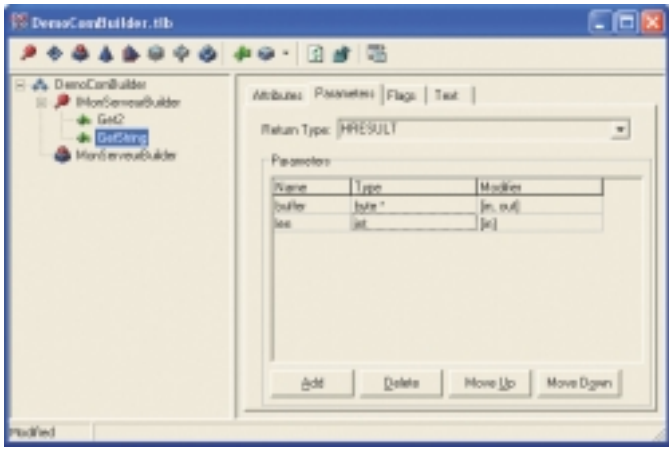
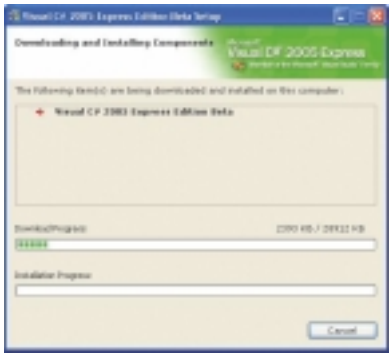


Fig.9 : Définissez les paramètres des méthodes depuis le visionneur de librairie de type.

COM. Nous sommes donc en mesure d'entendre le Shell de Windows, ce que nous ferons ensemble le mois prochain, si vous le voulez bien.

Frédéric Mazué - *fmazue@programmez.com*

Construisez vos rapports PDF avec Visual C# 2005 Express



Microsoft n'a pas implémenté nativement dans son framework .net des classes permettant de créer des documents PDF. Or il s'agit d'une fonctionnalité des plus intéressantes, étant donné que ce format est actuellement très répandu. Nous allons vous présenter quelques classes en provenance d'assemblages Open Source capables de réaliser ce travail.

De nombreux éditeurs commercialisent des bibliothèques de fonctions qui permettent à un programmeur C# de créer facilement des documents Adobe PDF (Portable Document Format). Citons de manière non exhaustive : Aspose PDF, PDF-Writer.NET et DynamicPDF MergerForNET. Cependant, et c'est assez récent, d'excellentes extensions Open Source peuvent aussi remplir ce rôle comme ItextSharp, SharpPDF et Report.net.

Un mot sur C# 2005 Express

Nous allons examiner comment les utiliser avec l'outil Microsoft Visual C# 2005 Express, mais vous pouvez évidemment aussi employer la ligne de commande (avec le compilateur C# livré gratuitement avec le Framework . net), ou encore évidemment utiliser Visual Studio.

C# 2005 Express est en bêta et son socle est le Framework .net 2.0. Microsoft a décidé de créer une gamme complète de produits "allégés" dans le but sans doute de fidéliser des développeurs Windows qui lorgnent du côté de l'Open Source. Ces produits sont en effet fort limités dans le cadre d'un usage professionnel, mais sont pour l'instant gratuits au téléchargement. Si vous ne connaissez pas C#, il s'agit d'un excellent moyen de s'initier (les écoles s'y intéresseront certainement). Remarquez au passage que C# Express en combinaison avec le Framework 2.0 permet de coder des itérateurs, d'appeler des méthodes anonymes, d'accéder à l'univers de la généricité, etc.

Report.net

Après avoir décompressé l'archive de Report.net, vous devez ouvrir un nouveau projet ("Bonjour_PDF_REPORT_NET") sous Visual C# 2005 Express. Ajoutez maintenant aux références, l'assemblage "Reports.dll" pour que l'espace de nom Root.Reports soit reconnu.

```
using Root.Reports;

namespace Bonjour_PDF_REPORT_NET
{
    class Program
    {
        static void Main(string[] args)
        {
```

Cet exemple de base est simple à comprendre : nous créons une nouvelle page PDF qui affichera au centre à la position Y 120, la chaîne "Bonjour !" le tout, en taille 18 et police Arial.

```
            Report report = new Report(new PdfFormatter());
            FontDef fd = new FontDef(report, "Arial");
            FontProp fp = new FontPropMM(fd, 18);
            Page page = new Page(report);
            page.AddCenteredMM(120, new RepString(fp, "Bonjour !"));
            RT.ViewPDF(report, "Bonjour.pdf");
        }
    }
}
```

Au final, Report.net invoquera Adobe Acrobat Reader pour afficher le fichier généré. La classe Report Tools contient également une méthode du nom de PrintPDF() qui, comme son nom l'indique, permet à l'utilisateur d'imprimer le résultat généré...

Mais il est fort probable que vous décidiez de créer des rapports PDF en partant d'une source de base de données. Avec le gestionnaire de table du nom de "TableLayoutManager" vous pourrez automatiser la mise en pages de votre rapport.

```
using (TableLayoutManager tlm = new TableLayoutManager(fp_Header)) {
...
```

Puis, pour chaque ligne :

```
using (OleDbDataReader oddr = odc.ExecuteReader()) {
        BrushProp bp_USA = new BrushProp(this, Color.FromArgb(255,
180, 180));
        while (oddr.Read()) {
         ...
          tlm.NewRow();
          tlm.Add(0, new RepString(fp, oddr.GetString(0)));
```

Report.Net est livré avec une documentation en anglais non complète. Cependant, son code source fait partie du package et le logiciel est distribué selon les termes de la GLPL (GNU Lesser Public License). Seule la bibliothèque "reports.dll" devra être distribuée pour que votre application fonctionne chez un client : il suffit de l'installer dans le même répertoire que l'exécutable (ou de partager l'assemblage *), avec évidemment le lecteur Adobe Acrobat Reader qui devra être présent sur cette même machine.

Sharp PDF

C'est un logiciel relativement récent, et vous pouvez soit télécharger directement la dll, soit le package complet, avec son code source et sa documentation en anglais (en GLPL).

L'objet de base est le pdfDocument, conteneur du document qui est vide au départ, auquel viendront se greffer des pages (pdfPage) dans lesquelles on pourra ensuite ajouter des objets graphiques, réels ou virtuels.

Les objets réels sont des objets qui peuvent s'écrire directement en code pdf (comme des lignes : page.drawline), par opposition aux objets virtuels qui se composent d'une collection d'objets réels (une table virtuelle composée de lignes et de textes réels).

```
using sharpPDF;
using sharpPDF.Enumerators;

namespace Bonjour_SHARP_PDF
{
    class Program
    {
        static void Main(string[] args)
        {
            pdfDocument monDoc = new pdfDocument("Exemple","Xavier");
            pdfPage maPage = monDoc.addPage();
                maPage.addText("Bonjour !", 200, 450, predefinedFont.cs
Helvetica, 20);
            monDoc.createPDF(@"c:\bonjour_C_sharp_pdf.pdf");
            maPage = null;
            monDoc = null;
        }
    }
}
```

Le document PDF de base est intitulé "Exemple" avec "Xavier" comme auteur, les polices potentielles sont prédéfinies par énumération (predefinedFont), et enfin, le fichier est créé avec CreatePDF. Pour modifier la taille de la page, vous pouvez passer les dimensions à AddPage (comme addPage(100,100);).

Examinons maintenant comment créer une table. Le code est plus logique à comprendre qu'avec Report.NET et nous allons ici vous fournir un listing complet et opérationnel. La construction d'une telle application est limpide : on crée le conteneur document, puis une page (ou plusieurs), et enfin dans cette page on y ajoute une table, pour laquelle on définira des noms de colonnes (TableHeader) et des valeurs pour chaque champ (columnValue). La mise en pages est quasi automatique, après avoir ajouté maTable.addRow(maLigne), mais on peut néanmoins choisir le style et la couleur de chaque ligne.

Ce qui donne comme code :

```
using sharpPDF;
using sharpPDF.Enumerators;
using sharpPDF.Exceptions;

namespace Table_Sharp_pdf
{
  class Program
```

A propos des assemblages partagés

Pour éviter l'enfer des DLL (c'est-à-dire l'écrasement d'une DLL par une autre, non compatible), . net vous permet de partager des assemblages. Ceux-ci doivent être installés dans un cache spécial du nom de GAC (Global Assembly Cache), sous l'arborescence physique "Windows\Assembly". Mais pour installer un assemblage dans ce répertoire, il ne suffit plus de bêtement le recopier (sinon on retomberait dans le même schéma des DLL hell).

Il faut que l'assemblage en question possède un nom fort (strong name), et il peut être signé par une clé privée et une clé publique. Le nom fort est composé du nom du composant, de numéros de version et de sa clé publique.

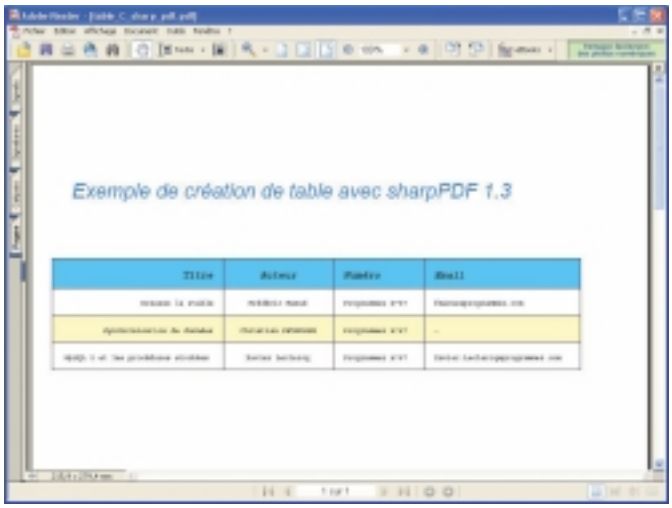
Toutes ces informations figurent dans un fichier baptisé AssemblyInfo.cs. C'est à ce niveau que vous devez y indiquer le titre, la description, le numéro de version (numéro majeur, numéro mineur, build number et revision number (vous pouvez indiquer * pour ces deux dernières informations qui seront alors générées automatiquement)).

```
[assembly: AssemblyTitle("Report.NET library")]
[assembly: AssemblyDescription("This library provides tools for
creating PDF reports")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("root-software ag, Otto Mayer, Stefan
Spirig, Roger Gartenmann")]
[assembly: AssemblyProduct("Report.NET")]
[assembly: AssemblyCopyright("Report.NET copyright 2002-2004
root-software ag, Bürglen Switzerland - O. Mayer, S. Spirig, R.
Gartenmann, all rights reserved")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]

// Version information
[assembly: AssemblyVersion("1.01.*")]
[assembly: AssemblyFileVersion("1.01.00")]
[assembly: AssemblyInformationalVersion("1.01.00")]

// Assembly signing
[assembly: AssemblyDelaySign(false)]
[assembly: AssemblyKeyFile("")]
[assembly: AssemblyKeyName("")]

Pour créer les clés vous pouvez utiliser l'utilitaire sn (strong name)
sn –K reportnet.snk

Puis indiquer [assembly: AssemblyKeyFile("reportnet.snk")], et
régénérer la DLL.
```

```
{
    static void Main(string[] args)
    {
        pdfDocument monDoc = new pdfDocument("Table avec Sharp PDF", "Xavier", false);
        pdfPage maPremierePage = monDoc.addPage();
        maPremierePage.addText("Exemple de création de table avec sharpPDF 1.3,40, 660,predefinedFont.csHelveticaOblique, 20, new pdfColor(predefinedColor.csCyan));

        pdfTable maTable = new pdfTable();

        maTable.borderSize = 1;
        maTable.borderColor = new pdfColor(predefinedColor.csDarkBlue);

        maTable.tableHeader.addColumn(new pdfTableColumn("Titre", predefinedAlignment.csRight, 170));
        maTable.tableHeader.addColumn(new pdfTableColumn("Auteur", predefinedAlignment.csCenter,110));
        maTable.tableHeader.addColumn(new pdfTableColumn("Numéro, predefinedAlignment.csLeft, 90));
        maTable.tableHeader.addColumn(new pdfTableColumn("Email", predefinedAlignment.csLeft, 150));

        pdfTableRow maLigne = maTable.createRow();
        maLigne[0].columnValue = "Dressez la Jtable";
        maLigne[1].columnValue = "Frédéric Mazué;
        maLigne[2].columnValue = "Programmez n°67;
        maLigne[3].columnValue = "fmazue@programmez.com";
        maTable.addRow(maLigne);
        maLigne = maTable.createRow();
        maLigne[0].columnValue = "Synchronisation de données;
        maLigne[1].columnValue = "Christian PEYRUSSE";
        maLigne[2].columnValue = "Programmez n°67;
        maLigne[3].columnValue = "-";
        maTable.addRow(maLigne);
        maLigne = maTable.createRow();
        maLigne[0].columnValue = "MySQL 5 et les procédures stockées;
        maLigne[1].columnValue = "Xavier Leclercq";
        maLigne[2].columnValue = "Programmez n°67;
        maLigne[3].columnValue = "Xavier.Leclercq@programmez.com";
        maTable.addRow(maLigne);

        maTable.tableHeaderStyle = new pdfTableRowStyle(predefinedFont.csCourierBoldOblique, 10, new pdfColor(predefinedColor.csBlack), new pdfColor(predefinedColor.csLightCyan));
        maTable.rowStyle = new pdfTableRowStyle(predefinedFont.csCourier, 7, new pdfColor(predefinedColor.csBlack), new pdfColor(predefinedColor.csWhite));
        maTable.alternateRowStyle = new pdfTableRowStyle(predefinedFont.csCourier, 7, new pdfColor(predefinedColor.csBlack), new pdfColor(predefinedColor.csLightYellow));
        maTable.cellpadding = 10;
        maPremierePage.addTable(maTable, 20, 600);
        monDoc.createPDF(@"c:\table_C_sharp_pdf.pdf");
        maTable = null;
    }
}
```

ItextSharp

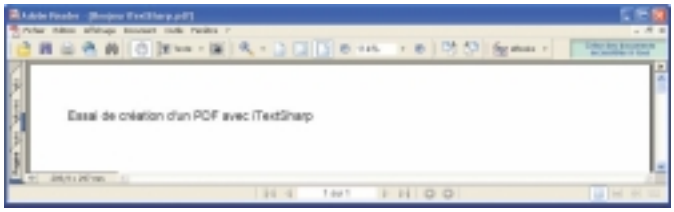
Il s'agit probablement de la bibliothèque la plus puissante, car la plus riche en fonctions. Elle provient d'un portage de la bibliothèque Open Source Itext Java. C'est également un logiciel Open Source, qui est accompagné d'une documentation (on-line et toujours en anglais) bien faite.

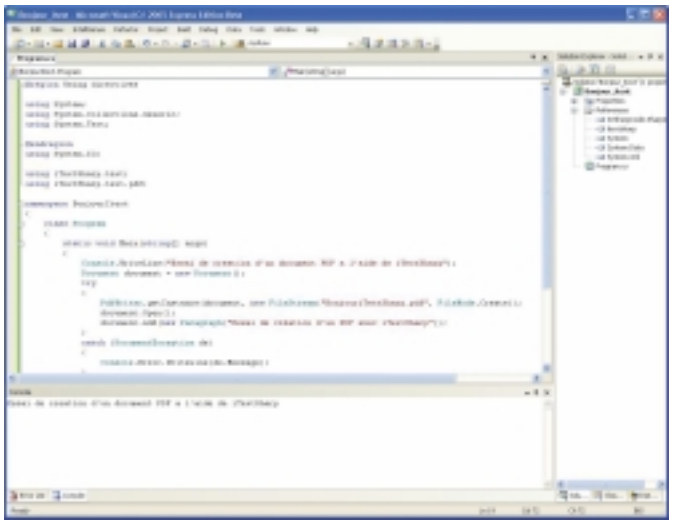


Un seul hic : à l'exécution du code ci-dessous C# 2005 Express déclenchera une exception. La raison ? Une erreur de dépendance, car la bibliothèque ICSharpCode.SharpZipLib.dll n'est pas installée. Attention, la version majeure/mineure de son "strong name" devra être 0.5. Or la dernière version officielle est la 0.81. Pour résoudre ce problème soit vous devez recompiler ItextSharp, soit vous devez modifier le fichier AssemblyInfo.cs pour qu'il corresponde (en téléchargeant les sources de ICSharpCode.SharpZipLib.dll), soit le plus simple, vous téléchargerez sur Internet une version 0.5...

```
using System.IO;
using iTextSharp.text;
using iTextSharp.text.pdf;

namespace BonjourItext
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Essai de crÃ©ation d'un document PDF Ã  l'aide de iTextShar);
            Document document = new Document();
            try
            {
                PdfWriter.getInstance(document, new FileStream("BonjouriTextSharp.pdf", FileMode.Create));
```

```
            document.Open();
            document.Add(new Paragraph("Essai de crÃ©ation d'un PDF
avec iTextSharp"));
        }
        catch (DocumentException de)
        {
            Console.Error.WriteLine(de.Message);
        }
        catch (IOException ioe)
        {
            Console.Error.WriteLine(ioe.Message);
        }
        document.Close();
    }
  }
}
```





Remarque : attention au conflit potentiel

Le script installGAC.bat installera ICSharpCode.SharpZipLib.dll dans le GAC

```
@echo off
cd bin
echo installing ICSharpCode.SharpZipLib.dll into the GAC
gacutil /i ICSharpCode.SharpZipLib.dll
cd ..
```

Mais attention : si vous avez installé mono, l'espace GAC de ce dernier risque d'entrer en conflit ; vérifiez bien où, au final, la bibliothèque sera installée...

Le mot de la fin

C# 2005 Express est tout à fait opérationnel, quoiqu'il s'agisse encore d'une version non définitive. L'outil est léger et agréable à utiliser, surtout si votre machine n'est pas un mastodonte de puissance.

Au sujet des DLL Open Source consacrées à la génération de PDF on peut regretter leurs dépendances au système Windows. Heureusement sous Linux il existe bien d'autres possibilités pour un programmeur. Et si jamais cela vous bloque, vous pourrez aisément invoquer un composant d'un serveur d'objets intermédiaire : PHP, Python et C++ sont bien fournis en extensions Open Source capables de générer du PDF.

Liste des liens :

Aspose PDF : http://www.aspose.com/Products/Aspose.Pdf/
PDF-Writer.NET : http://www.dbautotrack.com/products/pdfwriter.html
DynamicPDF MergerForNET : http://dynamicpdf.com/Products/MergerForNET/
Report.NET : http://report.sourceforge.net/
Téléchargement Report.NET : http://sourceforge.net/project/showfiles.php?group_id=58374
SharpPDF : http://sharppdf.sourceforge.net/
Itextshap : http://itextsharp.sourceforge.net/index.html
Téléchargement Itextshap : http://sourceforge.net/projects/itextsharp/
Téléchargement d'une version 0.5.00 d'ICSharpCode.SharpZipLib.dll : http://www.csharpfr.com/code.aspx?id=11506
SharpZipLib : http://www.icsharpcode.net/OpenSource/SharpZipLib/Download.aspx

Xavier Leclercq - *xavier.leclercq@programmez.com*

Abonnez-vous
et recevez 3 numéros GRATUITS
45 € au lieu de 65€ (prix de vente au numéro)
soit 20 € d'économie, équivalent de 3 numéros de Programmez!

Le copier-coller et le glisser-déposer sous Java 2

SOURCES DE L'ARTICLE
WWW.PROGRAMMEZ.COM

Délaissées aux origines de Java, ces fonctionnalités incontournables de toute application au design moderne, trouvent enfin une implémentation satisfaisante avec la JDK 1.4. Découvrons cela par la pratique.

Sun Microsystem déclare volontiers que Java est le langage de toutes les situations. Peut-être. Toujours est-il que la genèse de Java a été assez compliquée. Avec, au départ, un rôle dévolu aux réseaux et aux grille-pains, Java n'arrive sur les PC que sous l'impulsion de Netscape et conserve son orientation réseau. On ne s'étonnera donc pas que Java et sa JDK 1.0 n'aient rien proposé en matière d'impression, de copier-coller et de glisser-déposer. Des implémentations rudimentaires apparurent avec la JDK 1.1. En ce qui concerne l'impression, les choses deviennent satisfaisantes avec Java 2. Pour le reste, ce n'est qu'avec la JDK 1.4 que le copier-coller et le glisser-déposer, sont bien pris en charge. Par bien, nous entendons d'une part qu'une application Java peut placer des données dans le presse-papiers de la plate-forme hôte et que ces données sont récupérables, non seulement par une autre application Java, mais aussi par une application native, et d'autre part qu'il est possible de mettre en place un glisser-déposer sans s'arracher les cheveux ni écrire des kilomètres de code.

Le presse-papier et le texte

Commençons pas nous faire la main avec le plus facile : placer du texte dans le presse-papier. Créer un langage portable n'est pas simple. Les presse-papiers systèmes de Windows, Mac, ou UNIX/Linux n'ont rien de commun. Évoluant dans le monde d'UNIX et bien placés pour savoir qu'il n'y a pas, par exemple, de standard pour le transfert d'image sous X-Window, les concepteurs de Java ont d'abord fait le choix de se limiter à un plus petit dénominateur commun. Voilà pourquoi la JDK 1.1 ne permettait de placer que du texte dans le presse-papier système. En outre, sous UNIX la plupart des applications implémentent leur propre presse-papiers. La JVM ne fait pas exception à la règle. Ainsi la classe Clipboard du paquet java.awt.datatransfer encapsule, soit un presse papier local à l'application, soit le presse papier système. Lorsque le presse-papier est local, il est possible d'y transférer n'importe quel objet, mais nous n'étudierons pas ce cas dont l'intérêt est limité. Lorsque nous voulons travailler avec le presse-papier système, nous devons obtenir un singleton de la classe Clipboard comme ceci :

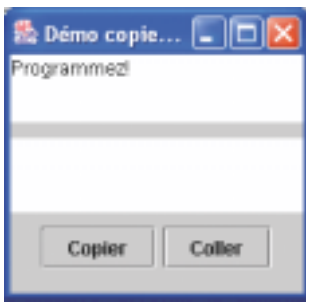


Fig.1 : Notre premier programme d'exemple sait placer ou récupérer du texte dans le presse-papier.

```
Toolkit.getDefaultToolkit().getSystemClipboard();
```

La classe Clipboard est censée travailler en collaboration avec d'autres classes et interfaces du même paquet :

Classe/Interface	Description
Interface Transferable	Tout objet qui veut pouvoir transiter par le presse-papier doit implémenter cette interface. Il en est de même pour le glisser- déposer qui travaille aussi avec cette interface.
Interface ClipboardOwner	Une classe qui veut être informée d'une modification externe du contenu du presse papier doit implémenter cette interface.
Classe DataFlavor	Cette classe permet de connaître le type de données déposées dans le presse-papier.
Classe StringSelection	La seule classe implémentant l'interface transférable dans JDK 1.1 et permettant le transit de texte dans le presse papier.
Classe UnsupportedFlavorException	L'exception qui est levée par un objet Transferable qui ne peut fournir les données dans le format demandé.

Faisons le point. L'existence de l'interface Transferable ne doit pas nous tromper si nous travaillons avec une JDK antérieure à 1.4. Dans ce cas, seul le texte peut passer dans le presse-papier système et implémenter Transferable ne présente d'intérêt que pour un presse-papier local. Si l'on tient à travailler avec le presse papier système, il faut définir un type Mime et convertir les données en texte, ce qui est très lourd. Moyennant quoi une autre application Java qui connaît le type Mime peut récupérer les données dans le presse-papier. Mais une application native travaillant avec les formats définis par la plate-forme hôte ne pourra pas le faire. Voilà pourquoi la JDK n'offre à l'origine qu'une classe implémentant Transferable, la classe StringSelection dédiée au transit du texte. Le bon côté de cette limitation est l'extrême simplicité comme vous pouvez le constater en examinant le code de notre premier exemple (Fig.1) (Les sources complets figurent sur le Cd-Rom accompagnant le magazine, ou si ce n'est pas le cas, sur le site officiel www.programmez.com). Placer du texte est enfantin :

```
Clipboard systemClipboard =
Toolkit.getDefaultToolkit().getSystemClipboard();
String text = jTextArea1.getText();
StringSelection selection = new StringSelection(text);
systemClipboard.setContents(selection, null);
```

Et le récupérer n'est pas beaucoup plus compliqué :

```
Transferable contents = systemClipboard.getContents(this);

try
{
text = (String)(contents.getTransferData
(DataFlavor.stringFlavor));
```

```
}
catch(Exception ex)
{
      System.out.println("Erreur!");
}
```

Le texte à placer dans le presse papier est enveloppé dans la classe StringSelection et devient donc "Transferable". On note que la méthode getContents de la classe Clipboard (Mais pourquoi diable cette méthode requiert-elle un objet en paramètre ? Mystère et boule de gomme) retourne un Transferable qui a la charge de nous fournir des données "au parfum," c'est-à-dire en accord avec le paramètre DataFlavor reçu par sa méthode getTransfertData. Si le Transferable en est incapable, il lève l'exception mentionnée plus haut. Le programme d'exemple permet donc de placer du texte dans le presse-papier et de le relire, mais aussi de lire le texte en provenance d'une application native. Tout ceci fonctionne fort bien... sous Windows, mais il n'est pas exclu que vous rencontriez des problèmes sous Linux par exemple, toujours à cause des particularités de X-Window. Bien entendu, de Java à Java cela va bien, mais il en peut en être autrement avec les applications natives. En effet, sous X-Window il y a deux presse-papiers : le primary et le selection. Le premier est en principe utilisé par le copier-coller des applications natives et le second correspond à la sélection directe, par exemple dans le terminal, et à l'utilisation du bouton du milieu de la souris. Ce dernier n'est bien pris en charge que par la JDK 1.4. Et bien sûr si l'application native ne connaît que son propre presse-papier il n'y a rien à faire.

Surveiller le presse-papier

Une application qui a placé des données dans le presse papier peut souhaiter être avertie, lorsque les données sont écrasées par une autre application. Dans ce cas, l'application se fait connaître auprès de la JVM comme détentrice du presse-papier. Cela consiste à implémenter l'interface ClipboardOwner dans une classe anodine, dont le seul rôle est de fournir la méthode qui sera invoquée par la JVM lors de la modification du presse-papier. Par exemple :

```
class MonClipboardOwner implements ClipboardOwner {

  JFrame frame;

  public MonClipboardOwner(JFrame parent) {
    frame = parent;
  }

  public void lostOwnership(Clipboard clipboard,
Transferable contents) {
    JOptionPane.showMessageDialog(frame,
"le contenu du presse-papier a été modifié");
  }
}
```

et on s'enregistre tout simplement au moment où l'on place le texte :

```
clipboardOwner = new MonClipboardOwner(this);
```

```
StringSelection selection = new StringSelection(text);
systemClipboard.setContents(selection, clipboardOwner);
```

L'application détentrice n'est avertie qu'à la première modification de contenu et pas ensuite, sauf si elle s'enregistre de nouveau, ce qui est logique.

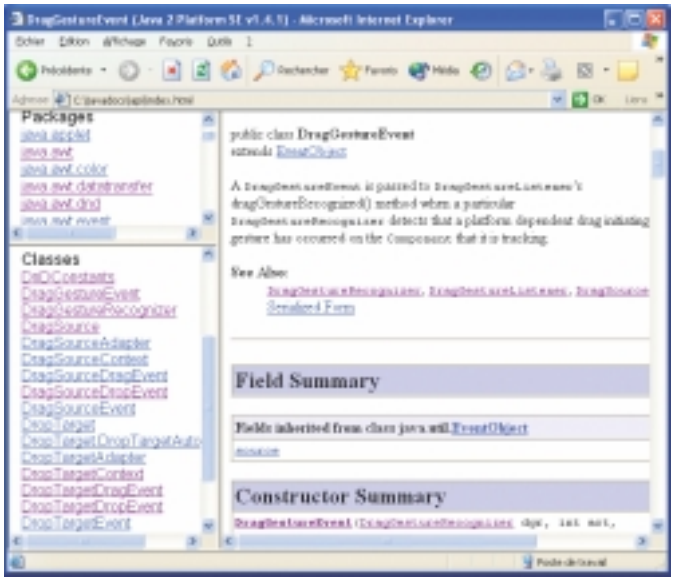


Fig.2 : Avant JDK 1.4 le programmeur devait manipuler beaucoup de classes pour mettre en place un glisser-déposer.

Le glisser- déposer

Ce mécanisme est encapsulé par les classes du paquet java.awt.dnd. Un coup d'oeil sur les classes (Fig.2) a de quoi décourager les meilleures volontés dès que l'on comprend que le programmeur doit s'occuper de tout, événement souris et autres, non pas une fois, mais deux. Une fois pour le composant source, une fois pour le composant cible. C'est pourquoi les améliorations de la JDK 1.4 sont plus que bienvenues. Swing compte désormais une classe TransferHandler qui encapsule le mécanisme de glisser- déposer. Mieux que cela, quelques composants Swing comportent un objet TransferHandler par défaut. En voici la liste :

Composant	Supporte le Glisser	Supporte le Déposer
JColorChooser	Oui	Oui
JEditorPane	Oui	Oui
JFileChooser	Oui	Non
JFormattedTextField	Oui	Oui
JList	Oui	Non
JPasswordField	Non	Oui
JTable	Oui	Non
JTextArea	Oui	Oui
JTextField	Oui	Oui
JTextPane	Oui	Oui
JTree	Oui	Non

Pour ces classes, il suffit d'invoquer la méthode setDragEnabled pour enclencher le mécanisme. Cette méthode est invoquée seulement depuis les composants source. Ainsi, le troisième programme d'exemple

Fig.3 : Le texte du label peut glisser dans la zone de texte du milieu.

(Fig.3) met en place un glisser-déposer entre deux composants JTextArea. Nous avons un comportement par défaut, à savoir, que la partie de texte sélectionnée est coupée du composant source et copiée dans le composant cible. Pour éviter la coupure dans le composant source, il faut maintenir la touche Crtl enfoncée pendant l'opération. Mais il est possible d'influencer ce comportement par programmation

Modifier un comportement de Glisser-Déposer

On peut souhaiter modifier le comportement décrit plus haut, ou même, pourquoi pas, doter un composant du Glisser-Déposer, même si ce composant ne le supporte pas à la base. C'est même très facile. Nous savons que la classe TransferHandler est la clé, et il se trouve que le composant racine Swing JComponent comporte une méthode setTransferHandler. Nous pouvons donc installer le mécanisme sur n'importe quel composant Swing. Instancier un objet TransferHandler ne présente aucune difficulté. Son constructeur reçoit une propriété sous la forme d'une chaîne. Il s'agit bien évidemment d'une propriété de Bean, car tout composant Swing est avant tout un Bean. Si besoin, vous pouvez deviner les noms des propriétés à partir des accesseurs de la classe. Il suffit de supprimer le préfixe des accesseurs et de mettre en minuscule la première lettre de ce qui reste. Par exemple, soient les accesseurs getText et setText, le nom de la propriété sous-jacente est tout simplement text.

Le point plus difficile est que le comportement ne peut être défini lors de la construction de l'objet. En effet, la méthode exportAsDrag du transferHandler qui enclenche effectivement le mécanisme de glisser demande en paramètre l'événement concerné, ce qui n'est pas sans logique. Nous devons donc invoquer la méthode susmentionnée dans un écouteur d'événement de souris, comme le montre l'extrait de code ci-contre (Encadré 1) Cette méthode reçoit en outre un type d'action, ce que nous utilisons pour copier le contenu de la source sans le couper. Nous avons appliqué cette technique dans le quatrième exemple (classe DemoDnD2), (Fig.4) à la fois sur la zone supérieure et sur le label. Le texte de ce dernier peut maintenant glisser dans la zone de texte du milieu. Faisons deux remarques. D'abord, nous pouvons appliquer la technique à toute propriété, par exemple 'foreground'. N'omettez pas d'expérimenter. Ensuite, en ce qui concerne la zone de texte de notre exemple, il y a un petit revers de médaille. Il y a un effet de bord sur la gestion de la souris, puisque nous avons changé l'écouteur. Ainsi, nous ne pouvons plus sélectionner de texte en déplaçant le caret, mais seulement depuis le clavier. Il faudrait enrichir la gestion des événements souris pour y remédier, ce que nous ne faisons pas ici.

Fig.4 : Glisser-déposer d'une image

Faire glisser une image.

Le fin du fin consiste à dériver la classe TransferHandler, ce que nous faisons dans le dernier programme d'exemple, afin de pouvoir travailler avec une image. Celle-ci, intégrée dans un JLabel peut glisser vers une autre application Java, ou une application native comme l'utilitaire Paint de Windows. Le bouton 'Copier' place l'image dans le presse-papiers, et ainsi de suite. L'image en question doit être définie au lancement de l'application. Par exemple :

```
java programmez.fred.dnd.DemoDnD3 c:/smile.gif
```

Sous Linux, le programme fonctionne très bien de Java vers Java, mais nous retrouvons le problème des presse-papiers privés des applications natives. La dérivation de TransferHandler présente deux points intéressants. Premier point, notre classe crée un Transferable à la volée en redéfinissant au passage quelques méthodes. N'oublions pas que tout objet destiné au presse-papier doit être un Transferable. Deuxième point, la méthode getTransferDataFlavors de notre Transferable, retourne un tableau de DataFlavor à un élément qui est un DataFlavor.imageFlavor. Nous profitons en cela des fonctionnalités de la JDK 1.4 qui convertit les images en données compréhensibles par le presse-papiers système, d'où une méthode getTransferData réduite à sa plus simple expression :-) L'extrait de code ci-contre (Encadré 2) peut être adapté à n'importe quel type devant transiter par le presse-papiers, ou glisser d'un composant à un autre. Une dernière remarque en ce qui concerne la gestion du bouton 'Copier' de l'exemple. Dans le gestionnaire d'événement de ce bouton, nous commençons par obtenir un objet 'Action de copie' du TransferHandler, puis nous simulons le déclenchement de l'événement de copie, en faisant croire que ce déclenchement émane du label lui-même, et le tour est joué. Même remarque en ce qui concerne le bouton 'Coller'. Nous n'entrons pas dans le détail ici, car le bricolage des événements Java mériterait un article à lui seul. Une prochaine fois peut-être... À bientôt.

Encadré 1

Code personnalisant un Glisser-Déposer
(Extrait du quatrième programme d'exemple, classe DemoDnD2)

```
// Reste requis même pour un
// comportement personnalisé
jTextArea1.setDragEnabled(true);

// Mécanisme de transfert personnalisé
// pour la zone de texte
jTextArea1.setTransferHandler(new TransferHandler("selectedText"
/* ou bien "text" */));
MouseListener listener1 = new MouseAdapter() {
  public void mousePressed(MouseEvent e) {
    JComponent c = (JComponent) e.getSource();
    TransferHandler th = c.getTransferHandler();
    th.exportAsDrag(c, e, TransferHandler.COPY);
  }
};
jTextArea1.addMouseListener(listener1);

// Mécanisme de transfert installé pour le label
```

```
jLabel1.setTransferHandler(new TransferHandler("text"));

MouseListener listener2 = new MouseAdapter() {
  public void mousePressed(MouseEvent e) {
    JComponent c = (JComponent) e.getSource();
    TransferHandler th = c.getTransferHandler();
    th.exportAsDrag(c, e, TransferHandler.COPY);
  }
};
jLabel1.addMouseListener(listener2);
```

Encadré 2

```
class ImageSelection
    extends TransferHandler {
  private static final DataFlavor flavors[] = {
      DataFlavor.imageFlavor};
  // L'image doit rester en place
  // dans le composant de départ
  public int getSourceActions(JComponent c) {
    return TransferHandler.COPY;
  }
  // Vérifier si on est "au parfum"
  public boolean canImport(JComponent comp, DataFlavor flavor[]) {
    if (! (comp instanceof JLabel)) {
      return false;
    }
    for (int i = 0, n = flavor.length; i < n; i++) {
      for (int j = 0, m = flavors.length; j < m; j++) {
        if (flavor[i].equals(flavors[j])) {
          return true;
        }
      }
    }
    return false;
  }
  public Transferable createTransferable(JComponent comp) {
    if (comp instanceof JLabel) {
      JLabel label = (JLabel) comp;
      Icon icon = label.getIcon();
      if (icon instanceof ImageIcon) {
        final Image image = ( (ImageIcon) icon).getImage();
        final JLabel source = label;
        Transferable transferable =
            new Transferable() {
          public Object getTransferData(DataFlavor flavor) {
            if (isDataFlavorSupported(flavor)) {
              return image;
            }
            return null;
          }
          public DataFlavor[] getTransferDataFlavors() {
            return flavors;
          }
          public boolean isDataFlavorSupported(DataFlavor flavor) {
            return flavor.equals(DataFlavor.imageFlavor);
          }
        };
        return transferable;
      }
    }
    return null;
  }
  public boolean importData(JComponent comp, Transferable t) {
    if (comp instanceof JLabel) {
      JLabel label = (JLabel) comp;
      if (t.isDataFlavorSupported(flavors[0])) {
        try {
          Image image = (Image) t.getTransferData(flavors[0]);
          ImageIcon icon = new ImageIcon(image);
          label.setIcon
              (icon);
          return true;
        }
        catch (UnsupportedFlavorException ue) {
          System.out.println("Erreur");
          System.out.println("Votre parfum sent mauvais");
        }
        catch (IOException ioe) {
          System.out.println("Exception IOException levée");
        }
      }
    }
    return false;
  }
}
```

Frédéric Mazué - *fmazue@programmez.com*