

2 CD-Roms **Windows XP SP2 + Pervasive**

Programmez!

LE MAGAZINE DU DÉVELOPPEMENT

WEB SERVICES

Mensuel
Novembre 2004
N°69 • 5,95 €

Pouvez-vous y échapper ?

TRUCS & ASTUCES

Débogage Linux,
Cookies sur PHP etc...

SGBD

Choisir sa base de données

Test : JBuilder 2005



XML

Sérialisation
sous Mono
et .net

OPEN SOURCE

C-JDBC, une alternative
de clustering

JAVA

Créer l'événement

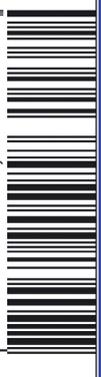
WINDOWS

La programmation
du Shell

C/C++

Optimisez
vos codes sources

M 04319 - 69 - F: 5,95 €



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

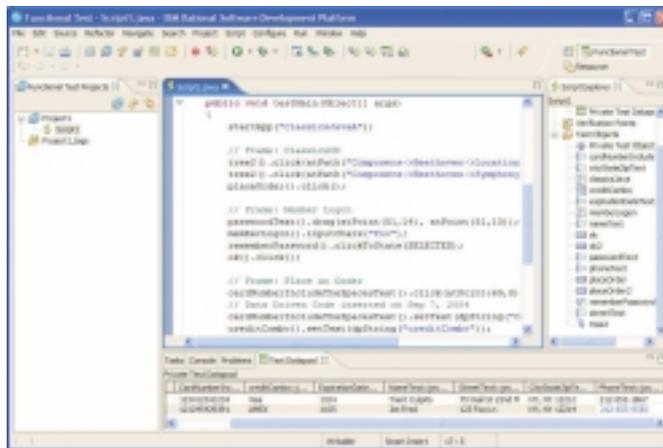
Atlantic : la nouvelle arme d'IBM Rational

Après un an d'effort, IBM a réussi son pari avec le rachat de Rational. Les outils de développement étant devenus pour le Grand Bleu un marché stratégique, il fallait pour concrétiser cela une offre forte et viable, face à une concurrence vivace sur le secteur du cycle de vie.

Connue sous le nom de code Atlantic, cette suite dédiée au cycle de vie reprend et étend considérablement les fonctions que l'on avait avec Rational Suite. Par rapport à Rational Suite, il s'agissait de simplifier l'approche du cycle de vie et d'avoir une meilleure productivité et intégration entre les différents outils. Elle reprend le concept déjà éprouvé par Borland. L'ensemble des outils " repose " sur une couche basse technologie / technique commune : Eclipse (IBM réaffirme ainsi son soutien actif à cet outil). Ce nouvel ensemble se dédie à Java / J2EE, mais aussi à la plate-forme .NET (via Visual Studio), même si ce n'est pas une priorité. Atlantic prend en charge de bout en bout le projet et son cycle : de l'élaboration à la maintenance. Outre la refonte de nombreux outils, trois nouveautés sont mises en avant : la modélisation (UML 2), l'architecture et les tests. Chaque axe possède son outil dédié (voir encadré). Outre l'architecte qui prend une place de plus en plus importante, IBM mise aussi sur le chargé de déploiement. Un rôle vital dans les projets complexes. Sur UML2, la spécification n'est pas encore entièrement finalisée, cela devrait être le cas dans quelques semaines.

Eclipse en noyau

En soi, Atlantic se présente ainsi : Eclipse comme base, puis un ensemble de standards et de plug-ins. IBM met en exergue le rôle et l'importance des standards : EMF, Hyades (standard de tests)... Il est cependant possible de construire sur un Eclipse " classique " une solution de cycle de vie avec des plug-ins projets et ceux d'IBM. Dans



cette situation, IBM doit apporter un réel " plus produit ". L'éditeur met en avant l'intégration des outils, la qualité et la cohérence de la plate-forme. Pour faire communiquer tout cela et échanger des données homogènes entre les différentes applications, on dispose d'un unique référentiel, basé sur des fichiers XML. La présence d'un référentiel est indispensable pour assurer la cohérence du cycle de vie. Au-delà des développeurs, testeurs, architectes, IBM a annoncé la création du Ready for IBM Rational. Cette nouvelle communauté doit aider les éditeurs à rendre compatibles leurs outils avec Atlantic, notamment via des tests. On aura droit à un beau logo de compatibilité sur les outils tiers qui auront réussi l'examen. De plus, ce programme n'est pas payant. En théorie, il sera possible de panacher Atlantic avec des outils tiers compatibles. On peut, grâce à la modularité apportée par la couche Eclipse, choisir uniquement les modules réellement utiles pour tel ou tel projet. Qu'en est-il des évolutions d'Eclipse ? IBM assure que la nouvelle plate-forme intégrera automatiquement toute nouvelle version.

Services & MDA

Les environnements de travail sont personnalisables, via les profils. Il est possible de combiner plusieurs méthodes : RUP, métriques, règles de codage, etc. Très pratique dans les projets complexes. Un système de notification sera disponible. IBM suit aussi la tendance actuelle avec JSF (même si Struts est toujours disponible). On peut même mixer

les deux. À partir d'un modèle UML on peut générer une structure JSF (via les fonctions de transformations). Très clairement, IBM tend à utiliser un modèle d'abstraction de haut niveau de type MDA. Avec l'importance prise par la modélisation et notamment UML, IBM a multiplié les possibilités : différentes vues d'un modèle, visualisation du code directement dans le modèle. Les Web Services (et encore moins SOA qui est totalement intégré) ne sont pas oubliés.

Là aussi, la procédure de création et de consommation a été simplifiée au maximum. Qui dit développement en équipe, dit aussi fonction de KM et de bibliothécaire. Le standard RAS (Reusable Asset Specification) est implémenté. Il s'agit de créer, qualifier, spécifier, archiver et rechercher des actifs logiciels que l'on peut partager et réutiliser. RAS s'appuie sur UML pour se décrire lui-même. L'ensemble de la gamme Atlantic sera disponible durant le mois de décembre 2004. Les tarifs n'ont pas encore été communiqués.

■ F.T.

Les principaux produits de l'offre Atlantic

IBM Rational Portfolio Manager : pour gérer le cycle de vie d'un portefeuille de projets. On gère les projets et les développements déjà réalisés comme un métier de l'entreprise.

IBM Rational Software Architect : outil dédié aux architectes. Il doit les aider à concevoir et à développer les projets en passant par l'aspect architecture et la qualité logicielle.

IBM Rational Software Modeler : outil de modélisation et de conception visuelle UML 2.0. Permet de documenter et communiquer les différentes vues d'un système ou d'une application. IBM n'abandonne pas pour autant ses autres outils du même genre (Rose et XDE).

IBM Rational Manual tester : outil de test manuel, surtout dédié aux analystes métiers et testeurs. D'une prise en main simplifiée, il permet de créer des scénarios de tests manuels. Il permet aussi de réaliser des tests manuels utilisables dans des équipes distribuées.

IBM Rational Functional tester : outil pour automatiser le test fonctionnel et de non régression. Supporte aussi les applications .NET. Se dédie aux développeurs et testeurs.

IBM Rational Performance tester : outil de test de charge et de performance afin de mesurer la bonne tenue des applications web. Incluant aussi des fonctions de mise à jour, évaluation, exécutions des tests avec analyses des résultats.

JBuilder 2005, un nouveau service pour un nouveau café

La sortie de la nouvelle JDK de Sun est imminente. Les outils de développement doivent relever le défi lancé par les améliorations du langage. Nous regardons aujourd'hui ce qu'il en est avec JBuilder 2005 de Borland, version bêta.

Java 5.0 alias Jdk1.5 alias Tiger ne va pas tarder à arriver sur vos plates-formes, et avec lui une nouvelle syntaxe pour la programmation générique en Java (ce n'est pas trop tôt...). Parallèlement, les Java Server Faces ou JSF en sont déjà à la version 1.1, concurrence de .Net oblige... Dans ce contexte, les outils de développement doivent suivre le rythme et s'adapter. Voyons comment s'en tire le JBuilder 2005 que nous avons en main.

Dès le premier lancement de la bêta, on s'aperçoit que l'interface a été complètement remaniée depuis JBuilder 9 (nous n'avons pas testé JBuilderX). (Fig.1) JBuilder reste JBuilder mais les influences d'Eclipse et de Visual Studio .Net se font nettement sentir. C'est une bonne chose. Nous trouvons l'outil plus agréable visuellement et plus ergonomique dans son ensemble. L'EDI semble avoir été totalement réécrit, car il répond sensiblement plus vite et ce gain n'est vraisemblablement pas dû à la JDK 1.4.2 qui le fait tourner. La palette de composants a été déplacée et se situe maintenant verticalement tout près du concepteur visuel. (Fig.2) De ce fait, ce dernier nous paraît plus homogène et agréable à l'utilisation. L'éditeur de code a été revu et amélioré. Nous avons apprécié la nouvelle gouttière qui reçoit un menu contextuel et depuis laquelle il est possible, à l'instar de Visual Studio, d'ouvrir ou replier classes et méthodes. C'est une excellente chose de ne plus devoir manipuler des sources d'une

longueur interminable. Les fonctionnalités d'achèvement, toujours excellentes de code, réagissent encore un peu plus rapidement. Une grande nouveauté est l'éditeur HTML et son concepteur visuel qui permet de construire des formulaires, mais aussi de déposer n'importe quelle balise d'un coup de souris.

JBuilder met un tigre dans son moteur

Notre version bêta vient sans Jdk 1.5. Nous avons donc téléchargé la JDK 1.5 Release Candidate séparément, et nous avons configuré l'EDI en conséquence. Après quoi, nous avons pu travailler sans problème avec les nouveautés du langage qui sont toutes reconnues. De plus, l'éditeur de code apporte une fonctionnalité de refactoring qui permet de transformer du code Java à l'ancienne en nouveau Java, en réécrivant les boucles, ou en introdui-

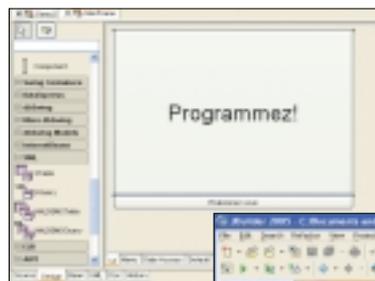


Fig. 2 : Le nouveau concepteur visuel de JBuilder

Fig. 3 : Le concepteur JSF de JBuilder

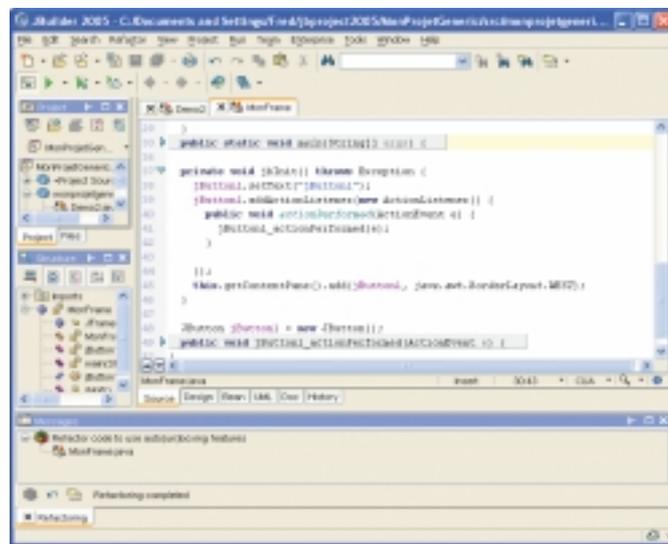


Fig.1 : Vue d'ensemble de JBuilder 2005

sant la généricité sur Vector et autres List. Bien entendu le débogage depuis l'EDI sous JDK 1.5 est possible.

Java Server Faces

JBuilder 2005 se devait d'offrir un concepteur visuel pour les Java Server Faces. Nous avons trouvé que l'outil s'intégrait bien à l'ensemble. Il est constitué de deux éléments principaux. (Fig.3) D'abord, le concepteur de flux qui présente un diagramme montrant comment les JSP se connectent l'une avec l'autre, en accord avec les règles de navigation. Si une JSP n'existe pas encore vous êtes prévenus et vous avez la possibilité de la créer immédiatement. Ensuite,

le concepteur de navigation qui visualise une règle de navigation avec ses aiguillages. Depuis ce concepteur également vous pouvez créer une JSP qui n'existerait pas pour la vue concernée. Cerise sur le gâteau, les deux concepteurs acceptent les JSP du projet par glisser-déposer.

Et plus si affinité

En plus des points forts que sont Jdk 1.5 et JSF, JBuilder comporte d'autres nouveautés. Nous avons noté, entre autres, un nouveau profiler EJB intégré à l'EDI, l'intégration des principaux systèmes de contrôle de version, dont le nouveau, Subversion, et l'intégration de CaliberRM. En outre, le serveur d'application BES qui accompagne l'édition entreprise bénéficie de nombreuses améliorations. Mais finalement, est-ce que cela vaut le coup de passer à JBuilder 2005 ? Ne serait-ce que pour le support de Jdk 1.5 et JSF, mais aussi par l'excellente tenue d'ensemble et son efficacité, nous pensons que oui.

■ Frédéric Mazué

fmazue@programmez.com

Crystal Reports 10 simplifie la conception de rapports

N'en déplaie aux adeptes de l'écriture de lignes de code, la nouvelle mouture de Crystal Reports met d'abord l'accent sur des outils qui limitent les besoins en programmation et en administration. Sans toutefois bouleverser les fondements de ce produit, bien connu des développeurs ...

Cystal Reports est probablement l'un des outils de création et de diffusion de rapports à un grand nombre d'utilisateurs (reporting de masse) les plus connus des développeurs. Sortie en janvier dernier, la version 10 est la première mouture signée Business Object, qui a acquis Crystal Reports il y a un an environ. Cette version, loin de révolutionner le produit, apporte des améliorations qui simplifient l'administration et la diffusion de rapports.

Une meilleure gestion des environnements J2EE

Livré en standard (mais en version bridée) avec de nombreux outils de développement du marché, dont WorkShop de BEA, C# et Jbuilder de Borland, ou encore Visual Studio de Microsoft, Crystal Reports a surtout étoffé dans cette dixième édition sa gamme d'outils java, afin de faciliter l'exploitation de rapports par des environnements J2EE. Au programme, l'apparition d'un moteur java et une bibliothèque de balises JSP (JSP tag library) qui permettent, sur simple clic de souris, d'intégrer les rapports aux applications conçues pour les serveurs d'applications WebLogic (BEA), Websphere (IBM et Tomcat (open-source), ou encore, d'encapsuler un rapport dans un document JSP. Des fonctionnalités équivalentes existaient déjà dans la version 9 pour .NET avec

des modules " Merger " pour ASP.NET, Web Services et Windows (client lourd).

Une plus grande simplicité d'administration

Également présent dans la version 9, le référentiel central a été enrichi d'outils qui facilitent la maintenance des gabarits et la recherche de rapports. Ce référentiel peut également servir de socle au stockage de commandes SQL, de fichiers bitmap, ou encore de fonctions ou commandes personnalisées que le développeur veut pouvoir réutiliser. Côté présentation, le module de création permet enfin de formater plusieurs cellules en même temps et ainsi, éviter les tâches répétitives pour appliquer des éléments de style (police de caractère, couleur de bordure, etc.) aux rapports. Une fonction d'audit établit également des statistiques sur le taux de consultation par utilisateur sur lesquelles l'administrateur peut se baser, pour ensuite optimiser la diffusion, la gestion des droits d'accès, etc.

Des exports mieux contrôlés vers Office de Microsoft

Dans les évolutions significatives, il faut encore noter l'ajout d'une fonction pour imprimer des rapports au format DHTML et la présence d'un plus grand nombre de contrôles pour permettre aux utilisateurs de personnaliser la restitu-

Dotée d'un portail, la version 10 de Crystal Reports permet de réaliser simplement des vues personnalisées en fonction du profil de l'utilisateur.



tion des rapports dans la suite bureautique de Microsoft, Office System. " L'amélioration des exports vers Excel, mais également PowerPoint et même Word, constitue un des gros efforts réalisés avec cette version, estime Walid Farad, Responsable Produit Crystal chez Business Objects qui ajoute : en fait, nous avons, avant tout, cherché à simplifier au maximum la publication des états, en fournissant aux développeurs des assistants afin de limiter le plus possible l'écriture de lignes de code ".

Une ouverture sur des sources de données qui ne cessent de s'enrichir

Multi-plate-forme en déploiement, Crystal Reports l'est également en source de données. Aujourd'hui, l'éditeur compte 35 connecteurs natifs à son catalogue, dont ODBC, OLE DB et JDBC pour s'interfacer avec les plus grandes bases de données du marché, mais également des connecteurs pour les Javabeans, ACT!, les dossiers Exchange, les event log NT, etc. Même si l'interfaçage ADO.NET est présent, on s'étonnera toutefois de l'absence de

support direct de la classe .NET "SQLClient".

La version 10 introduit également les " Crystal business views " qui facilitent l'intégration de données provenant de sources disparates et hétérogènes, en constituant des " vues logiques " qui masquent les différences de structure et de résolution entre les plates-formes.

Malgré toutes ces améliorations, il n'en reste pas moins que, si Crystal Reports demeure de loin l'outil le plus complet de sa catégorie, la plate-forme accuse son âge et fait preuve d'une lourdeur qui la rend quelque peu inadaptée aux travaux les plus simples.

■ Marie Varandat

Fiche technique

Produit :
Crystal Reports, outil de reporting de masse pour environnements Unix IBM AIX, Sun Solaris, Windows 2000 et 2003 compatible java et .NET

Editeur : Business Object

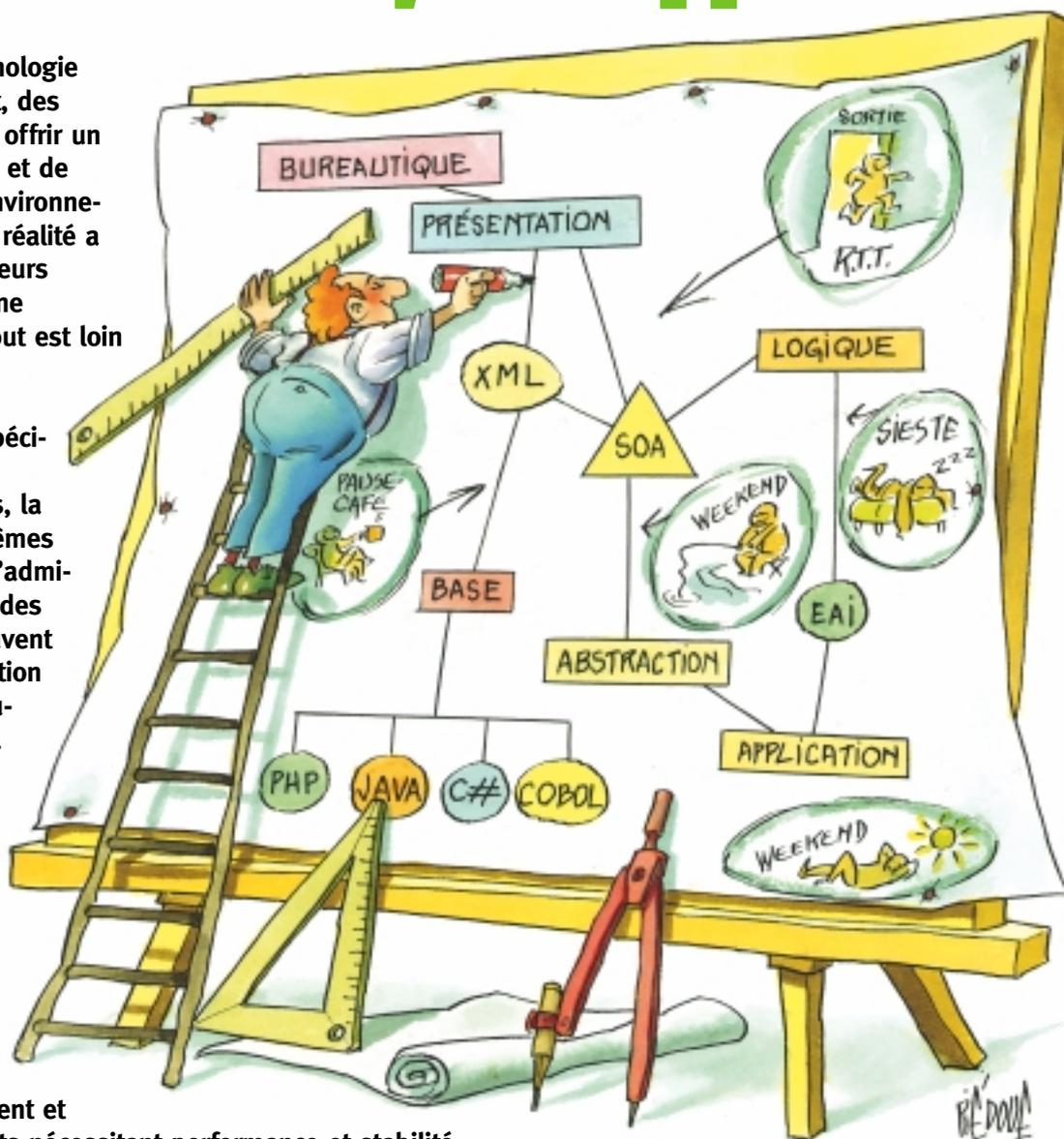
Prix : à partir de 550 euros HT, 2000 pour la version Advanced Developer

Principaux concurrents :
Actuate 8 d'Actuate et Reporting Services de Microsoft.

Web Services : Pouvez-vous y échapper ?

S'il est vrai que la technologie possède, théoriquement, des atouts indéniables pour offrir un socle solide d'échanges et de communications inter environnement et applications, la réalité a mis tout de même plusieurs années à se forger, même si encore aujourd'hui, tout est loin d'être optimal.

La lenteur à fixer des spécifications, la guerre des standards entre éditeurs, la multiplication de ces mêmes standards, le manque d'administration et de gestion des processus reconnus peuvent encore freiner une adoption que les éditeurs de solutions ont déjà effectuée. Tous les ténors des logiciels proposent du Web Services, d'une manière ou d'une autre. Si des progrès ont été réalisés dans le monde Java, dans la sécurité et l'interopérabilité, il reste du travail à accomplir pour pouvoir déployer massivement et dans des environnements nécessitant performance et stabilité.



À se demander, si au bout du compte, le Web Services ne sera pas tout simplement LE composant logiciel par excellence, multi-plate-forme, reconnu pour tout et n'importe quoi, communicant, léger. C'est ce qu'il est déjà dans des proportions encore modestes, mais dans le monde de l'intégration d'applications et de données hétérogènes, le Web Services s'implante. Cependant, outre les soucis de performances et des standards, il faudra aussi disposer de développeurs compétents, maîtrisant les protocoles et standards, même si certains éditeurs veulent nous faire croire que ce n'est pas nécessaire. Pour du service léger, oui sans aucun doute, mais dès que l'on cherche l'optimisation ou à mettre en place une structure complexe, la compétence technique est plus qu'indispensable.

François Tonic

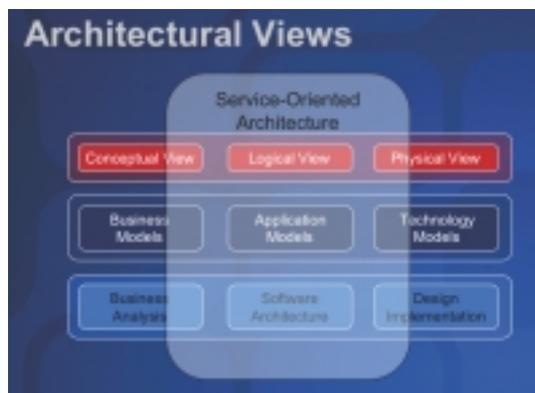
Web Services : entre existant et architecture

Le service web modifie profondément l'architecture applicative et la façon de créer une application modulaire et distribuée. Si depuis plusieurs années la logique d'une application multi couche était préférée, ce n'est pas pour autant que l'application monolithique était définitivement enterrée.

Le service Web, associé ou non avec du SOA, oblige à utiliser une couche d'abstraction qui, sur le long terme, s'avèrera compétitive et indispensable pour préparer les évolutions. À condition, cela va de soi, que l'architecture dé耦lée (que l'on nommera ici couplage lâche ou mou) soit bien réalisée, avec une définition stricte des fonctions/ métiers à mettre sous forme de services. Profitez des Web Services pour repenser votre architecture tout entière, quitte à mettre en place du SOA si cela se révèle utile. Car, cela oblige à mettre à plat votre organisation et à penser non plus en terme de technique, mais de logique métier. C'est peut-être cette nouvelle manière de penser l'informatique qui est le véritable enjeu et intérêt du service Web (et du SOA) : le métier et non plus la technique. Même si, au demeurant, la technique (et donc les développeurs) constituera toujours un élément vital, car même si les outils actuels tendent à simplifier et à uniformiser les couches basses, le codage ne va pas disparaître pour autant.

L'enjeu de l'orienté service

SOA (architecture orientée service) n'est pas une notion nouvelle dans le monde informatique, même si cela portait d'autres noms précédemment. Mais contrairement au passé, la notion de SOA prône le métier en lieu et place du technique. Cela signifie que la logique métier - ce qui fait fonctionner l'entreprise et son business- est l'approche à privilégier dans une démarche orientée service, le technique étant là pour réaliser concrètement l'architecture. Avec le SOA, on adapte l'environnement applicatif aux besoins réels de l'entreprise et non plus l'inverse, adapter l'entreprise et sa logique selon la technique et l'application. En SOA, les fonctions, les logiques métiers sont découpées en services (dits "métiers"), réutilisables en interne ou en externe. C'est là que

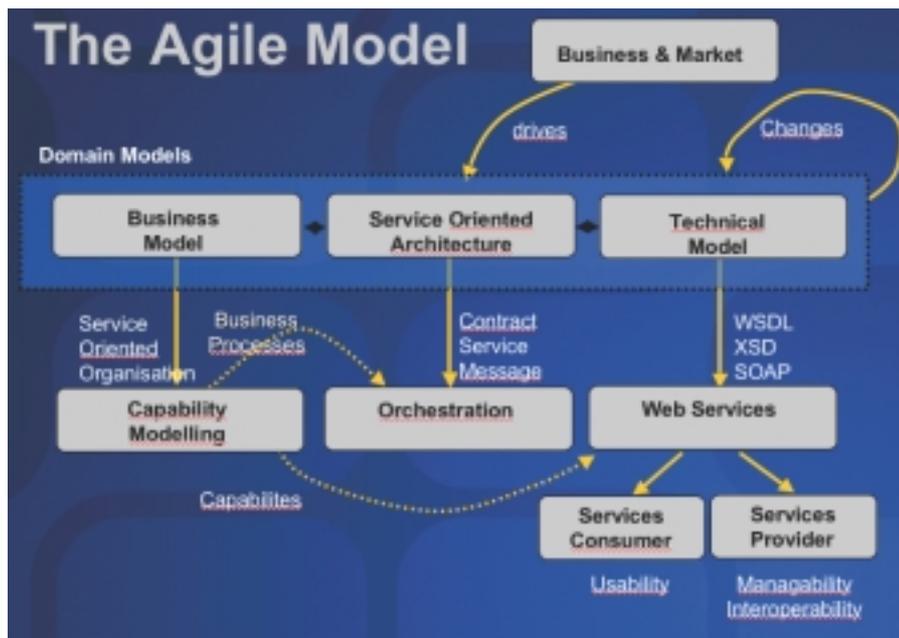


la notion de consommation de service est importante. On fournit un service qui sera "consommé" (utilisé) par un tiers, que ce soit en interne ou par un intervenant externe à l'entreprise, avec un appel dans son application. À cela s'ajoute bien entendu toute une partie d'orchestration des services, pour qu'ils puissent échanger, communiquer ensemble, réaliser une transaction, un processus. Le SOA est une couche d'abstraction de haut niveau qui se "greffe" sur l'architecture technique (le socle applicatif). SOA fournit une interface, la plus homogène possible entre le frontal et le back-office. Cette interface - ici le middleware s'appuie sur les standards "reconnus". Le système de messaging sera unique et homogène de bout en bout. C'est pour cela que le Web Service est utilisé en SOA, car ce sont des

standards, et relativement simple à implémenter.

Le SOA permet une abstraction, mais concrètement à quoi se rapporte-t-il ? Souvent, les applications, en front et back, sont fortement liées (couplage fort). Pour les performances, c'est une bonne chose, mais pour l'évolution du modèle, cela se complique généralement rapidement, car les deux couches étant trop liées, les modifications sont importantes, d'où l'utilisation d'environnement, d'architecture, d'applications multi couches, avec une bonne dose d'abstraction, afin d'acquiescer une souplesse dans les changements. Le SOA et aussi le Web Service introduisent un fort découplage, que l'on nomme souvent couplage "lâche" ou "mou". Cela signifie que le "front" et le "back", pour faire simple, ne sont pas liés. La couche intermédiaire assure une communication homogène, standard et ouverte sans être liée aux autres couches. En environnement hétérogène, distribué, ce type d'architecture est d'une souplesse extraordinaire, car on

rajoute et supprime des messages, des traitements sans grand problème. C'est dans cette interface que l'on met la logique métier. Il devient donc plus aisé d'adapter rapidement une logique métier (via un service métier) : rajouter des metadata, modifier une extraction de données, exposer de nouvelles fonctions... Cependant, pour que le SOA soit efficace, tout comme le Web Service, il faut une analyse rigoureuse de la logique métier de l'entreprise, déterminer le contenu des messages, les flux de données à créer, à gérer, les services qui doivent être exposés ou non, quels services doivent être capables de communiquer ensemble, les standards à utiliser, etc. Aujourd'hui, on fait beaucoup de SOA sur du Web Service. C'est d'ailleurs un des enjeux importants pour les éditeurs. Ce n'est pas un



s'est complexifiée, afin de prendre encore en compte l'agilité et la souplesse nécessaires. Bref, une fois de plus, on simplifie un élément d'interface, tout en rendant la plomberie plus lourde, avec de nouvelles contraintes de définitions et d'implémentations.

Intégration : SOA, Web Services, l'EAI nouvelle génération

Cette architecture "nouvelle génération" influence le modèle d'intégration et de gestion de l'existant. Là aussi, le Web Services (mais aussi le SOA) va supplanter l'EAI que l'on connaît actuellement, pour arriver à quelque chose de plus souple, de plus modulaire et surtout de plus standard (et donc homogène) au moins dans les fondations.

Web Service & le legacy de l'entreprise

Le fameux legacy d'une entreprise (l'existant) est souvent un élément délicat à migrer ou à faire évoluer. Là, il faut définir les éléments à migrer ou à faire évoluer, ceux à garder dans le legacy. Aujourd'hui, avec la logique d'intégration, le Web Services offre une solution extrêmement souple. Ainsi, on crée un frontal (des applications Web ou non), connecté au legacy, via du Web Services. Avantage : peu d'interventions sur le legacy, utilisation de technologies modernes en couche de présentations et fonctionnelles, tout en offrant un couplage mou. Après, il s'agit d'une stratégie

hasard si BEA (avec WebLogic 8 et surtout 9 et le framework Beehive), IBM (avec WebSphere v6 et la suite Atlantic), Microsoft (avec l'ensemble .NET) et tous les autres (Borland, Compuware, WebMethod...), proposent des environnements SOA / Web Services pour la définition du modèle, le développement, le déploiement et le cycle de vie. C'est finalement ce qui se cache derrière le terme "On Demand" d'IBM, ou encore l'agilité selon Microsoft qui prône une architecture souple pour plus de réactivité.

Voilà pour la théorie et l'utilisation idéale,

mais on constate que dans la réalité, les choses diffèrent quelque peu. Trop souvent, l'orienté service est conçu par l'entreprise d'un point de vue technique et non, métier. Les outils de conception "services" entretiennent le doute, car bien souvent, ce sont des environnements proches du technique. Or, pour que l'orienté service soit réellement compris et utilisé optimalement, il faut absolument sortir de la logique technique pour aller à une logique métier. Si le Web Services n'est aujourd'hui pas plus compliqué à concevoir qu'une application Web, l'architecture, quant à elle,

Automatiser, réutiliser

Questions à Henri Cajal (Directeur avant-vente, Unify)

Programmez ! : On parle souvent de l'impact de l'architecture Web Services sur son environnement applicatif, est-ce la réalité ? Peut-on dire que le Web Services sera LE composant par excellence ?



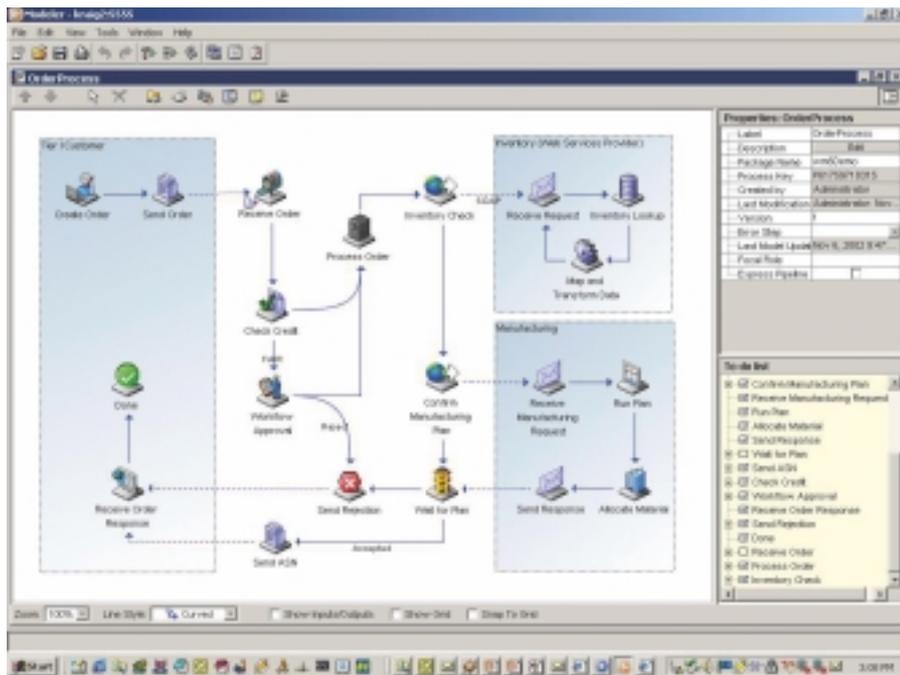
Henri Cajal : Je pense que c'est l'architecture au sens du développement. On a deux possibilités : importer un Web Services existant, c'est le plus simple et évite de réinventer la roue. La seconde possibilité, est d'avoir un code, et de le transformer en Web Services. C'est le niveau fonctionnel, plus orienté métier. Des outils tels que NXJ génèrent les enveloppes SOAP et la descrip-

tion WSDL. Le troisième point est de pouvoir générer la partie cliente à partir du WSDL. Toujours à partir du WSDL, on peut aussi générer un squelette. Ensuite, quand on a le Web Services, il suffit de l'implémenter via un appel dans son code. La tendance, que ce soit chez Unify ou chez d'autres éditeurs, est d'automatiser au maximum. Le Web Services permet des choses plus complexes : EAI, processus... Surtout, il est réutilisable. Par exemple, avec le Web Services, on ne s'occupe plus de connaître la source de données. Cependant, il est important d'assurer l'interopérabilité. Selon certains, dans 10 ans, toutes les applications seraient basées sur une architecture Web Services. Au niveau

du développement, le Web Services ne modifie pas grand-chose, la problématique en entreprise n'est pas forcément sur le développement, mais sur la logistique. De plus, les éditeurs ayant des outils Web Services, ont souvent fait fi de la partie cliente.

PI ! : Une des critiques est la multiplication des standards, mythe ou réalité ? Cette variété nuit-elle aux Web Services, car on serait tenté d'attendre qu'un standard s'impose aux autres ?

HC : Je ne suis pas persuadé de cela. La standardisation a un avantage, mais aussi un inconvénient. Tant que cela concerne l'interopérabilité, c'est très bien, au-delà elle peut nuire aux performances. La performance est aussi bien liée au matériel et à la puissance des logiciels utilisés.



globale à mettre en place. Mais même si vous optez pour une évolution du legacy, il y a des chances que vous gardiez des éléments dessus, d'où le réflexe d'un couplage le plus lâche possible, pour avoir toute liberté d'adaptation des traitements, sans effectuer d'importantes modifications de codes.

Les notions d'intégration et de virtualisation

Dans une architecture distribuée, à couplage lâche, avec de l'intégration, l'ouverture à son environnement, l'implémentation d'une forte logique métier, le service Web et ses standards proposent une bonne solution. Comme dit plus haut et dans le reste du dossier, on dispose d'une architecture message homogène, s'appuyant sur des standards reconnus, utilisant un format et une enveloppe unique (XML / SOAP). Dans un environnement de plus en plus ouvert, distribué et hétérogène, avec la manipulation de données diverses et variées et la nécessité de rapidement s'adapter, le service Web offre un socle fonctionnel intéressant par rapport aux solutions passées. À la place d'un environnement EAI souvent propriétaire et lourd (notamment à cause des connecteurs spécifiques), le Web Services propose autre chose, une interface uniforme partout. On peut ainsi procéder facilement à une virtualisation des données en allant chercher la donnée là où elle se trouve, tout en lui appliquant les traitements nécessaires dans le service Web.

Surtout, pour pouvoir communiquer aisément avec son environnement, le Web service permet une exposition contrôlée et rapide (le WSDL le permet). Le Web Services est un point d'entrée simple et facilement compréhensible, cachant tout ce qu'il y a dans le back office. Un prestataire qui a besoin d'échanger des données n'a nullement besoin de connaître votre architecture applicative, mais uniquement d'un point d'échange que le Web Service expose (et que l'on consomme).

Le "oui, mais"

Cependant, il ne faut pas se leurrer. Le Web Services demeure une architecture difficile à maîtriser dans des contextes complexes et lourds. Le problème des standards peu ou pas matures, ou encore en évolution n'est pas à négliger, ni l'absence de standards éprouvés pour les scénarios complexes. L'orchestration, la transaction demeurent des points noirs, même si des progrès sont faits, notamment dans les processus avec BP4WS et BPML. C'est pour cela, qu'il faut être très rigoureux dans le choix des outils de développement et de déploiement, le support de telles ou telles versions de standards ou de spécifications "propriétaires" sont des points à évaluer. Ainsi, pour assurer une bonne interopérabilité (sur les couches basses du service Web), vérifiez que votre outil utilise WS-I Basic Profile. Car, si les bases fondamentales sont identiques partout, pour les autres couches, il peut

Dépasser les standards

Questions à Philippe Bessis (Marketing & Business Development Manager, WebMethod)

Programmez ! : quels seraient les points faibles des standards du Web Services ?



Philippe Bessis : Ils ne sont pas finis. On se retrouve comme il y a 4-5 ans avec Java. Il était réputé instable, mais prometteur. Maintenant, le souci n'est pas forcément technique, mais il s'agit plutôt de la définition même du service. Si le Web Services c'est faire du RPC, ce n'est pas nouveau et on sait faire. Il s'agit plutôt de créer un " plat de spaghettis ", orienté services et non plus données. Le poids de l'existant est important. À partir de quelle masse critique de services, on peut utiliser une architecture service ? Tant que la notion de service restera du côté technique, cela ne changera pas. Une entreprise, quand elle a un besoin, doit avancer et ne pas attendre.

Programmez ! : Cependant, concernant les standards il y a une sorte de paradoxe. Les éditeurs les implémentent, mais n'hésitent pas à y rajouter des extensions "propriétaires" pour les compléter. Cela n'aide pas à la compatibilité.

Philippe Bessis : C'est évident, mais dans le cas contraire, rien n'évoluerait. C'est un risque, un mal nécessaire pour l'évolution. Les standards constituent un niveau commun de base que tout le monde cherche à enrichir.

y avoir d'énormes différences qui peuvent réduire considérablement l'interopérabilité (emploi d'extensions aux standards ou standards "propriétaires" ou exotiques). Les solutions intégrées comme BEA, Microsoft, IBM, WebMethod, Compuware, SeeBeyond, etc., permettent aussi bien de développer que de déployer le Web Services. Côté performance, cela est assez variable. Cela dépendra de la complexité du Web Services, des applications et bases de données interrogées, de la capacité serveur, du nombre de traitement et de couches utilisées. Dans un contexte lourd, les performances seront sans doute médiocres. N'hésitez jamais à ajuster les services Web lors des tests.

■ François Tonic

Web Services : vers la maturité des standards ?

Microsoft, avec IBM et quelques autres, est à l'origine du concept des services web. Après trois ans d'efforts et d'évolution, nous avons voulu faire le point sur la situation actuelle, avec Marc Gardette (responsable de la division architecture de Microsoft France) et Jean Christophe Cimetière (chef de produit pour la plate-forme d'entreprise, Microsoft France).

Programmez ! : Est-il possible de préciser la situation actuelle des Web Services et comment se déroule la fabrication des standards ?



Marc Gardette

Marc Gardette : Depuis un an, on continue à travailler sur une ligne tracée il y a maintenant trois ans, qu'on avait défini avec IBM une roadmap de ce qui allait devenir les Web Services. C'est un changement fondamental avec à la clé, l'interopérabilité. Cette roadmap comportait des zones où des réponses étaient nécessaires. Le Web Service est une approche modulaire. IBM, Microsoft, BEA ont formé un "noyau dur". Le but était d'arriver à des propositions concrètes. On proposait ainsi des spécifications, puis il y a eu une phase de réaction, de feedback. Pour aller plus loin, nous avons mis en place des "workshop" de feedback, rassemblant des experts du domaine. Quand une spécification est assez mature, on peut éventuellement la proposer à un organisme de standardisation comme OASIS. Ensuite, on peut travailler à l'implémentation de ces spécifications. On veut s'assurer, par ce long travail, que les différents acteurs sont interopérables. C'est pour cela qu'il existe aussi un "workshop" d'interopérabilité où les éditeurs effectuent des tests. Si on prend WS-I, c'est un consortium qui est là pour accélérer l'adoption des Web Services. Il s'assure de la bonne documentation des spécifications. Il propo-

se aussi des "guides" de bonnes implémentations, ainsi que des best practice pour l'interopérabilité. Le WS-I définit pour cela des profils. Par exemple, le basic profil 1.0 concerne l'interopérabilité sur les briques fondamentales des services Web. C'est le WS-I qui gère les profils. Ceux-ci se basent sur des scénarios.



Jean-Christophe Cimetière

C'est un point de repère pour les éditeurs et une garantie pour l'utilisateur.

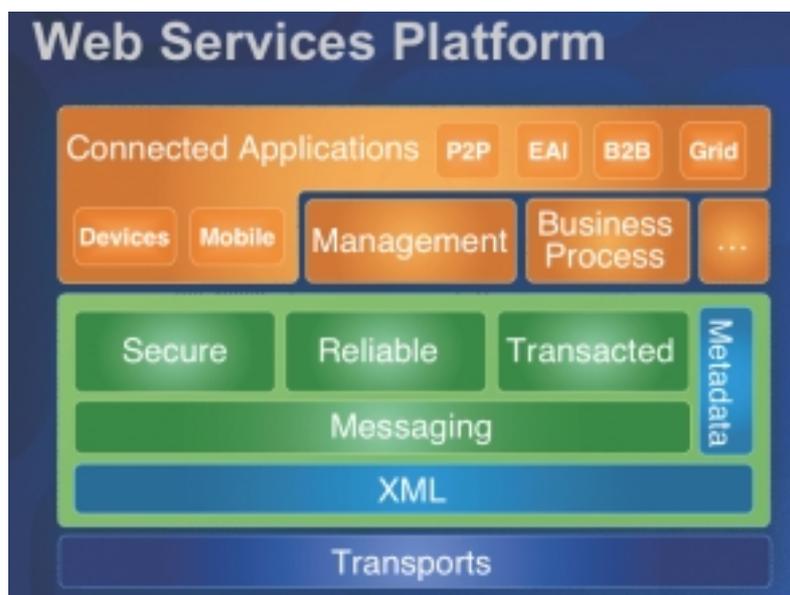
P ! : Aujourd'hui, quels types d'applications peuvent utiliser des services web ?

MG : Le premier exemple, sans doute le plus fréquent est l'interopérabilité entre les couches

Java et .NET. Par exemple, on peut avoir un frontal ASP.NET et avoir un back end en J2EE. Le 2e cas concerne le besoin de posséder une interface homogène au niveau du back end. On remplace, par exemple, les connecteurs "propriétaires" par des connecteurs "services" relativement bien standardisés. Le Web Service répond à ces besoins d'abstraction. Le 3e cas permet d'avoir des services plus dynamiques. On peut désormais, grâce aux standards, faire des choses intelligentes et dynamiques avec les messages. On peut ajouter des services métiers ou d'infrastructures. Par exemple, dans une banque, on peut avoir différentes qualités de services selon le type de client. Grâce au Web Service, on possède une interface homogène et on peut router dynamiquement les messages vers un nœud A ou B. Ou encore, selon le contexte "géographique" du message, appliquer tel ou tel traitement. L'interface est indépendante de l'implémentation et il est facile de réorchestrer des services. Le même service est réutilisable, quelle que soit la couche de présentation. L'interface du service est ce qu'il y a de plus important.

P ! : le service web s'adresse principalement aux entreprises. Mais, cela concerne-t-il toutes les entreprises ou seulement les grandes ?

JCC : Tout le monde est concerné, même s'il faut avoir un minimum de capacité de développement. Il faut bien comprendre que cela concerne l'aspect métier. Si l'implémentation en elle-même ne concerne pas l'entreprise, il y a les progiciels intégrant les Web



Services. Cependant, il y a encore un gros travail à faire pour dire à quoi cela sert. Le DSI, je pense, sait ce que c'est. Le plus difficile demeure sans doute le passage à l'acte, la prise de conscience.

P ! : En même temps que le service web, on évoque l'architecture de services de type SOA. À quoi et/ou à qui s'applique le SOA ? Faut-il lier SOA et Web Services ?

MG : sur le côté SOA, il y a des questions d'organisation. Peut-être est-ce une occasion pour le métier de reprendre la main sur la technique. C'est une démarche assez complexe à mettre en œuvre. La démarche SOA doit aussi se faire sur un ROI clairement identifié. Un des scénarios possibles est l'agrégation de données. Chez Microsoft, on a réalisé une démarche " orientée clients ", pour avoir une vue à 360° d'un client. On a mis un frontal Web Service pour pouvoir extraire les données selon les attentes et besoins des responsables, se trouvant dans différentes bases. Beaucoup d'architectures SOA démarrent sur

une problématique de CRM. Dans le SOA, on ne fait que du fonctionnel. Cette architecture est aussi un pari pour l'avenir : on répond à un besoin réel, tout en préparant le futur.

P ! : Qui doit s'occuper du SOA dans une entreprise ?

JCC : Idéalement, le DSI et les utilisateurs fonctionnels et définir ce qui peut-être exposé à l'extérieur. Le DSI a une vision globale, les gens du métier ont souvent une vision à plus long terme. Cela implique aussi la coordination de nombreuses personnes de l'entreprise. Mais, on peut aussi démarrer par un groupe restreint.

P ! : Au moment du lancement des Web Services, les éditeurs annonçaient la révolution logicielle partout. N'est-on pas allé trop vite dans l'annonce, surtout à cause de l'absence de certains standards comme dans la sécurité ? Les entreprises ne sont-elles pas aujourd'hui prudentes à cause de cela ?

MG : Je ne sais pas si on peut parler de pro-

blème. Le modèle évolue, il existe encore des " vides ". Le socle des Web Service évolue lui aussi, les spécifications ne sont pas sèches. On peut se poser la question suivante : qu'est-ce que l'on peut faire dès aujourd'hui et qu'est-ce que l'on fera demain. Sur la sécurité, cela peut coûter cher, il n'y avait pas jusqu'à récemment d'implémentation des standards. Aujourd'hui, se pose la question des performances. On ne fera pas du temps réel avec du service web !

JCC : Je pense que l'on n'est pas allé trop vite. Il est certain que le service web n'est pas utilisable pour tout et partout. Le Web Service c'est pour l'interaction, un besoin d'adaptation rapide. Certes tout n'est pas parfait mais le modèle s'enrichit peu à peu. On sait échanger avec l'extérieur sans souci.

MG : on arrive à maturité sur les fondamentaux. On possède des protocoles simples tels que SOAP, où le consensus est vite arrivé.

■ *Propos recueillis par François Tonic*

Découvrez et approfondissez les technologies PHP et MySQL avec

Direction | PHP

Programmer avec Quanta Gold p. 76
Le Journal des professionnels de PHP/MySQL N°4 - Janvier 2004 - 5,9€

Ecrire une extension pour PHP

php

Actualité
PHP 5.0 et MySQL 5.0 p.4
Certification PHP p.80

Technologie
Sessions p.45 personnalisées
PHP/XSLT p.70
Cache_Litep.1

Actualités
Trucs et astuces
Alertes de sécurité
Expertise technologique
Dossier entreprise
Interviews (Rasmus Lerdorf, ...)
Dossier du mois (PHP5, XML, ...)
Revue d'applications

12 numéros au format PDF + les codes sources offerts = 59,90 €

Pour commander, rendez vous sur le site : <http://www.directionphp.biz/>
Abonnement pour particuliers et entreprises dans la section " **S'abonner** "
Retrouvez tous les extraits gratuits dans la section " **Extraits gratuits** "

édité par **NeXen.net**

Développez des Services Web avec des outils Open Source

Le monde du Libre et de l'open source a connu ces derniers temps une ascension fulgurante. Les raisons de ce regain sont multiples, un dynamisme retrouvé, une communauté grandissante et surtout la naissance d'un modèle économique nouveau, en quête de maturité, celui de l'Open Source professionnel. Et pourtant, il reste encore des domaines dans lesquels le Libre peine à s'imposer face aux géants de l'industrie. Les Services Web en font partie, et plus généralement celui des architectures orientées services. La situation a-t-elle évolué ? Quels sont les outils libres du moment ? Peut-on aujourd'hui envisager le déploiement de services web uniquement basés sur des logiciels Libres ? Etat des lieux.

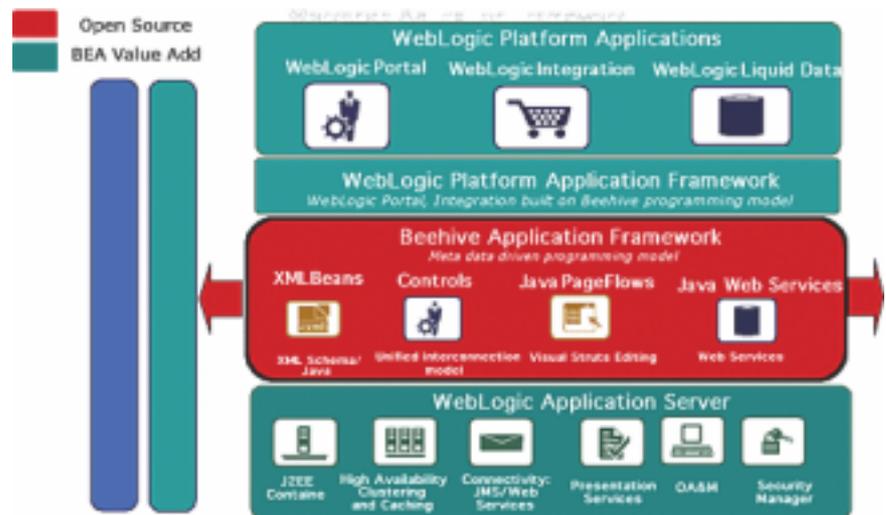
Services Web et Architectures orientées services

Il est assez commun de faire l'amalgame entre Services Web et Architectures orientées services. Il existe en effet une multitude de modes de communication et de protocoles différents permettant de faire communiquer deux composants applicatifs. Les services web au travers du protocole SOAP n'en sont qu'une infime partie. Dans cet article, nous nous intéresserons plus globalement à l'architecture SOA illustrée par les différentes briques du schéma 1.

Les implémentations du protocole SOAP

Nombreux furent les outils Open Source à implémenter le protocole d'échange SOAP. Que ce soit en Python, C++ ou en Java, ils ont tous à leur manière, marqué de leur empreinte la très courte histoire des services web. Aujourd'hui, le candidat le plus sérieux se nomme Apache Axis. Ce dernier est en effet le projet le plus en vogue dans le monde Open Source. Essentiellement destiné aux développeurs Java, il dispose d'un support du langage C++ compatible avec les dernières spécifications du W3C et adresse le standard d'interopérabilité WS-I Basic Profile, gage d'une certaine qualité. Mais la force principale d'Apache Axis réside dans la simplicité de son modèle de déploiement basé sur les formats JWS et WSDO (Web Service Deployment Descriptor). Côté WSDL, les outils WSD2Java et Java2WSDL fournis en standard permettent de générer les squelettes des Services Web (client et serveur) en partant soit de l'interface Java, soit du contrat WSDL.

Parmi les outils connexes à Apache Axis, figurent WSDL4j et WSIF, d'abord développé et

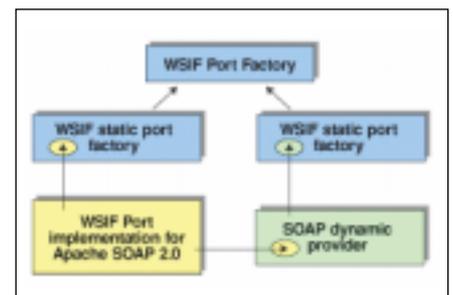


L'architecture selon BEA

maintenu par IBM, puis gracieusement offert à la communauté Open Source sous une licence spécifique (Common Public License 1.0). Aujourd'hui, nombreux sont les grands éditeurs tels que HP, SAP, IBM ou Oracle à intégrer Apache Axis ou son implémentation Soap d'origine,

Les annuaires de Services Web Open Source

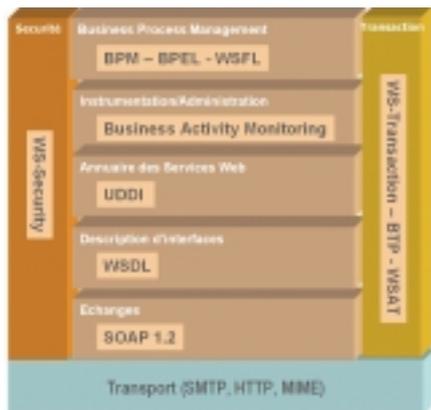
UDDI (Universal Description, Discovery and Integration) est le standard le plus largement répandu concernant les annuaires de Services Web. La plupart des produits Open Source existants s'appuient là encore sur la plateforme Java. Historiquement, la première implémentation fût JUddi (prononcez " Judy ") du projet Apache qui proposait déjà à l'époque le déploiement sur une architecture haute disponibilité, le support de la plupart des bases de données (MySQL, DB2, Sybase, ...) ainsi qu'une intégration exhaustive des serveurs d'applica-



tion du marché (Tomcat, Jonas, Websphere, Weblogic, ...). Au fil du temps, UDDI4j, appuyé par IBM, s'est rapidement imposé au sein de la fondation Apache aux dépens de JUddi pour occuper aujourd'hui la place de meilleur annuaire UDDI Open Source du marché.

Quand BEA offre Beehive à Apache

Beehive est assurément l'un des outils Open Source le plus complet concernant la fourniture de services web. Initialement axé sur l'API



XMLBeans, Beehive constitue un ensemble de briques logicielles permettant d'implémenter de bout en bout une architecture SOA en Java. Offert gracieusement à la fondation Apache, il s'appuie sur un modèle déclaratif (spécification des MetaDonnées de Java 1.5) et propose pour la partie serveur de nombreux composants tels que la gestion de ressources ou de contextes utilisateur. Côté client, Beehive intègre NetUI, un Framework basé sur Apache Struts accompagné d'un ensemble d'assistants graphiques. Le produit peut être déployé sur tout type de serveur d'application, y compris JBoss. Cette nouvelle politique de BEA, confortée par les récents succès d'IBM (eclipse) s'inscrit dans une certaine logique de compétitivité face à la concurrence accrue de Microsoft avec .NET et WSE (Web Service Enhancement). Pour Thomas Murphy du META Group, " Adopter une licence OpenSource est un moyen d'accélérer la création et la standardisation de nouvelles fonctionnalités dans Java plutôt que de nouvelles spécifications". Dans cette démarche, BEA cherche donc à faire adopter son implémentation en s'octroyant une plus large visibilité dans le " vaste " monde de l'Open Source. Dans la pratique, cette soudaine irruption de logiciels propriétaires dans le Libre ne recueille pas toujours un engouement débordant auprès des développeurs. Ces derniers, plus enclins à fédérer autour de projets " originaux ", ne voient pas toujours d'un bon œil l'arrivée des poids lourds de l'industrie dans leur communauté.

L'administration et le monitoring absents

L'administration et le monitoring sont les parents pauvres des Services Web Open Source. Ce marché, essentiellement de niche, est constitué d'outils commerciaux dont la vocation première

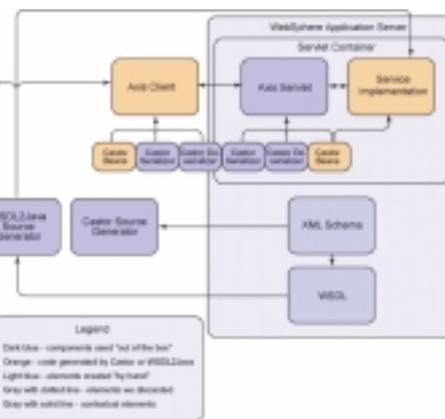
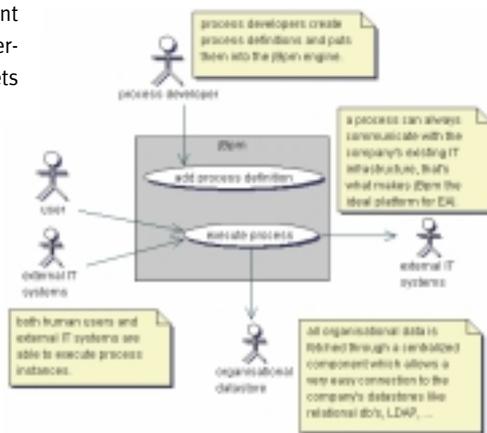
Produit/Editeur	Caractéristiques	Type de licence
JBPM	Définition des processus basée sur des diagrammes UML, J2EE	LGPL
Open Business Engine	Conforme à la norme WfMC (Workflow Management Coalition) , J2EE	Apache licence modifié
Workflow	Support de BPML et BPMLWS	Licence OpenSource spécifique
OSWorkflow / OpenSymphonie	Supporte plusieurs outils de persistance (Hibernate, JDBC, OJB) pour le workflow. J2EE	Apache
Open for business /OFBiz	EIP qui s'appuie sur WfMC's XPDL pour la définition des processus	MIT Licence
Workflow toolkit	Développé en C, ne supporte les protocoles standards	GPL
JFolder/PowerFolder	Très léger. J2EE	LGPL
OpenFlow	Architecture Zope (Python), inspiré du standard WfMC	GPL

(source : jbpm.org)

consiste à gérer l'exploitation des Services Web en production comme par exemple la remontée d'alertes ou l'optimisation des flux (compteurs de performances, règles de gestion, etc...). Les plus évolués fournissent des tableaux de bord permettant d'administrer (parfois en temps réel) un ensemble de composants distribués. D'autres assurent la virtualisation du réseau de Services Web pour répondre aux exigences de montée en charge. On trouve peu ou pas d'outil de ce type en licence Open Source, et force est de constater que le gap à combler est important pour atteindre le niveau de produits commerciaux comme Amberpoint, Actinal ou Blue Titan.

Business Process Modeling (BPM)

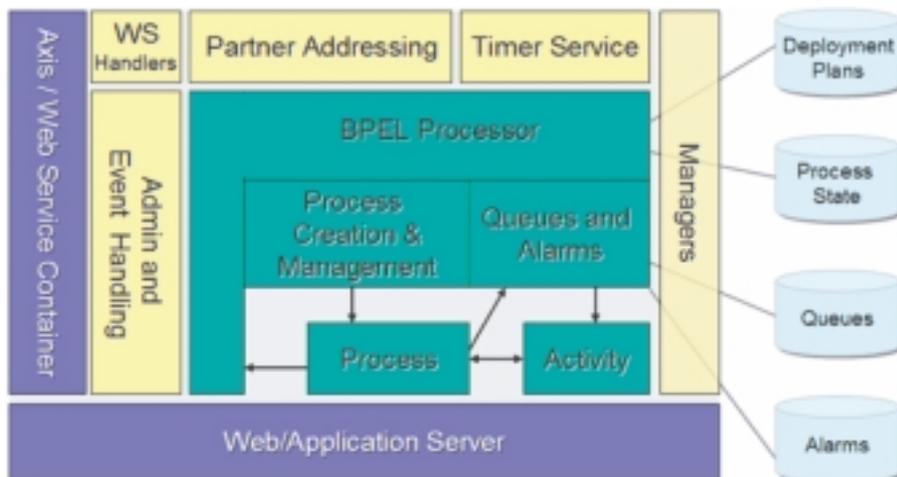
Dans le domaine de la modélisation de processus métier, plusieurs initiatives existent aujourd'hui. C'est le cas notamment de JBPM, un moteur de workflow Java BPM (Business Process Modeling) Open Source. L'outil consiste à modéliser ses processus au travers de dia-



grammes UML. Compatible avec la norme J2EE, JBPM s'appuie en standard sur le serveur d'application Jboss et reprend certains concepts de la norme WfMC (Diagrammes d'activités, diagrammes d'état, ...). Un mode cluster est fourni pour assurer la haute disponibilité et la répartition de charge. Une fonctionnalité habituellement réservée aux outils commerciaux. Il existe également plusieurs projets tiers, de plus ou moins grande envergure permettant de répondre au besoin de modélisation. Le tableau suivant synthétise la plupart d'entre eux.

Le projet WSFX

Le projet WSFX est actuellement l'un des projets les plus prometteurs dans le monde des Services Web Open Source. Créé à l'origine par la fondation Apache, il a pour but de combler le déficit actuel d'implémentations Libres des spécifications WS-*. Pour rappel, le consortium WS-* a pour objectif d'étendre les fonctionnalités de base des Services Web pour leur ajouter des services techniques, comme le support des transactions interopérables, la sécurité, etc.. Si trois modules sont aujourd'hui en cours de développement sur le site ws.apache.org (WS-Addressing, WS-Reliable Messaging et WS-Security), la tâche s'annonce



effet un compilateur WSDL, un studio de développement de Services Web, un serveur web (XSP) et un support limité des spécifications WS-*

Les outils Php

La communauté Php a pris assez tardivement le train des Services Web et XML-RPC fut pendant longtemps le protocole le plus utilisé pour accéder aux composants distribués en Php. La généralisation des Services Web est intervenue avec l'avènement des premières implémentations SOAP comme NuSOAP ou PEAR :SOAP. Mais ces deux dernières bibliothèques sont développées en C et tranchent avec l'homogénéité du Framework Php. Un

rude car les moyens mis en œuvre par les grands éditeurs sont largement supérieurs à ceux de l'équipe actuelle du projet WSFX. Et l'absence d'une implémentation Open Source des API WS-* serait un handicap sérieux pour la communauté Open Source.

Active BPEL

BPEL (Business Process Execution Language) s'inscrit comme le langage d'orchestration par excellence des Services Web. Largement adopté par le marché (un des rares standards à réunir Microsoft et la communauté J2EE), BPEL dispose de plus d'une cinquantaine d'implémentations en tout genre.

Dans le monde Open Source, le Framework le plus célèbre est sans aucun doute Active BPEL. A l'origine Active BPEL a été créé pour accélérer l'adoption du standard BPEL. Nombre de produits gratuits ou commerciaux s'appuient sur le moteur d'exécution d'Active BPEL. En revanche, son interface graphique Active EndPoints et son moteur de workflow, ActiveWebflow, sont payants et complètent l'offre Open Source de l'éditeur.

Les implémentations "exotiques"

Le site officiel des projets OpenSource, SourceForge.net recèle une quantité considérable d'outils et de Framework pour le développement de Service Web. On n'y compte pas moins d'une quarantaine d'outils différents relatifs au protocole SOAP. De python (Python WS) à C++ en passant par Php, Perl ou Java. Mais aussi certaines curiosités, comme cet outil de Grid Computing basé sur les Services Web baptisé Clarens ou encore phpUDDI, cette version Php de l'annuaire UDDI du W3C.



Mono, outil de référence pour les Services Web Open Source en .NET

Il est de notoriété publique que Microsoft n'est pas le partisan le plus volontaire de la cause Open Source. En dehors du projet Mono, il existe peu d'outils Open Source disponibles sur le marché. Et pour cause, l'éditeur, avec sa plate-forme WSE (Web Service Enhancements) propose déjà "gratuitement" le Framework GXA (Global XML Web Services) et de nombreux outils "prêt à l'emploi". A noter toutefois sur gotdotnet.com, l'équivalent Microsoft du site SourceForge quelques projets assez originaux, tels des validateurs de schémas XML, un navigateur WSDL ou des générateurs de Proxies SOAP. Du côté du projet Mono, l'offre est assez fournie, même si moins complète que celle de Microsoft. Mono propose en

problème désormais résolu avec la dernière version de PHP (5) qui marque un tournant dans le support natif de SOAP 1.2, toutefois encore expérimentale.

Conclusion

Développer des Services Web et une architecture SOA avec des outils Libres est aujourd'hui possible. Que ce soit avec Apache Axis, Beehive ou Mono, les implémentations J2EE et .NET sont d'ores et déjà disponibles. Par ailleurs, une nouvelle vague semble animer le monde du développement Open Source. De plus en plus d'éditeurs commerciaux commencent à prendre la mesure de ce marché parallèle en "offrant" leurs produits à la communauté du Libre.

Ainsi, elles rompent avec l'effet d'isolement dans lequel se trouvent souvent ces éditeurs et contribue à améliorer la crédibilité et l'image de leur produit.. IBM et Novell ont été les premiers à ouvrir le bal avec Eclipse ou Mono, BEA a entériné la démarche avec Beehive et il y a fort à parier que d'autres éditeurs emboîteront le pas. Si ces initiatives se confirment, elles contribueront à généraliser l'emploi des Services Web dans le monde du Libre. Les développeurs eux, bénéficieront d'outils et d'API régulièrement remis à jour face à l'évolution effrénée des spécifications des organismes de normalisation (W3C, Oasis,...). Seul petit bémol, les fortes disparités existant entre les outils Open Source et l'absence d'un véritable studio d'orchestration intégré comme peuvent l'être Biztalk ou WebMethods.

■ Sami Jaber

Quand les Web Services vont-ils sortir de l'intranet ?

Depuis le renfort de choix qu'ils reçurent en 2001, lors du lancement avec cotillons et paillettes de l'architecture .NET de Microsoft, les Web Services n'ont pas encore connu en 2004 le grand rush tant attendu, avec une utilisation bien souvent confinée à la sphère intranet. Cet article revient sur certains points qui semblent entraver la progression des Web Services, et s'intéressera aux solutions apportées par les éditeurs.

Si la première impression, corroborée par le discours marketing enjoué des éditeurs de solutions impatientes, peut laisser penser que les Web Services sont extrêmement simples à maîtriser, puis à mettre en place, il faut d'emblée tempérer tout enthousiasme excessif. Nous devons garder à l'esprit qu'un usage massif des Web Services ne peut se faire sans une remise en question de la construction d'applications, et plus généralement de l'organisation du Système d'Information. De fait, pour être adaptés aux problématiques d'externalisation, les Web Services vont de pair avec le concept de SOA (Service Oriented Architecture) [1], qui implique un découplage fort entre clients (présentation, ergonomie) et services (logique métier). A elle seule, cette propriété anodine représente un frein majeur pour beaucoup de SI. En effet, la grande majorité des systèmes d'informations en place repose sur des couplages forts entre composants logiciels, rendant l'utilisation de Web Services extrêmement complexe, avant même d'aborder la problématique de l'externalisation de ces derniers vers l'extérieur de la société.

Un changement d'architecture indispensable

Ce problème connu depuis plusieurs années par les architectes du système d'information trouve sa solution aujourd'hui avec la définition de modèles architecturaux de type SOA, utilisant différentes couches logicielles d'abstraction, l'une d'entre elles étant dédiée à l'hébergement de Services (Web ou non). Avec un tel découpage, les services offrent des points d'entrée vers les processus métier, leur responsabilité étant de satisfaire leur client, en coopérant avec les éléments des couches de plus bas niveau, sur la base cette fois de couplages le plus souvent forts.

Hier, le développement de solutions logicielles dans leur grande majorité commençait par la définition d'un schéma de base de données,

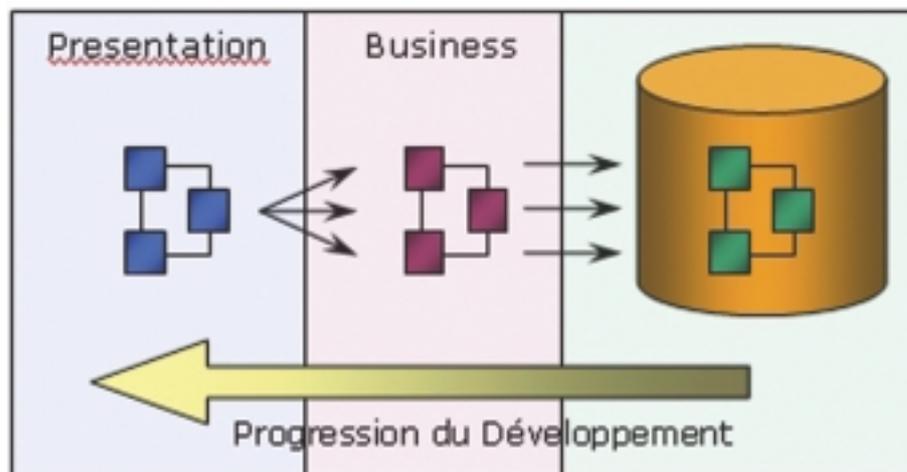


Fig. 1 : Cycle de développement d'une application classique

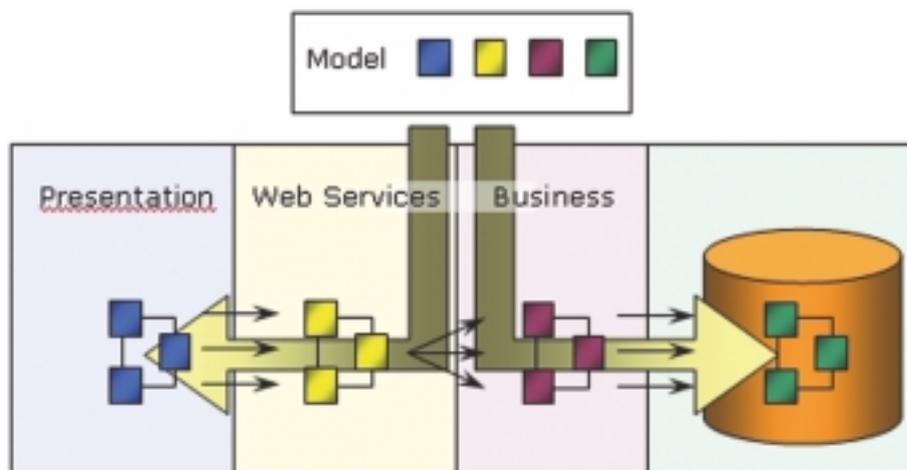


Fig. 2 : Cycle de développement d'une application avec l'utilisation d'un modèle

qui était exploité directement au niveau des interfaces graphiques, la logique métier étant soit intégrée aux interfaces, ou stockée en base de données.

Une architecture orientée service permet, à l'opposé, la définition d'une solution multicouche, adaptée à l'utilisation de Web Services. Les démarches actuelles de développement de type MDA permettent d'ailleurs d'intégrer automatiquement les meilleures pratiques, et donc ce type d'architectures, tout en

tentant d'alléger la démarche projet. Sur le même principe, un certain nombre d'outils basés sur les technologies objets Java ou .NET (dont DTM de Evaluant [4], OptimalJ de Compuware, ...) sont ainsi dotés de modèles permettant de générer automatiquement des couches Web Services, Business et Data Access indépendantes de la base de données, et reconfigurables sur simple changement de la conception UML d'origine. Ce type d'outils offre davantage de malléabilité au SI, en faci-

litant la maintenance des services et des couches sous-jacentes. D'autre part, la propriété de découplage des Web Services affranchit le client de toute connaissance de la logique métier appliquée côté serveur, ce qui tend à favoriser le découplage du SI.

Un second axe limitatif des Web Services est lié aux compromis acceptés lors de leur standardisation, aux fins de ménager les différentes parties impliquées dans cette technologie. La conséquence de cela est une réduction de l'espace pour l'émergence de vendeurs de services, qui en assurant une certaine indépendance entre fournisseurs, permettrait à cette technologie de franchir le palier de l'intranet/extranet, pour atteindre un niveau de service commercial.

De fait, un certain nombre de propriétés, légitimement attendues dans toute architecture applicative, n'ont pas ici fait l'objet de normalisation. Pour n'en retenir que trois, nous pourrions citer le transport d'objets complexes, la sécurité applicative ou la définition de Web Services à états. Aussi chaque éditeur de serveur d'applications à base de Web Services a implémenté sa propre solution à ces problématiques, compliquant significativement leur compatibilité et par voie de conséquence, l'externalisation des Web Services.

Ce manque de standardisation a entraîné une certaine frilosité quant à l'utilisation des Web Services en dehors des applications basiques, qui garnissent la quasi-totalité de la littérature disponible sur le sujet. Par effet de boule de neige, le business model prévu à l'origine (Fournisseur de Services, Détaillant et Client) s'est limité à un schéma Fournisseur de Service, Client.

La place au métier

Nous pouvons légitimement penser que l'absence de " détaillants " contribue grandement au cantonnement des Web Services aux intranets/extranets des entreprises. En effet, la définition de ce rôle métier se justifiait d'une part avec la nécessité de libre concurrence que doivent respecter les fournisseurs de services

entre eux, et d'autre part avec la volonté de séparer les aspects d'accès aux services, qui incombent au " détaillant ", des aspects de fourniture du service, qui restent à la charge du " fournisseur ". Aussi, nous allons montrer dans la suite de cet article, les deux solutions proposées par les éditeurs de solutions à base de Web Services, pour favoriser l'émergence de ce rôle métier de " détaillant ".

Le premier aspect concerne le faible niveau de proactivité induit par le modèle métier que nous venons de présenter [5]. En effet, un

d'applications prenant en charge les Web Services commencent à proposer des signalisations adaptées à ce type de problématique, qui permettent d'envisager une ouverture des Web Services vers un accès plus large et véritablement commercial.

De manière générale, une utilisation de Web Services passant par un acteur intermédiaire suppose que l'architecture est capable de déporter les informations d'accès aux services, d'évaluation de la qualité de service, voire de collaboration entre services, au niveau du détaillant. A ce niveau encore les fournisseurs de solutions Web Services (Obliv pour ne citer qu'eux, avec leur solution Oblivsv [6]) proposent une solution que nous allons présenter plus en détail.



Fig. 3 : Le modèle métier associé aux Web Services

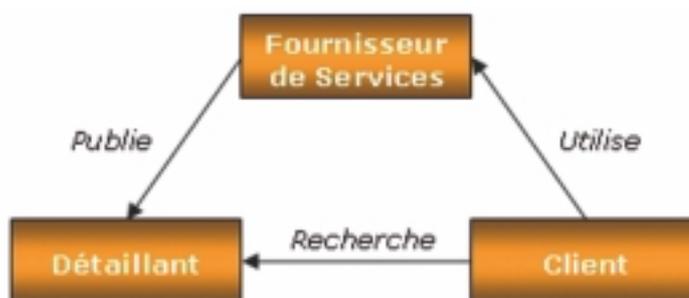


Fig. 4 : Le modèle métier " proactif "

environnement réaliste de production devrait permettre à un détaillant de connaître en temps réel la disponibilité et le temps de réponse des services (hébergés dans le domaine de ses fournisseurs) qu'il propose à ses clients. À partir de cette information, le détaillant devrait être en mesure de faire remonter des alertes à ses clients, pour que leurs applications soient capables à leur tour de s'adapter aux états des services déployés sur le réseau.

Dans un contexte d'utilisation de Web Services en dehors de l'intranet, il paraît naturel que ce soit le détaillant qui assume la transmission des informations de disponibilité des services, démarche faite naturellement au sein de l'entreprise dans le cadre d'utilisations intranet. C'est pour aller dans ce sens que les serveurs

Avoir une bonne démarche

Avec une telle architecture, un Web Service exporté par un fournisseur doit, outre son traditionnel descripteur WSDL, exporter un fichier de configuration, précisant au détaillant les termes de l'utilisation du service. Le détaillant centralise ainsi les Web Services proposés par ses fournisseurs, dont il a la charge de mise à disposition à ses clients, au moyen, là encore, d'un fichier WSDL. Il faut préciser que dans ce modèle, le détaillant aura imperceptiblement modifié la structure du des-

cripteur, pour faire en sorte que le chemin d'invocation du service ne soit plus un accès direct client-fournisseur, mais qu'il passe maintenant par lui. En effet, suite au traitement du fichier de configuration émis par le fournisseur, le détaillant met en place une pile d'intercepteurs auprès d'un policy manager, co-localisé à sa Gateway, par laquelle devra passer l'appel au Web Service. Dans cette pile, chaque intercepteur est spécialisé dans un aspect du fichier de configuration. Les solutions basées sur ce principe proposent ainsi en standard des " intercepteurs ". Ces derniers permettent selon les offres de traiter de sujets tels que :

- L'authentification, en assurant la collecte des informations d'identification du client et la mise à disposition de ces informations aux autres intercepteurs de la pile.

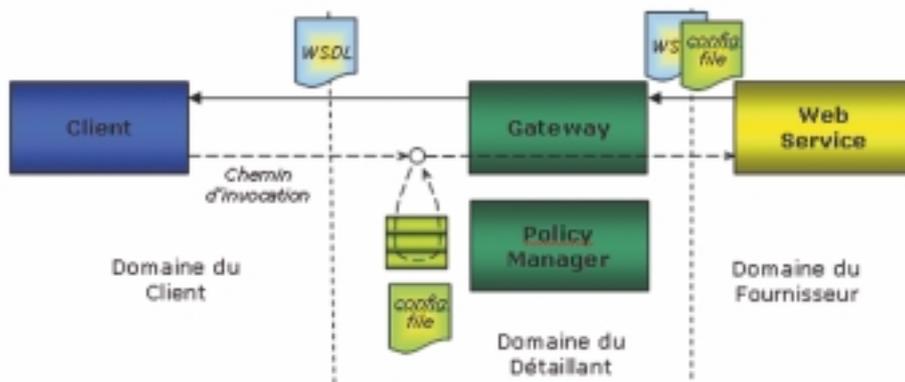


Fig. 5 : Schéma d'utilisation d'une gateway de Web Services

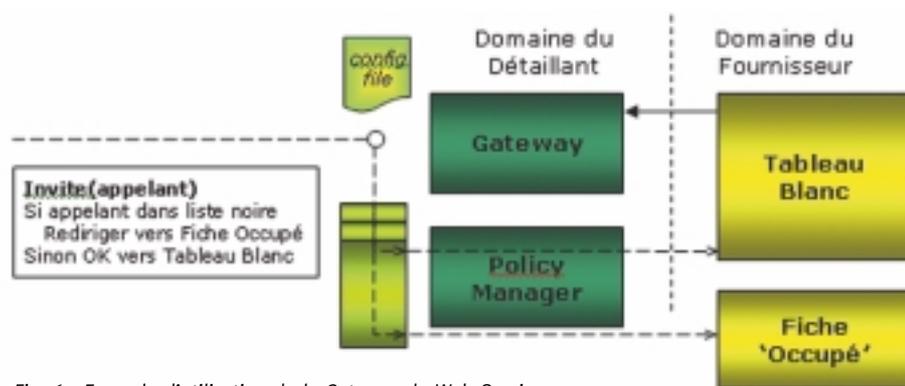


Fig. 6 : Exemple d'utilisation de la Gateway de Web Services

- La gestion de transaction, en autorisant le démarrage ou la poursuite d'une transaction.
- La gestion d'état, en assurant le chargement de données en base, pour donner l'illusion d'un Web Service à état.
- L'Appel à un service tiers. En utilisant les mécanismes de réflexivité offerts par le langage d'implémentation de la Gateway de services, le détaillant ou le fournisseur de services peuvent demander à combiner l'utilisation du Web Service avec l'invocation d'un service distant, pour entrer dans une logique de composition de services.

Ce dernier point est au cœur du problème de l'emploi généralisé de Web Services à tous les niveaux du SI. En effet, si le schéma actuel Invocation-Réponse des Web Services est suffisant et bien adapté à leur utilisation dans un contexte Intranet ou extranet simple, il trouve ses limites dans des usages de coopération entre services, de personnalisation du comportement pour l'utilisateur final, voire de traitements dynamiques de cas d'interactions indésirables entre services. La Gateway de services a ainsi été introduite, pour permettre d'effectuer des traitements en amont, pendant, voire après l'invocation du Web Service, et récipro-

quement en amont ou après la transmission de la réponse du Web Service au client. Cette construction, dérivée de la programmation par contrats [7], est utilisée dans le monde des télécoms pour permettre par exemple à un utilisateur de préciser les conditions d'utilisation de Web Services auxquels il a souscrit. Prenons ainsi un Web Service de "tableau blanc", utilisé pour établir une session de travail collaboratif entre plusieurs utilisateurs. Si un utilisateur souhaite restreindre l'accès à son service, il a la possibilité de définir un contrat, identifiant les utilisateurs avec lesquels il refuse d'entrer en relation. Un tel contrat, déployé dans l'environnement du Détaillant, sera interprété avant l'invocation proprement dite du Web Service, pour déter-

miner si l'appel sera poursuivi, ou si au contraire il devra être redirigé vers un service annexe, de mise en attente de l'utilisateur.

Web services encore trop immatures ?

Nous avons relevé que l'utilisation de Web Services en dehors de simples applications Intranet demandait une remise en question de fond, tant sur l'architecture du Système d'information, que sur l'approche de la conception d'applications. En outre, le modèle de Web Services, régulièrement mis en avant dans les conférences d'évangélisation, n'apparaît pas être suffisamment adapté pour répondre aux problématiques de mise à disposition de services au détail, et encore moins de composition ou d'interactions entre services. Cependant, nous avons pu constater que les solutions émergentes apportent des solutions intéressantes, pour offrir aux Web Services un second souffle, qui leur permettrait de sortir de l'entreprise, et ainsi entrevoir des applications ouvertes, sur lesquelles nous aurions davantage de maîtrise.

Il reste tout de même un débat important autour de l'utilisation de ces solutions. Dans le cadre de la gestion de la sécurité ou de la qualité de service, elles sont parfaitement compatibles avec une urbanisation saine du SI. En effet, les WebServices sont comme des portes autour d'une ville, et gérer leur bon fonctionnement et filtrer les accès est plus que pertinent. En revanche, venir ajouter des fonctionnalités métier, est une possibilité offerte, mais dangereuse pour la stabilité du SI. En effet, qui viendrait construire une maison sur les portes d'une ville ! Une fois de plus, ce n'est pas parce que c'est faisable, voire simple à réaliser, que c'est souhaitable.

■ **Arnaud Fontaine**

chef de Projet chez ASTON Rhône-Alpes.



SSII de 270 personnes, implantée à Paris, Rhône-Alpes et Sud Ouest. www.aston.fr

Biblio :

- [1] <http://www-106.ibm.com/developerworks/webservices/newto/>
- [2] <http://www.omg.org/mda/>
- [3] <http://developpeur.journaldunet.com/tutoriel/cpt/040706-intro-conception-mda.shtml>
- [4] <http://www.evaluant.com/fr/solutions/dtm/default.aspx>
- [5] <http://www-106.ibm.com/developerworks/webservices/library/ws-wssla/>
- [6] <http://www.oblix.com/products/coresv/index.html>
- [7] <http://www.fiadeiro.org/jose/papers/ecoopws.pdf>

Des développeurs estampillés Web Services ?

Que représentent à l'heure actuelle les Web Services pour ceux qui donnent vie de manière quotidienne aux applications, je veux bien sûr parler des développeurs. Les Web Services font-ils partie de l'environnement récurrent de tout développeur, ou bien leur mise en oeuvre est-elle réservée à des spécialistes ? Il est temps de faire le point sur un aspect de cette technologie rarement mis en avant.

Le développeur et la spécialisation

Le terme de spécialisation est ambivalent. Que faut-il deviner derrière ? Le cloisonnement ou bien l'expertise ? Dans un précédent article paru dans *Programmez !*, Hervé Crespel prévoyait un changement de fond du métier de développeur avec une segmentation des métiers du développement. Il identifiait en outre les profils suivants : le "développeur d'orchestration", le "développeur de services" et le "développeur spécialiste".

Ce qu'il faut comprendre, c'est que le processus de spécialisation est général, et pas seulement présent dans le domaine des Web Services, il fait partie de l'avenir même des développeurs. Alors, que faut-il attendre d'un développeur spécialisé dans les WS? Je pense qu'il faut aller plus loin que la "simple" expertise technique et voir derrière le terme de spécialiste un acteur prenant part à l'accompagnement d'une entreprise en terme de choix technique, quelqu'un capable de mettre en oeuvre une formation, et qui a une culture des standards de bas niveau. En outre, le développeur devra se différencier de façon pragmatique, de l'architecte ou du consultant, par la partie mise en oeuvre technique de ces Web Services. Cependant, cela demande forcément un effort de veille technologique pour lequel un développeur n'a pas toujours le temps imparti. En effet, seules les équipes de développement de Microsoft peuvent s'accorder 2 heures de veille par jour

Il persiste un point qui ne va pas dans le sens d'une spécialisation dans les Web Services, c'est la reconnaissance des compétences, ainsi qu'en est-il des certifications ? Pourquoi dans un tel cadre d'interopérabilité entre les différents éditeurs de solutions logicielles, n'existe-t-il pas une certification officielle, mise au



point par un regroupement d'éditeurs, dans le but de certifier des développeurs ou des consultants sur l'ensemble des plates-formes usuelles, et de s'assurer qu'ils en maîtrisent bien les subtilités. Car si les langages sous-jacents sont standardisés (SOAP, WSDL, UDDI pour ne citer qu'eux), c'est loin d'être le cas pour les APIs de consommation des Web Services, ou les mécanismes de déploiement. Néanmoins, des initiatives telles que JAX-RPC dans le monde JAVA traduisent l'effort d'unification, d'autant plus que les éditeurs suivent. C'est le cas, par exemple, de BEA qui a refondu son moteur de Web Services en passant d'une API propriétaire en version 6.1 à une implémentation JAX-RPC en 8.1 (d'où la réécriture des anciennes applications en cas de migration...). Enfin, La spécialisation devient

un critère de différenciation important pour les développeurs qui veulent "sortir du lot", la connaissance des Web Services peut être un de ces critères, d'autant plus qu'un article récent relate que la France manque de ressources maîtrisant cette technologie, c'est un point important à une époque où le développeur ne peut plus se reposer sur ses acquis, mais doit toujours se remettre en question, afin de rester compétitif.

Les stratégies éditeurs

Les stratégies des principaux éditeurs ne facilitent pas la mise en oeuvre des Web Services. Ainsi, certaines approches sont radicalement opposées, prenons l'exemple du déploiement d'un Web service :

- En environnement .NET la mise à disposition d'une méthode d'une classe se fait tout simplement en ajoutant au début de la méthode un tag [webmethod]. Le déploiement est donc intégré dans les classes C#.
- En environnement J2EE, le déploiement d'une classe et de ces méthodes est décorrélé et réalisé par le biais d'un fichier de déploiement WSDO (Web Service Deployment Descriptor)

Ce n'est qu'un exemple, parmi tant d'autres, de la conception différente d'une même idée par des éditeurs concurrents.

L'effort à fournir par le développeur pour se spécialiser est important, dans un domaine où chaque éditeur essaie de tirer son épingle du jeu en ajoutant de petites subtilités techniques aux APIs existantes. Par exemple, BEA avec le WLCall et le WLSERVICE qui sont des versions étendues du Call et du Service de la librairie JAX-RPC. De même pour la gestion des pièces jointes en SOAP au format DIME, uniquement supportée en environnement .NET. On peut encore citer l'exemple de BPELJ, nouvelle API ne fonctionnant qu'entre des éditeurs JAVA

(IBM, Oracle, BEA) alors que pourtant ce sont IBM et Microsoft qui en sont à l'origine, avec la fusion de XLANG et WSFL, du standard de gestion de processus métier BPEL4WS (rebaptisé depuis BPEL)... Ainsi, les choix techniques seront fortement liés à l'interopérabilité entre les plates-formes d'exécution, et le développeur devra être au fait des travaux du WS-I (Web Services Interoperability) avant d'employer telle ou telle fonctionnalité, ou de légitimer tel ou tel choix technique lors du développement d'un Web Services.

En parallèle de ces choix techniques, la maîtrise des environnements de développement est elle aussi le véritable atout du développeur et l'intégration des Web Services ne fait que renforcer ce phénomène. Par exemple, dans Visual Studio .Net, il suffit de pointer sur l'URL du fichier de description WSDL d'un Web Service pour générer une interface de connexion (ou stub). Il ne reste plus qu'une ligne de code à écrire pour se connecter au service distant. Cet exemple, que l'on retrouve dans bien des articles, est bien évidemment simpliste, puisque le véritable enjeu n'est pas de faire un simple appel à un Web Service, mais bien de bâtir des applications complètes à l'aide des standards. C'est là qu'une maîtrise d'un IDE sera un atout indéniabile en terme de productivité, et c'est également sur ces briques " haut niveau " que les différences entre un développeur spécialisé et un généraliste se feront le plus sentir.

Les nouvelles manières de développer

Les Web Services sont également un des vecteurs participant à la modification de fond des méthodes de développement. En effet, la notion de service par opposition à la notion de composant bouleverse le mode de développement actuel, en opposant " orienté processus " et " orienté fonction ", " cycle incrémental " et " cycle long ", " couplage faible " et " couplage fort ". Ainsi, les architectures reposant sur des Web Services que l'on retrouve souvent sous le terme de SOA (Service-Oriented Architecture) sont considérées comme des

ensembles de services autonomes qui sont ensuite reliés par de savantes règles d'orchestration dans le but de modéliser un processus métier.

Cela débouche naturellement sur les nouvelles approches en terme de gestion de projets (méthodes agiles, eXtreme Programming) qui ont, elles aussi, des incidences sur les développements (programmation en binôme, automatisation des tests, etc.). Il est important de préciser que l'approche SOA au travers de l'utilisation de services donne, si cela était nécessaire, encore plus de crédit aux nouvelles méthodes de gestion de projet qui s'imposent maintenant de fait...



Les Web Services pour le développeur

Quand on parle de Web Services, il me paraît important de resituer la notion, et de ne pas la cantonner au simple appel RPC (la figure 1 propose une progression de la complexité selon le traitement désiré). Il faut préciser que si la partie gestion du transactionnel est la plus complexe c'est qu'elle reste embryonnaire pour le moment, en terme de spécification, et que c'est d'ailleurs la partie la plus critique à traiter à la vue de l'hétérogénéité des mécanismes transactionnels. De même, il faut distinguer la problématique relativement simple de consommation d'un Web Service, de la fourniture d'un Web service qui requiert les compétences d'un développeur spécialiste.

Il apparaît que dans un avenir proche les Web Services seront une composante incontournable des projets informatiques allant de la simple agrégation de contenu distant à l'ESB (EAI stan-

dardisé) en passant par le client riche (ou client intelligent). Dans ce contexte, il deviendra primordial de disposer d'un expert maîtrisant les couches basses. En effet, si les API visent à faciliter le travail des développeurs, il est indispensable de savoir comment est constitué un flux SOAP, ou de savoir déchiffrer un fichier WSDL. Si par exemple on souhaite valider un flux SOAP à l'aide d'une grammaire XSD, cette action ne devra pas être positionnée au niveau du traitement métier, mais intégrée dans un Handler JAX-RPC, une classe qui vient s'interposer entre la réception d'un flux et le traitement métier. Elle permet en outre de récupérer les informations du header SOAP et du body SOAP

sous forme de XML, c'est donc le seul endroit (avant désérialisation) où la validation XML/XSD peut être réalisée, mais pour en arriver là, encore faut-il savoir ce que sont les header SOAP et les body SOAP... On en revient au problème du développeur Web Services qui doit trouver son chemin dans le labyrinthe des standards (ou pseudo-standards...) au sein duquel le non-initié peut vite se perdre. D'autant plus que le terme de standard est bien souvent usurpé et ne désigne en fait qu'une spécification. Il est bon de rappeler que les seuls

"standards" disponibles à l'heure actuelle sont la " partie haute de l'iceberg " composée de SOAP, WSDL et UDDI, WS-Security pour la partie sécurité et je serais tenté d'ajouter, au risque de brûler quelques étapes, BPEL pour la partie gestion des processus métier qui s'impose de fait par le biais de l'OASIS.

Conclusion

La spécialisation des développeurs dans chacun des domaines applicatifs semble irréversible, ce constat se vérifie d'autant plus dans le monde des Web services, où la complexité des standards et des produits appelle intrinsèquement cette notion d'expertise. Cependant, cette spécialisation est finalement très positive, car elle permet d'approfondir les notions, d'autant plus que le sujet est vivant, riche et répond souvent à des problématiques métier d'intégration ou de partenariats passionnantes.

■ Cédric DUPUI, développeur, Groupe SQLI

WSE 2.0 : la nouvelle étape des Web Services ?

Le groupe OASIS pour le développement, l'uniformisation et l'adoption des standards d'échange pour le e-business par services web poursuit son effort pour établir un ensemble de spécifications basées sur XML et SOAP qui permet d'offrir aux services web des protocoles inter-opérables pour la sécurité, la garantie d'échange de messages et les transactions. Ces standards ne sont pas tous aujourd'hui finalisés et WSE 2.0 est l'implémentation d'un ensemble de ces spécifications pour la plate-forme .NET.

Les spécifications implémentées sont :

Spécification	XML namespace
WS-Security	http://docs.oasis-open.org/wss/2004/01/oasis-2004-01-wss-wssecurity-secext-1.0.xsd
WS-SecurityPolicy	http://schemas.xmlsoap.org/ws/2002/12/secext
WS-SecureConversation	http://schemas.xmlsoap.org/ws/2004/04/sc
WS-Trust	http://schemas.xmlsoap.org/ws/2004/04/trust
WS-Referral	http://schemas.xmlsoap.org/ws/2001/10/referral
WS-Addressing	http://schemas.xmlsoap.org/ws/2004/03/addressing
WS-Policy	http://schemas.xmlsoap.org/ws/2002/12/policy
DIME	June 17, 2002 version
WS-Attachments	http://schemas.xmlsoap.org/ws/2002/06/dime/soap

Les messages

WSE fonctionne au niveau du message SOAP véhiculé par les services. Il est de fait indépendant de la couche de transport utilisée pour véhiculer les messages. Ainsi, le SDK de Microsoft supporte les transports In Process, HTTP et TCP en permettant aussi de développer son propre transport. De la même façon qu'un service Web implémenté avec ASP.NET se déclare avec les attributs [WebService] et [WebMethod], un service Web implémenté avec WSE 2.0 se déclare avec [SoapService] et [SoapMethod], la différence majeure étant que le service ASP.Net ne sera exposé que via HTTP.

L'adressage

L'adressage des messages est une des avancées majeures de WSE 2.0. Les consommateurs de services web nécessitent aujourd'hui de connaître l'endroit où se situent les services, ce qui résulte souvent en des environnements où les systèmes ont une connexion point à point entre eux. WSE 2.0 apporte une nouvelle méthode d'adressage des messages, qui les véhicule d'un nœud à un autre, sans nécessiter d'indiquer dans l'en-tête du message le chemin à prendre, mais uniquement le destinataire final. Ceci permet de mettre en place des serveurs dédiés au bon routage des

messages, sans que les consommateurs se soucient du serveur offrant le service. Cette souplesse supporte bien entendu les autres standards et n'affectent en rien la sécurité du transport par exemple.

La sécurité

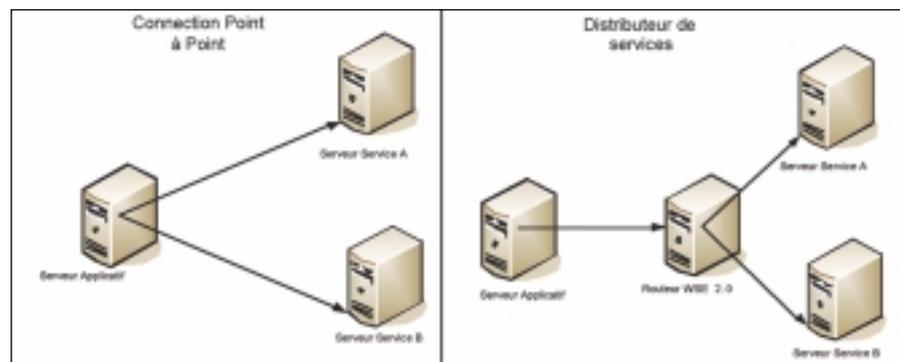
WSE 2.0 offre le support du standard WS-Security qui offre une façon simple et étendue de gérer l'authentification, la signature et l'encryptage des messages SOAP. L'authentification permet de sécuriser l'ensemble de la communication d'un message en supportant l'encryptage et la signature de tout type de jeton de sécurité dans l'en-tête du message SOAP. Ainsi, tout système participant à la communication d'un message et supportant l'authentification fournie

par le développeur dans l'en-tête SOAP pourra exploiter de façon sûre les informations rendues par WSE 2.0. La signature d'un message offre la possibilité de garantir que le message n'a pas été modifié lors de son transport, depuis qu'il a été signé par son expéditeur, et peut-être mise en œuvre par les fournisseurs de jeton de sécurité comme les certificats X509, les couples username/password, les tickets Kerberos ou autres. L'encryptage est l'opération consistant à brouiller le contenu du message pour qu'il ne soit lisible que par le destinataire, garantissant ainsi aux deux parties que le message ne peut être lu par un système tiers. L'expéditeur utilise la clé publique du destinataire du message, qui ne devient alors lisible qu'au propriétaire de la clé privée correspondante.

La validation

La validation des messages par WSE 2.0 est l'implémentation du standard : WS-Policy. Son principe est de valider des règles sur le contenu d'un message échangé entre le fournisseur du service et son consommateur. La validation s'effectue à l'expédition du message par le consommateur et à la réception du message en amont du service. Les règles sont définies au niveau d'un fichier de configuration XML et peuvent consister en un ensemble de règles prédéfinies, telles que la signature obligatoire par un certificat X509 ou en des règles custom validant le contenu de paramètres par exemple. Toute non-conformité résulte en la levée d'une exception SOAP ; si elle est levée à l'envoi du message par le consommateur du service, celui-ci a la possibilité de modifier le contenu du message pour le forcer à suivre la règle.

■ Eric Bouguen - Senior Consultant, Avande



WSE 2.0 : installation & programmation

Le framework .Net fournit nativement de nombreuses classes, permettant de développer des applications qui consomment et produisent des WebServices. Microsoft propose, avec les Web Services Enhancements (WSE), un add-on au framework .Net et à Visual Studio .Net. Il permet au développeur de construire des WebServices sécurisés, basés sur les derniers standards et spécifications en vigueur.

Les WSE 2.0 étendent les fonctionnalités de sécurité, de routage, et d'intégration de pièces jointes dans des messages SOAP de la version 1.0. Ils permettent aux développeurs de construire des applications basées sur les dernières spécifications publiées par Microsoft et différents acteurs de l'industrie (IBM, Verisign, SAP...), telles que WS-Security, WS-Policy, WS-SecurityPolicy, WS-Trust, WS-SecureConversation et WS-Addressing.

L'architecture de WSE 2.0

SOAP propose un modèle standard de description de messages contenus dans un document XML. Un message SOAP est constitué d'une enveloppe, contenant le corps d'un message et d'un en-tête pour le décrire. L'en-tête (<soap:Header>) est utilisé pour transmettre des informations sur la manière dont le message doit être traité : par exemple, des informations de cryptage, d'authentification.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Digest>B839D234A3F87</Digest>
  </soap:Header>
  <soap:Body>
    <StockReport>
      <Symbol>MSFT</Symbol>
      <Price>74.56</Price>
    </StockReport>
  </soap:Body>
</soap:Envelope>
```

WSE est constitué d'un jeu de classes qui implémentent de nouveaux protocoles (WS-Security ...) et d'un jeu de filtres, hébergés par ASP .Net, qui interceptent les messages SOAP entrants et sortants. Ces filtres interprètent ou génèrent les en-têtes permettant de prendre en charge les fonctionnalités requises.

Prenons le cas d'un fichier de configuration Policy défini. A l'exécution, le runtime WSE vérifie que tous les messages entrants et sortants respectent les prérequis définis dans le fichier Policy. Si ce n'est pas le cas, alors une exception SOAP est levée (Soap Fault). Par défaut, un certain nombre de paramètres sont prédéfinis : par exemple, le corps du message doit être signé avec un certificat X.509.

Installation de WSE 2.0

Le package d'installation de WSE 2.0 SP1 (Microsoft WSE 2.0 SP1.msi) est téléchargeable depuis le Microsoft Download Center à l'adresse sui-

vante : <http://msdn.microsoft.com/webservices/building/wse/default.aspx>

Le menu d'installation propose plusieurs types de mises à jour en fonction des profils utilisateurs. Les développeurs choisiront l'option "Visual Studio Developer" qui installe le runtime, les exemples, la documentation, les outils et intègre WSE 2.0 dans l'environnement de développement, via des boîtes de paramétrage. (Fig. 1)

Plusieurs outils sont proposés avec WSE 2.0 :

- **WseWsdllz.exe** : Il permet de générer un proxy client à partir d'un fichier WSDL (Web Services Description Language)
- **WseSettings 2.0 Tool** : Il permet d'activer et de paramétrer les fonctionnalités WSE à partir de l'environnement de développement Visual Studio .Net
- **WseCertificatez.exe** : Il permet de visualiser les certificats présents sur un ordinateur.

La compatibilité entre WSE 1.0 et WSE 2.0

Bien qu'un degré de compatibilité ascendante soit prévu entre les versions 1 et 2 de WSE, WSE2 .0 implémente des spécifications des Web Services mises à jour, mais aussi une version standard OASIS de WS-Security. Les développeurs devront donc retravailler leur code, afin d'intégrer le nouveau modèle objet WSE 2.0. Il est possible d'exécuter des applications utilisant ces deux versions côte à côte sur une même machine, en effet les assemblies et namespaces sont différents. Cela permet de conserver un code existant développé avec les WSE 1.0, tout en intégrant de nouveaux services s'appuyant sur les fonctionnalités de WSE 2.0. Remarque : Un service Web ne peut utiliser les deux SoapExtension (WSE 1.0 et WSE 2.0) en même temps. Voici quelques-unes des différences d'implémentations entre ces deux versions :

- Les assemblies et namespaces sont différentes : Microsoft.Web.Services pour WSE 1.0 et Microsoft.Web.Services2 pour WSE 2.0.
- Des messages SOAP transmis entre des applications implémentant WSE 1.0 et WSE 2.0 aboutiront à un message d'erreur SOAP (SOAP fault)
- La récupération d'un objet SoapContext se fait désormais, via la propriété Current des classes RequestSoapContext et ResponseSoapContext et non plus en utilisant la classe HttpSoapContext.

```
using Microsoft.Web.Services2; // Référence à la nouvelle assembly WSE 2.0
public class MaClasse
{
  [WebMethod]
  public string MaMethode()
  {
    SoapContext reqCtx = RequestSoapContext.Current;
    SoapContext resCtx = ResponseSoapContext.Current;
```

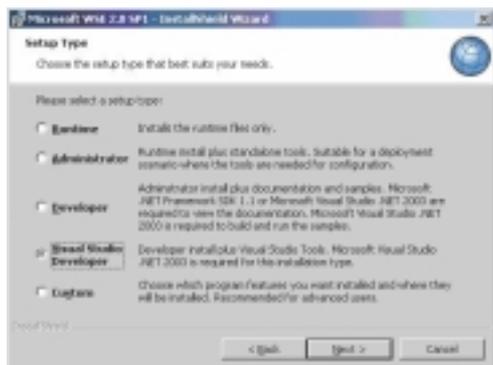


Fig. 1
Installation des
WSE 2.0 SP1

```

}
ooo
}
    
```

- Suppression du namespace Microsoft.Web.Services.Security. Idecryption KeyProvider au profit du namespace Microsoft.Web.Services.ISecurity TokenManager
- Il n'est plus nécessaire de définir une classe implémentant l'interface IpasswordProvider pour authentifier des utilisateurs Windows. Désormais WSE supporte l'authentification Windows d'un UserNameToken.
- La classe IdecryptionKeyProvider devient obsolète.

WS-Security

WS-Security permet à une application de sécuriser ses Web Services et :

- D'identifier un utilisateur qui demande l'exécution d'un service Web (authentification)
- De vérifier le rôle de l'utilisateur et les droits qui lui sont attribués (autorisation)
- De s'assurer qu'un message n'a pas été corrompu durant le transport (signature)
- De s'assurer que seul le destinataire d'un message peut le lire (encryptage)

WS-Security utilise des jetons de sécurité pour identifier un utilisateur. Les informations permettant la mise en œuvre de la sécurité sont stockées directement dans les en-têtes SOAP. L'authentification est gérée soit à partir de jetons UserNameToken pour représenter un utilisateur/mot de passe, soit à partir de jetons binaires (BinarySecurityToken) comme les tickets Kerberos ou les certificats X.509.

Sécurisation d'un service Web avec un UserNameToken

Regardons maintenant comment authentifier un utilisateur à partir d'une identité constituée d'un nom d'utilisateur et d'un mot de passe. Tout d'abord, il est nécessaire de créer un Webservice et une application cliente. Ces deux applications doivent être paramétrées afin de supporter les extensions WSE2.0. Il suffit de sélectionner l'option WSE2.0 Settings de chaque projet et de cocher dans l'onglet " General " l'activation des WSE pour le projet, ainsi que l'ajout des extensions SOAP (option applicable uniquement pour les applications ASP .Net). Ces options sont directement rajoutées dans le fichier Web.Config. (Fig. 2)

```

<configSections>
  <section name="microsoft.web.services2" type="Microsoft.Web.Services2.
Configuration.WebServicesConfiguration, Microsoft.Web.Services2, Version=
2.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
    
```

```

</configSections>
...
<webServices>
  <soapExtensionTypes>
    <add type="Microsoft.Web.Services2.WebServicesExtension, Microsoft.
Web.Services2, Version=2.0.0.0, Culture=neutral, PublicKeyToken=31bf
3856ad364e35" priority="1" group="0" />
  </soapExtensionTypes>
</webServices>
    
```

Pour utiliser les fonctionnalités WSE côté client, il est nécessaire de référencer les assemblies correspondant à WSE. Nousinstancions ensuite notre classe proxy qui porte le nom de "Service1Wse" au lieu de "Service1". Cette nouvelle classe hérite de la classe Microsoft.Web.Services2.WebServicesClientProtocol qui supporte les jetons de sécurité. Nous créons ensuite un jeton de sécurité en lui précisant le nom et le mot de passe de l'identité à transmettre au Webservice. Enfin, nous rajoutons le jeton de sécurité à la collection des jetons de sécurité du contexte soap. Il sera lu par le service Web, afin de déterminer l'identité de l'application cliente.

```

using Microsoft.Web.Services2.Security;
using Microsoft.Web.Services2.Security.Tokens;

...

ServiceDemo.Service1Wse ws= new ServiceDemo.Service1Wse();

UsernameToken Token = null;

Token = new UsernameToken("@peyrusse", "motdepasse", PasswordOption.
SendPlainText);
ws.RequestSoapContext.Security.Tokens.Add(Token);
ws.WsTest();
    
```

Nous constatons que l'en-tête SOAP transmis a été modifié, et intègre un UserNameToken (le nom d'utilisateur, le mot de passe) et la date de création. Le mot de passe apparaît en clair, et peut être caché en utilisant le paramètre PasswordOption.SendHashed pour l'encrypter.

```

<wss:Security soap:mustUnderstand="1">
  <wss:UsernameToken xmlns:wsu="http://docs.oasis-open.org/
wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id=
"SecurityToken-c726d91d-61fe-4e90-b834-358c078fd3ea">
  <wss:Username>peyrusse</wss:Username>
  <wss:Password Type="http://docs.oasis-open.org/wss/2004
    
```

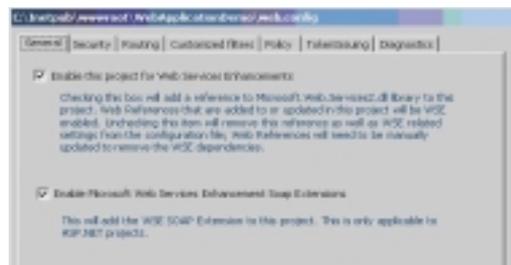


Fig. 2
Activation
des WSE
au sein d'une
application

```

/01/oasis-200401-wss-username-token-profile-1.0#PasswordText" >mot
depasse</wsse:Password>
  <wsse:Nonce>8u3w/SHU1o4LAdjgh3d1KQ==</wsse:Nonce>
  <wsu:Created>2004-09-08T14:52:11Z</wsu:Created>
</wsse:UsernameToken>
</wsse:Security>

```

Pour vérifier l'identité de l'utilisateur à l'origine de la demande, le service Web récupère le contexte Soap courant. Il parcourt la liste des jetons de sécurité, afin de rechercher ceux de type UsernameToken, et vérifie que le nom d'utilisateur transmis est bien celui du jeton. Si l'utilisateur est bien authentifié à partir de comptes Windows, alors WSE renseigne la propriété principale de l'objet UsernameToken avec son nom. Nous utilisons cette propriété pour vérifier avec la méthode IsInRole si l'utilisateur appartient au groupe " Admin " de la machine locale.

```

SoapContext ctx = RequestSoapContext.Current;

foreach (SecurityToken tok in ctx.Security.Tokens)
{
    if (tok is UsernameToken)
    {
        UsernameToken user = (UsernameToken)tok;

        if (user.Username == @"peyrusse")
        {
            if (user.Principal.IsInRole(System.Net.Dns.GetHostName() + @"\Admin"))
                return "rôle administrateur";
        }
    }
    if (tok is X509SecurityToken)
    {
        ...
    }
}

```

Sécurisation d'un service Web avec un BinarySecurityToken

Le mécanisme mis en place précédemment, ne permet pas de s'assurer de la non-corruption du message durant le transport. Il est alors nécessaire de signer notre message SOAP avec une signature numérique telle qu'un certificat X.509. Vous pouvez utiliser l'outil fourni avec WSE, permettant de visualiser les certificats disponibles sur une machine (wse-certificate2.Exe). Pour notre test, nous utiliserons un certificat de test installé par WSE. Un certificat est identifié par une clé codée en base 64, dans notre cas la clé est "gBfo01471M6cKnTbbMSuMVvmFY4=".

Nous définissons une méthode qui retourne un jeton de type X509SecurityToken. Cette méthode récupère un objet X509CertificateStore qui contient un ensemble de certificats, puis recherche le certificat identifié par le paramètre KeyId. Enfin, il retourne un jeton constitué à partir de ce certificat.

```

private X509SecurityToken GetX509Token(string keyId, string storeId)
{
    X509CertificateStore store = X509CertificateStore.LocalMachineStore (storeId);

```

```

store.OpenRead();
X509CertificateCollection certs = store.FindCertificateByKeyIdentifier
(Convert.FromBase64String(keyId));
    store.Close();
    return new X509SecurityToken(((X509Certificate)certs[0]));
}

```

Nous référençons l'assembly permettant de manipuler des certificats X509. using Microsoft.Web.Services2.Security.X509;

Nous utilisons la classe MessageSignature, afin de générer une signature numérique à partir de notre jeton de sécurité X509, puis nous nous servons pour signer notre message SOAP.

```

X509SecurityToken X509Token =
GetX509Token("gBfo01471M6cKnTbbMSuMVvmFY4=", X509Certificate
Store.MyStore);

ws.RequestSoapContext.Security.Tokens.Add(X509Token);
ws.RequestSoapContext.Security.Elements.Add(new MessageSignature
(X509Token));

```

Nous constatons, en exécutant le code qu'un élément BinarySecurityToken a été ajouté dans l'en-tête SOAP de notre message. L'attribut " ValueType " indique que le jeton est de type X509v3, l'attribut " EncodingType " précise la méthode d'encodage, en l'occurrence " Base64Binary ".

```

<wsse:BinarySecurityToken ValueType="http://docs.oasis-open.org/wss
/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" Encoding
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-soap-message-security-1.0#Base64Binary" xmlns:wsu="http://docs.
oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd" wsu:Id="SecurityToken-c822f48c-e95f-4066-86d9-
5d38dbb2cea6">MlIBxDCCAW6gAwlBAGlQYpjr4Fok3IFNSd3lj6ltzA
NBgkqhkiG9w0BAQQFADAWMRQwEgYDVQQDEwI290IEFnZW5j
eTAeFw0wMzA3MDgxODQ4MTBafw0zOTEyMzE5MzU5NTIaMB8xH
TAbBgNVBAMTFdTRTJRdWlja1NOYXJOU2VydMvYmIGfMAOGCSqGSI
b3DQEBAQUAA4GNADCBiQKBggQDLkqglArlnukRIDwXbcjN3zxfjeald
+lvfyD5o35pUjptkPwPXmApScr8UVQxB5JDRSVIMz1IUQ6CBFLGIAQ
OpbPKn2oul3VmKAF9nRQf9PLU+biWozZXkhebya43D75r5+5NUq1R
bQiCC4qlobRqUdg6adujBY333wjy4YgwIDAQABo0swSTBHBgNVHQ
EEQDA+gBAS5AkIBh0dTwCNYSHCfMjRjORgwFjEUMBIGA1UEAxMLUum
9vdCBBZ2VuY3MCEAY3bACqAGSKEc+41KpcNfQwDQYJKoZIhvcNA
QEEBQADQQAGSGNKz1gZqbXN8JYIOPQM7ngkHfW1mQ88NRYA
DmoHw5A/rUJZDHAPs5HLSn3i5iXIRwT91v3SU6iuaAid+Mwyyq
</wsse:BinarySecurityToken>

```

Il existe d'autres moyens de sécuriser notre service Web : par exemple, crypter le contenu du message afin que seul son destinataire puisse le lire ; nous ne traiterons pas cette partie dans cet article. A priori, une version majeure WSE 3.0 est prévue avant la sortie d'Indigo.

■ Christian Peyrusse

Paris Expo s'appuie sur les Web Services

CAS D'ENTREPRISE

Paris Expo (Groupe Unibail) est le leader français des espaces dédiés aux expositions, congrès, séminaires... Il exploite 7 sites sur Paris (Porte de Versailles, Palais des Sports, Carrousel du Louvre, Champerret) et à La Défense (CNIT, Espace Grande Arche, Cœur Défense) : plusieurs centaines d'événements sont organisés tous les ans dans ces lieux, du congrès aux salons, comme le Mondial de l'Automobile.

Jusqu'en 2002, les différents sites possédaient leur propre système d'information. Il fallait donc unifier tout cela dans un unique SI. Paris Expo a opté pour un progiciel complet, dédié à ce genre d'activité permettant de la suivre de bout en bout : de la commercialisation des espaces aux organisateurs, des prestations aux exposants, jusqu'à l'édition des factures. Ce progiciel Magic Expo est conçu par l'éditeur Magic Software Enterprises. Il est interfacé à un datawarehouse et un système de comptabilité qui fut lui aussi unifié et changé en début 2004.

Web Services au cœur des commandes

Paris Expo souhaitait proposer un portail de saisie des commandes des prestations aux exposants, pour améliorer la qualité de service de la commande, grâce à la visualisation de son traitement en temps réel, et réduire les temps de traitement de ces commandes. Jusqu'à présent, celles-ci se présentaient sous forme papier, qu'il fallait saisir dans le SI métier ! Peu rapide, et générateur de pertes de temps et d'erreurs "C'est dans ce contexte qu'un portail fut mis en place pour que les exposants puissent commander des prestations directement en ligne" déclare Philippe Bouton, responsable du système d'information de Paris Expo. La mise en place d'une solution de vente en ligne de ces prestations était l'objectif initial et permettait d'éviter les risques liés à la ressaisie des commandes



Ce portail, totalement développé en interne (écrit en HTML et Javascript, avec Tomcat, Java et Oracle 8i pour la partie serveur) est connecté au SI métier, via des Services Web. Ceux-ci sont développés avec la technologie eDeveloper de Magic Software. " Nous avons besoin des informations en temps réel " précise M. Bouton. Les Services Web répondent à cette attente vitale. Les informations doivent transiter sans délai entre le portail et le SI (communication bidirectionnelle). Lorsqu'un exposant saisit une commande, celle-ci est transmise par un Service Web qui en assure l'intégrité avant de l'injecter dans Magic Expo ; cœur du SI métier qui traite alors ces données. Car la commande génère tout un mécanisme de traitements de suivi et de mise en œuvre de ces prestations. Par exemple : si un exposant désire 10 lignes de téléphones, l'outil doit affecter les numéros de téléphones automatiquement. Il faut que la communication entre le front et le back office soit performante et rapide. Pourquoi les Web Services ? Philippe Bouton explique " Notre partenaire Magic Software et nos équipes internes maîtrisent totalement la technologie Web Services. Ceux-ci permettent une réutilisation simple et une intégration rapide des processus métiers au sein du système d'information, gage d'un investissement pérenne. La sécurité et la performance apportées par cette technologie confortent ce choix à long terme".

Des performances à la hauteur !

Côté performance, cela n'a guère posé de soucis. Certains Web Services ont été adoptés afin de mieux répondre aux contraintes. La mise à disposition des premiers Services Web a été développée en moins de 20 jours. Depuis, les Web Services évoluent au fur et à mesure des besoins fonctionnels. "L'évolution des Services Web est rapide et aisée à déployer en environnement de production" se félicite M. Bouton. Au niveau de l'architecture, Paris Expo a ainsi, pu dissocier la partie client (le portail) de la partie serveur (le SI métier) et les relier par la couche indépendante constituée des Web Services. Cela a apporté une grande souplesse qui dans le futur permettra de s'ouvrir à d'autres environnements. " Nous avançons par étape, et avons toujours procédé en adaptant les Web Services au portail et au SI ; non pas l'inverse ! " poursuit M. Bouton.

Cette nouvelle architecture " orientée service " optimise la gestion des commandes, et les Web Services développés par Magic Software complètent intégralement les attentes fonctionnelles requises par Paris Expo. " Il est totalement acquis que le portail et les Web Services sont générateurs de gain de temps lors du traitement des commandes de prestations sur le portail ; les services apportés aux visiteurs et à nos clients sont, ainsi accrus chaque jour..."

■ François Tonic

ESIGELEC : de l'électronique à l'informatique

Créée en 1901 à Paris et spécialisée à l'origine dans les courants forts et systèmes électriques, l'ESIGELEC est installée depuis 1978 à Rouen, sous l'égide de la chambre de commerce et d'industrie et vient de s'insérer encore plus étroitement dans le tissu économique local, en rejoignant la technopole du sud de la ville. Généraliste, l'école forme néanmoins beaucoup d'informaticiens.



Claude Guillermet, Directeur général de L'ESIGELEC



Affiner les options informatiques

Claude Guillermet, directeur général de l'école souligne que l'enseignement de l'informatique au sein de l'ESIGELEC privilégie une "approche structurante", avec un fort aspect méthodologique. Après le tronc commun qui compte quelque 130 heures d'enseignement en informatique, les deux options informatiques sont "développement de projets et génie logiciel" (méthodes de conception d'applications, bases de données, Java, C++, Rational Rose) et "ingénierie de l'information et de la communication" (plus de Java, et les Web services...). Cette dernière option est bilingue, ce qui devrait à terme être le cas de toutes, précise Claude Guillermet. Pour équilibrer les demandes entre ces deux options, la seconde ayant eu la préférence des élèves pour son côté Internet, ce qui s'est traduit par une désaffection pour la première, le directeur général de l'école explique que ces deux options auront à partir de l'année prochaine un tronc commun, avant de proposer un choix entre une dominante architecture et l'autre dédiée aux applications.

D'autres options font largement appel à l'informatique : celle dédiée aux " réseaux et services télécoms " d'une part, celle " ingénierie des systèmes embarqués " d'autre part.

L'école propose deux mastères : L'un en management des systèmes d'information, en partenariat avec l'Ecole supérieure de commerce de Rouen, l'autre en informatique embarquée.

L'international obligatoire

Claude Guillermet souligne que l'école a fortement développé les aspects non scientifiques, qui représentent presque un quart du temps d'enseignement, contre 12% il y a quelques années. Au programme, notamment, la conduite et gestion de projet ou encore des approches métier : ingénieur R&D, ingénieur projet, ingénieur d'affaires, ingénieur à l'international, ingénieur qualité et chef d'entreprise. L'international tient également une part prépondérante : un stage ou une partie du cycle ingénieur doit être effectué à l'étranger. L'école compte une quarantaine d'universités ou écoles partenaires, sur tous les continents.

■ Carole Pitras

L'école en bref

Accès

Sur concours ou après le bac, en prépa intégrée. Également accessible aux titulaires de DEUG sur dossier+entretien.

Effectifs

100 enseignants vacataires, 250 diplômés par an, soit 750 élèves dans le cycle ingénieurs + 160 en prépa intégrée. Environ 60 élèves par an en apprentissage. L'école compte augmenter le nombre d'élèves par promotion, à 280 cette année, et 300 à terme.

Organisation des études

Trois ans pour le cycle ingénieur, répartis en 1 an 1/2 de tronc commun et 1 an 1/2 de dominante technologique. Ces dominantes sont au nombre de huit : trois en télécoms, deux en informatique et trois en systèmes industriels.

Débouchés

L'informatique est le premier débouché pour les diplômés de l'ESIGELEC : 36 % des diplômés y ont trouvé leur emploi, dans les SSI essentiellement. Viennent ensuite les télécoms (15 %). Le salaire d'embauche moyen pour la promo 2002 était de 32 000 euros.

Frais de scolarité

Environ 5 000 euros par an pour le cycle ingénieur, 3 630 en prépa intégrée

Business & Décision : décisionnel, CRM et portails collaboratifs

Dès son origine, Business & Décision a, son nom l'indique d'ailleurs, choisi de se focaliser sur le domaine du décisionnel. Au fil du temps, deux autres domaines ont complété ce métier d'origine : la gestion de la relation client (CRM, analytique notamment) et l'e-business, avec une spécialisation sur les portails collaboratifs ou d'entreprises, annuaires et méta-annuaires.



A noter dans ce dernier domaine, où l'Open Source est assez présent, que business & Décision a fait l'acquisition en novembre 2003 d'Aurora, une des plus anciennes SSLL. Sur l'exercice clos au 30 juin dernier, l'activité business intelligence a représenté 68 % de l'activité du Groupe, en croissance de 14 % par rapport à l'année précédente. La gestion de la relation client (15 % du chiffre d'affaires consolidé) et l'e-business (17 %) affichent une hausse de 4 % d'une année sur l'autre.

110 recrutements d'ici à juin 2005

Après avoir pratiqué un recrutement assez diversifié dans les années fastes, la société s'était durant la crise re-focalisée sur des profils plus " classiques " (écoles d'ingénieurs, Miage, DESS), et pour les projets de mise en œuvre d'outils de consolidations, de titulaires du DECF ou DESCF. Depuis un an, Patrick Bensabat indique que cette logique de recrutement plus varié est à nouveau à l'ordre du jour, avec des profils issus de formations économiques, ou de diplômés d'écoles de commerce. Il explique que si le temps passé à faire monter ces profils atypiques en compétence technique est un peu plus long, ils seront finalement des consultants maîtrisant bien le modèle Business & Décision, " qui privilégie l'expertise et la vision projet ".

Selon Alexandra Girard, Responsable ressources humaines, les profils les plus recherchés actuellement sont ceux orientés sur le CRM. Côté décisionnel, les compétences sur les solutions d'Informatica et de Microstrategy, ainsi que Datastage d'Ascential, sont assez

demandées, au côté des " classiques " du décisionnel et du reporting financier : Hyperion, Cognos et Business Objects. Et bien évidemment, tout ce qui touche aux bases de données, Oracle et SQL Server en premier lieu. Patrick Bensabat souligne que la notoriété de l'entreprise sur ses métiers lui permet de recevoir de nombreuses candidatures spontanées, et de garantir aux futurs recrutés de travailler sur des projets décisionnels ou CRM.

À l'instar de nombre de ses consœurs, Business & Décision embauche beaucoup ses stagiaires, qu'elle recrute de préférence dans les écoles d'ingénieurs assez généralistes, notamment l'ESIEA, l'EISTI, l'ESIM, Polytech Montpellier (ex ISIM), ou parmi les titulaires de DESS (à Lyon notamment). Alexandra Girard souligne que la société attend de ses futurs collaborateurs qu'ils sachent écouter, faire preuve " d'une certaine solidarité " sur un projet, être communicants et avoir la capacité de travailler en équipe. Une connaissance fonctionnelle est un plus, mais ne faut pas être trop fonctionnel non plus... bien que certains profils du monde de l'entreprise soient appréciés : directeur marketing, des achats, contrôleur de gestion ou directeur financier assurent une connaissance des fonctions de l'entreprise qui peut manquer aux informaticiens...

Le processus de recrutement est bien rodé : les candidats sont vus par les ressources humaines, un manager et en dernier lieu, mais systématiquement pour ceux qui ont franchi ces étapes par le P.d.g. de la société Patrick Bensabat, qui prend la décision finale.

En terme d'évolution de carrières, celles proposées chez Business & Décision sont assez classiques : manager, chef de projets (mais les places sont limitées) pour ceux qui savent



développer leurs qualités commerciales et relationnelles, expertise pour les moins... communicants. Patrick Bensabat insiste sur les nécessaires " ouverture d'esprit et intérêt pour le métier du client ", qualités " primordiales " qui amènent " plus de variété dans les possibilités d'évolution de carrière ".

■ Carole Pitras

Chiffres clés

58,4 M€ de chiffre d'affaires et 1 M€ de bénéfice net pour l'exercice 2004/2005. Présence en régions (agences et filiales à Aix-en-Provence, Caen, Grenoble, Lille, Lyon, Nantes), en Ile-de-France et en Europe (Grande-Bretagne, Belgique, Suisse, Espagne, Pays-Bas).

Effectif actuel :

704 personnes en fin d'exercice (30 juin 2004), en hausse de 16 % par rapport à l'année précédente

Recrutement :

110 consultants et experts sur l'exercice 2004/2005 (1er juillet au 30 juin)
70 expérimentés et 40 débutants, 70% à Paris, 30% en Province. Mobilité requise.

Profil "expérimentés" :

double compétence fonctionnelle (banque, finances, ressources humaines, marketing opérationnel...) et technique (bases de données, data warehouse, modélisation des données, alimentation, restitution...)

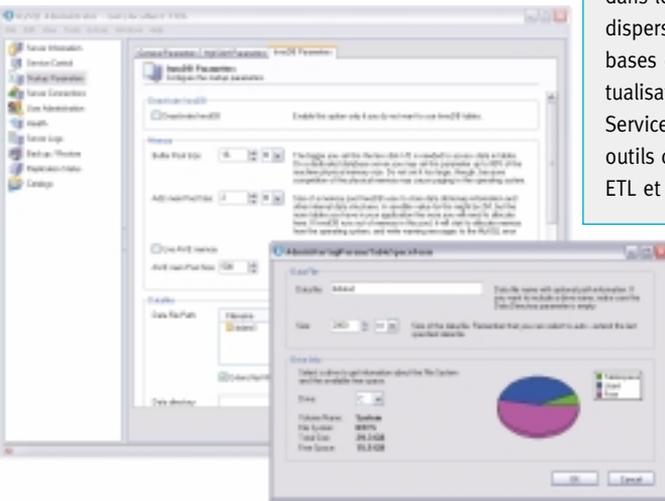
Profil "débutants" :

diplômés Bac+4/5 (universités et écoles d'ingénieurs), ayant effectué un stage de fin d'études dans le secteur de la banque ou des services
pour en savoir plus : www.businessdecision.com

là aussi, un SGBD de type Oracle demeure la règle. Car ce sont des outils éprouvés depuis des années. Dans le critique et la haute disponibilité, un outil libre de type MySQL n'a pas encore fait ses preuves, alors que pour des projets moyens / moyennes importances, il est adapté. L'environnement dans lequel tourne l'application joue aussi un rôle pour déterminer le SGBD. On peut toujours envisager des procédures d'intégration en environnement hétérogène, mais cela est lourd et coûteux. Si votre nouveau SGBD doit fédérer diverses sources de données, vérifier que le SGBD est capable de ce genre de choses! Cependant, n'oubliez pas que les besoins à l'instant T peuvent évoluer. Donc, préparez dès le départ le terrain. Il faut une SGBD capable de s'adapter rapidement à une modification des contraintes de l'environnement.

Critère 3 : le prix

La déferlante des SGBD libres a provoqué une baisse des prix de la part des grands éditeurs, avec une segmentation de l'offre selon le marché et l'utilisateur. À vous de bien faire attention au nombre d'utilisateurs ou de processeurs que vous achetez. Si vous devez utiliser votre SGBD sur plusieurs processeurs, souvent, il faut payer des licences supplémentaires. Quelle est la procédure de mise à jour ? Faut-il la payer ? Si oui, à quel prix ? Bénéficiez-vous d'un support, d'une assistan-



ce. Quel est le coût pour une maintenance annuelle de la base ? Les services de support/assistance / mise à jour sont souvent coûteux, soyez donc très vigilant. D'autre part, libre / open source, ne signifie pas gratuit. Chez MySQL, les versions pros et "entreprises" sont

Quelle base pour quelle application ?

Site Web : il n'y a aucune règle particulière. Si vous avez simplement besoin d'un stockage de données pour un site dynamique, une solution légère ou libre suffira dans 99 % des cas. Si vous êtes en environnement Microsoft, privilégiez une solution Microsoft, surtout en ASP. Si vous avez besoin d'une base critique, là, vous devrez opter pour les outils reconnus : Oracle, IBM, voire Microsoft, ou Sybase, ou des SGBD moins connus, mais fiables (SoftwareAG, Pervasive, Gupta...). Attention cependant si vous utilisez des fonctions avancées de cluster, répartition et reprise de charge, réplication ou encore le fonctionnement sur plusieurs processeurs.

Application Web : ici aussi pas de règles précises. Tout dépendra de la charge, des performances, de la quantité de données ou encore des fonctions de disponibilité souhaitées. Même remarque pour les applications critiques / non critiques que pour le site Web. Selon le type d'application Web (de langage utilisé, ou la plate-forme), le choix peut s'imposer de lui-même.

Applications mobiles : utiliser en priorité des SGBD dédiées à la mobilité. Les principales solutions du marché proposent une telle déclinaison adaptée aux applications mobiles. Là, préférez tout de même des solutions éprouvées. Ici peut se poser un problème de support du terminal, de plate-forme, voire de langage, à vous de bien vérifier que le SGBD est compatible (et vice versa avec l'application). Surtout, ne négligez pas la partie serveur, synchronisation et réplication des données. Soit vous pourrez mettre en œuvre les solutions de l'éditeur, soit passer par des solutions tierces. Ne négligez pas non plus la sécurité : sessions sécurisées, cryptages des données sur le terminal et lors des transferts, intégrité des données, etc.

Applications stand alone : une fois de plus, pas de règles précises. C'est selon les besoins de l'application locale. S'il s'agit uniquement d'un stockage statique ou avec peu de manipulation, vous pouvez opter pour des fichiers plats ou XML. Pour une utilisation plus "lourde", optez pour des moteurs de SGBD à implémenter ou des bases légères (Access a longtemp servi à ce type d'application). Pour du local exigeant, ou en mode client / serveur, on reviendra aux offres habituelles du marché ou alors à des SGBD incluant IDE / RAD.

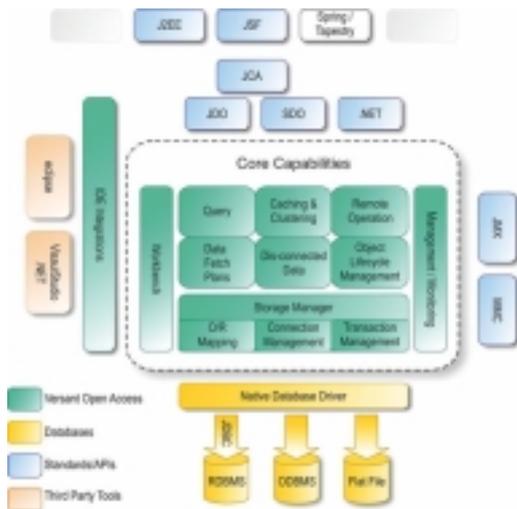
Intégration, fédération, virtualisation de données : un thème de plus en plus mis en avant dans les architectures applicatives distribuées, ou dans le cas d'entreprises géographiquement dispersées, ou dans le cas de filiales. Ici, en général ce sont des SGBD orientées très grosses bases de données (de type datawarehouse). Que ce soit la fédération de données ou leur virtualisation, cela nécessite des mécanismes lourds et complexes, souvent orientés XML (et Web Service). Privilégiez désormais une approche XML. L'intégration de données impose aussi des outils complexes, le libre n'offre rien de concret dans ce domaine. Ne parlons pas des concepts ETL et transformations qui peuvent se jouer au niveau middleware et applicatif.

payantes et toute une gamme de supports aussi.

Critère 4 : pérennité de l'éditeur

On n'y pense pas toujours, mais dans le cas d'un SGBD, ce critère n'est pas anecdotique. Par exemple, les utilisateurs qui avaient misé sur la base de données Java Cloudscape n'ont pas été déçus quand ce petit éditeur a été racheté par Informix au début des années 2000 qui, d'ailleurs, s'est ensuite fait avaler par IBM... Même si une solution innovante vient juste de

sortir d'un éditeur inconnu, évitez de vous jeter dessus sans avoir pesé le pour et le contre. Même remarque pour les grands éditeurs et leurs nouvelles versions, patientez quelque temps avant d'envisager une migration. Attendez les retours utilisateurs, testez la nouvelle release et la migration. Les solutions libres / Open Source sont aujourd'hui entrées dans le panorama normal des SGBD. Si pour les projets Web et autres applications de moyenne importance, elles conviennent, pour le critique et les très grosses bases, on préférera des valeurs sûres du marché, avec qui on possède le recul nécessaire. Certains types de



SGBD ; malgré un engouement certain, n'ont eu qu'un succès limité (ex. : SGBDOO ou SGBD 100 % Java).

On constate bien souvent que le choix d'un SGBD demeure conservateur, que ce soit pour une nouvelle base, pour un nouveau projet, ou pour migrer (cependant pour des projets spécifiques, vous serez amené à mettre en œuvre des solutions spécifiques). Même si cela n'est pas toujours le cas, on demeure avec des réflexes : SQL Server pour Windows, Oracle pour Unix, DB2 pour Mainframe, MySQL pour Linux.

Comment évaluer le coût ?

Le coût d'un SGBD et de la gestion du cycle de vie des données (et de la base) n'est guère simple à appréhender. Ici comme ailleurs, le coût de la licence n'est qu'un leurre. Si aujourd'hui, les éditeurs proposent des SGBD d'entrée de gamme gratuits ou peu cher, il peut exister des limitations fonctionnelles et dans la

Mesurer la performance d'une base de données

Le Transaction Processing Performance Council (www.tpc.org) est devenu au fil des années une association réunissant aussi bien des constructeurs informatiques que des éditeurs de bases de données, ayant pour but d'éditer régulièrement des résultats de tests objectifs mettant en scène des serveurs et des SGBDR. Quatre mesures ont été définies :

TPC-C : ce benchmark simule un environnement informatique complet dans lequel une population d'utilisateurs exécute des transactions sur une base de données. Ce test est axé sur des transactions classiques, constituées par des lectures/écritures d'enregistrements représentant une industrie quelconque qui gère des produits. Techniquement, des ordres complexes sont exécutés sur la base de données, comme le parcours de millions d'enregistrements, la vérification de l'intégrité des transactions (ACID) ou la contention possible lors de la mise à jour simultanée d'importantes quantités de données. La performance du test TPC-C se mesure en nouvelles transactions par minute (tmpC) et son prix par transaction (\$/tmpC).

TPC-H : ce benchmark est plus orienté décisionnel et consiste en l'exécution de requêtes complexes sur de très gros volumes de données, permettant de prendre des décisions critiques pour l'activité d'une entreprise. La métrique se mesure en requête/heure (QphH@size) et son pendant au niveau coût (\$/QphH@Size).

TPC-R : moins connu, ce benchmark est très similaire au précédent, mais autorise des optimisations complémentaires, basées sur la connaissance avancée des requêtes.

TPC-W : ce dernier benchmark est dédié au monde transactionnel du Web. La charge est testée dans un environnement de commerce électronique qui simule des sessions multiples lancées par des navigateurs Web, des générations de pages dynamiques, des objets Web persistants, etc. La performance est mesurée par le nombre d'interactions Web exécutées par seconde.

charge autorisée. Si vous êtes une PME/PMI, un indépendant, optez pour ce type de base de données. Pour un développeur " amateur ", les solutions libres sont intéressantes (sauf dans certains types d'applications). Outre la licence, les coûts cachés sont nombreux : support / assistance / évolution, intégration dans les projets, conception des bases, maintenance dans le cycle de vie de l'application, les contraintes de haute disponibilité. Sans comp-

ter la compétence des DBA et développeurs. C'est pour cela que toute la phase d'analyse des besoins du projet et des données est vitale. Si le SGBD choisi se révèle sous évalué par rapport aux besoins réels, les coûts explosent, car cela implique une modification de la conduite du projet (et donc des délais). Une surévaluation n'est guère mieux, car cela provoque un surcoût final. Dans le cas d'une migration d'un SGBD à un autre, ou d'une mise à jour vers une version plus récente, les coûts sont fluctuants. L'analyse du processus de migration / mise à jour devrait être scrupuleusement étudiée quelle que soit l'importance des données. Vérifiez aussi que la nouvelle base fonctionne dans l'environnement applicatif (soit le code ne change pas, soit il faudra redévelopper des morceaux).

On ne le répétera jamais assez : la licence n'est qu'une infime partie du coût global d'acquisition d'un outil, qu'il soit libre ou commercial. Ne choisissez pas un SGBD libre en arguant qu'il est libre. Ce raisonnement est faux et dangereux. Ayez une approche pragmatique, technique et fonctionnelle.

■ François Tonic

MaxDB 7.6 : l'ultime SGBD sous Linux ?

SAP et MySQL ont signé en 2003 un partenariat permettant aux deux éditeurs de développer une nouvelle base de données très puissante, sur les fondations du produit SAP-DB. MySQL continue toutefois à commercialiser 4 versions de son produit phare (versions 4.0 Standard, Max, Pro et Classic) mais recommande MaxDB 7.6 pour des applications très gourmandes en performance. Cette dernière version de MaxDB supporte dorénavant le 64 bits sur Linux et HP-UX, ainsi que de multiples fonctions de supervision, de clustering, de réplication, tout comme le framework de développement IBM Eclipse écrit en Java. Par conséquent, MaxDB est disponible dans de nombreux environnements. MaxDB se pose en sérieux concurrent d'Oracle, milieu de gamme sur Linux/Unix, avec des prix très agressifs : 49 \$ par utilisateur sur un serveur monoprocesseur (5 utilisateurs minimum).



Prévention et performances

N'importe quel système de commerce électronique comprend une base de données. Comme tous les autres composants d'une application, elle peut introduire des problèmes de fonctionnalité ou simplement de fiabilité au niveau de l'application. Elle a aussi le potentiel pour corrompre, écraser ou détourner l'utilisation de données vitales. C'est pourquoi il est important que ces problèmes soient détectés et éliminés aussi tôt que possible, pour réduire leur impact sur le fonctionnement de votre base de données et en fin de compte, de vos affaires. Voici une introduction aux méthodes de prévention de ce type de problèmes.

Les interactions base de données – application, vulnérables aux erreurs

Une communication efficace entre applications et bases de données est nécessaire au bon fonctionnement des systèmes d'entreprise. Même de petits programmes ou bases de données peuvent être à l'origine de problèmes tels que :

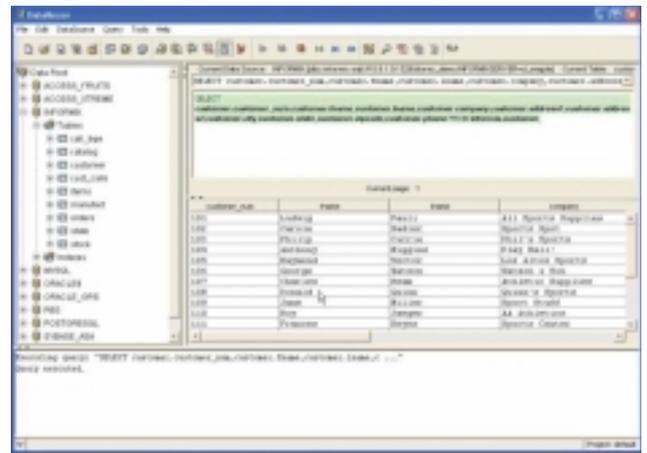
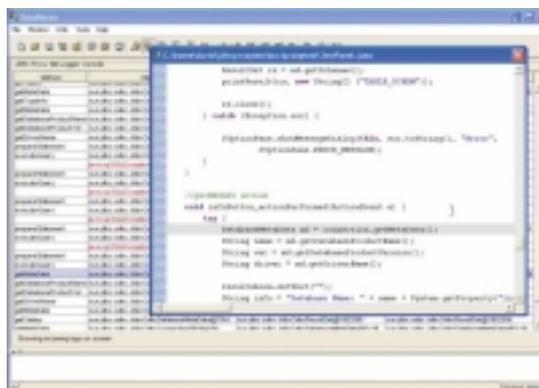
- Les goulets d'étranglement et problèmes de performance
- Une application qui se comporte de façon anormale
- Les plantages d'applications
- Le stockage de données inexactes, incomplètes ou inutilisables.

Les interactions application - base de données dans les systèmes d'entreprise constituent donc une articulation particulièrement vulnérable aux erreurs. En effet, avec l'accumulation de données, les modifications de structure ou design de SGBD, les bases de données changent fréquemment au cours de leur cycle de vie. À chaque fois qu'une nouvelle fiche est soumise, une fiche existante est modifiée, un index est retiré, un champ est ajouté, ce qui peut affecter la performance et/ou fonctionnalité du système. Chaque modification effectuée sur un composant d'application ou de middleware qui interagit avec la base de données augmente en même temps le risque de problèmes liés à la base de données.

Nous avons découvert que le meilleur moyen d'éviter les problèmes de performance et de fonctionnalité liés aux bases de données consiste à développer des tests dès le début du processus de développement. Il faut ensuite les exécuter et les étendre au fur et à mesure de l'évolution des applications et bases de données.

Pour que cela soit vraiment faisable, les développeurs ont besoin d'outils qui leur permettent de mettre en place un filet fiable pour soutenir les interactions applications – BD. L'automatisation doit leur permettre de vérifier les fonctionnalités et détecter les faiblesses cachées de cette articulation au plus tôt et sans perdre de temps.

Les développeurs peuvent dès lors corriger ces problèmes à un stade du développement pendant lequel les réparations sont encore faciles et peu chères, tout en appliquant des pratiques qui éviteront que des problèmes similaires se produisent par la suite.



Prévention et performances dans les requêtes.

Organiser la prévention

En général, le développeur est le membre de l'équipe qui définira le mieux comment les bases de données et applications devraient interagir et les conditions nécessaires à un bon fonctionnement : il est donc la personne la plus qualifiée pour développer une suite de tests qui va effectivement vérifier que ces conditions, notamment en matière de design, structure, intégrité des données, fonctionnalité et performance sont respectées.

Les équipes d'administrateurs et assurance-qualité réutiliseront et étendront ensuite ces suites de tests pour vérifier si le système en pleine évolution continue de remplir les conditions critiques.

Le département AQ hérite habituellement des suites de tests, lorsque les modifications des applications et BD réalisées par les développeurs ont été intégrées dans le système de test complet. À ce stade, l'équipe d'assurance-qualité utilise les suites de tests comme cadre pour leurs tests de non-régression, vérifiant ainsi que la migration n'a pas introduit de problèmes ou de modifications imprévues. Ils étendent également ces suites de tests pour contrôler les interactions application-base de données dans tout le système. L'administrateur reprend la suite de tests lorsque les modifications sur les bases de données et applications sont passées en production. En lançant l'exécution de cette suite de tests automatiquement à intervalles réguliers, il peut gérer la fonctionnalité et performance des interactions

bases de données – applications alors même que la BD en production évolue.

Une surveillance vigilante du système après déploiement est essentielle parce que la base de données change avec l'accumulation des données, les réglages, etc. et chaque changement, rend le système très vulnérable aux problèmes de fonctionnalité et de performance. Une gestion plus traditionnelle permet à l'administrateur de détecter les problèmes qui affectent les transactions dans la base de données de production, par une surveillance des statistiques de performance telles que les tampons, le temps de transaction, l'utilisation de la mémoire... mais cela ne suffit pas pour prévenir, détecter et corriger correctement les nombreux problèmes qui peuvent apparaître.

Miser sur la qualité

Compléter cette démarche avec une suite de tests, créée par les développeurs et étendue par l'équipe d'assurance-qualité, comporte plusieurs avantages significatifs :

- Cette méthode permet de toucher un spectre plus large des problèmes de fonctionnalité et de performance qui affectent communément le système (par exemple, détecter les champs supprimés involontairement ou les données mal formatées provoquant des problèmes de fonctionnalité de l'application). La gestion traditionnelle cible généralement la performance, mais la performance sera le dernier de vos soucis si le système réagit de façon imprévue, ou plante, suite à des modifications de la BD ou de l'application. En élargissant votre champ d'action, vous relevez le niveau d'abstraction de votre gestion sans pour autant abandonner les détails.
- Cela permet à l'administrateur bases de données

d'anticiper et corriger les problèmes avant qu'ils n'affectent les transactions et les utilisateurs. Avec une suite de tests complète pour outils, les administrateurs peuvent être proactifs et localiser / réparer les problèmes potentiels en faisant des tests sur un système isolé, ou en exécutant des transactions- tests sur le système de production. En comparaison, l'approche traditionnelle est forcément réactive ; il y a peu de place pour la prévention, lorsque les tests concernent simplement les transactions réelles des utilisateurs en production.

- Cela permet aux administrateurs de diagnostiquer et résoudre les problèmes remontés plus facilement. Les résultats rapportés par une suite de tests de prévention révèlent en effet quelles opérations de la base de données, quelles actions de l'application ou quels éléments de la BD correspondent à chaque problème détecté. Dans le cas d'un monitoring traditionnel, les résultats ne renvoient pas aux opérations / actions spécifiques des BD et applications et il est donc beaucoup plus difficile de localiser la source du problème. Ce niveau de gestion des applications et bases de données correspond à l'essence du travail réalisé par tous les membres de l'équipe qui implémentent et/ou contrôlent les interactions applications- base de données. Chaque développeur s'assure que ses segments d'application communiquent normalement avec la BD. Chaque développeur produit une suite de tests qui servent à détecter les problèmes liés à la base de données et qui pourraient affecter l'application par la suite. Les membres de l'équipe assurance-qualité exécutent et étendent les suites de tests des développeurs pour s'assurer que les interactions application-base de données continuent à fonctionner normalement une fois dans le système de test complet. Résultat, l'administrateur peut exécuter les suites de tests et déterminer rapidement si des actions doivent être entreprises pour maintenir une bonne interaction (par exemple, si il/elle remarque une mauvaise performance de récupération des données, il pourrait avoir besoin d'indexer des colonnes supplémentaires, retirer ou archiver les données anciennes, etc.).

■ Adam Kolawa

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

LE SGBDO

En réaction à la première partie du dossier, la société Versant a désiré s'exprimer.

Les applications développées au standard J2EE, sont confrontées à ce problème appelé communément "mapping objet relationnel" qu'on peut définir en une phrase "expliquer aux objets comment se retrouver hébergés dans une ou plusieurs tables, en un seul ou en plusieurs morceaux".

(...) Ce mapping, généralement évalué à 30 % du prix d'un projet, doit souvent être effectué par un expert Objet et SQL qui sait transformer un objet en requêtes avec jointures entre tables. Il est très coûteux en CPU et en performance, ce qui rend difficile la montée en charge des solutions.

Pour faire face au problème du mapping objet relationnel, beaucoup pensent qu'il y a seulement 2 solutions : le mapping avec le standard JDO ou le mapping propriétaire. C'est vrai que toutes les solutions de mapping ne sont pas équivalentes en terme de performances (cf le bench Torpedo sur <http://www.middlewared-research.com/torpedo/results.jsp>). Versant, éditeur de Versant Open Access JDO (ex. JDO Genie) et ancien distributeur de KODO JDO sait qu'en réalité le mapping est hétérogène. Il n'y en a pas un pareil et chacun a ses spécificités. Ceci dit, vous l'avez bien expliqué, il vaut mieux choisir des solutions de mapping au standard JDO car elles permettent d'interchanger les bases de données en fonction des contraintes techniques (SGBDR, SGBDO notamment). Mais dans votre dossier SGBD ière partie, vous ne parlez pas de la solution qui permet d'éviter complètement le mapping ! Et oui, il est possible de développer une application en la déchargeant complètement du poste mapping O/R. Comment ? Il suffit de la développer avec un SGBDO. Certaines applications chez Air France, France Telecom, EDF, Nortel, Natexis, Thales, Ministère de la Défense, Sagem. en ont fini avec ce "bouffe temps" qu'est le mapping. D'autres décident de faire du "O mapping" seulement sur leur prototype. Comme Versant est sous format JDO, les décideurs testent les performances de la solution "O mapping" avec différents SGBDO/R. Et vous que ferez vous contre ce fléau du développement de l'informatique d'aujourd'hui ?

■ Jean Claude Bellando

Directeur Commercial de Versant France, filiale du groupe Génitech

Datarecon : Un outil de cycle de vie

Parasoft propose l'automatisation de la plupart de ces étapes et d'autres tâches encore, avec Datarecon. Cet outil intervient depuis la conception de la base de données, jusqu'à son déploiement, et pendant sa croissance. Pour en savoir plus sur les stratégies de prévention et de contrôle des problèmes typiques de la base de données, consultez le livre blanc " Ensuring Database Functionality and Performance Throughout the Software Lifecycle" disponible gratuitement sur www.parasoft.com.

"XML est un format d'échange génial, mais pas de stockage"



Le marché du mapping évolue beaucoup. Nous avons voulu en savoir un peu plus, en rencontrant un des acteurs côté Java, Éric Samson, Cofondateur et responsable technologique du français Xcalia (anciennement Libelis), spécialiste du mapping basé sur JDO.

Programmez ! : L'accès aux données a toujours été un problème. Le mapping répond-il à cette problématique en simplifiant l'approche ?

Éric Samson : Tout le monde sait que l'accès aux données est complexe, compliqué. Les relations ne sont pas connues en Java. C'est une faiblesse des langages modernes. Dans Java ou C#, il n'y a pas ce type de concept. Il existe différentes solutions de prise en charge : le développement, un outil, un standard. Aujourd'hui, tout le monde ou presque fait de l'objet et du relationnel. Jusqu'en 2002, Java n'avait rien ou presque. Depuis, on dispose de JDO qui constitue un effort de standardisation sous Java. Ce problème d'accès aux données est tellement complexe que bien souvent, on ne s'y intéresse qu'à la fin d'un projet. Il met le doigt sur beaucoup de points : performances, couche d'accès aux données, simplicité du développement... JDO tente d'y répondre en fournissant une transparence et une simplicité d'accès. Le mapping prend du temps et peut poser un souci de qualité, sans compter les problèmes de performances. Des outils tels que LiDo sont là pour gérer l'accès, assurer les performances, intégrer des niveaux de caches. C'est un ensemble de fonctions que le développeur ne peut pas réaliser.

Programmez ! : Est-il possible de mesurer le ROI sur du mapping ?

Éric Samson : Des clients ont réalisé des études de ROI. En s'appropriant la technologie,

en moyenne, un code technique à la main représente 20-25 % des efforts du développement pur. L'accès aux données tourne entre 20 et 30 %. En utilisant un outil de mapping, on peut s'attendre à un facteur de réduction de 80 à 90 %. Il restera une phase de mise en œuvre. Le ROI dépend aussi du modèle des données.

Programmez ! : Ne pourrait-on pas avoir le réflexe de développer en interne son propre mapping, au lieu d'utiliser des outils existants ? La technologie n'est pas toujours très bien perçue par les entreprises et les développeurs.

Éric Samson : Le mapping n'est pas une technologie nouvelle. Il existe une centaine de solutions. Le marché est mature. Il y a quelques années, il y avait beaucoup de sceptiques. Les outils actuels sont capables de répondre à 99 % des besoins d'un projet. C'est du délire de vouloir tout faire soi-même. Le coût financier dans un gros projet serait élevé avec beaucoup de développement, et sans doute la conception d'un framework difficile à maintenir. Un framework de mapping est aussi complexe qu'un moteur de SGBD. Le cœur de LiDo, par exemple, représente 16 000 classes !

Programmez ! : L'intégration du XML dans les SGBD est désormais systématique. Une divergence de vues semble exister sur l'usage à faire du XML. Pour les uns, il doit se limiter aux flux de données, pour les autres, surtout

des éditeurs, veulent le transformer en format de stockage. Que pensez-vous de ce débat ?

Éric Samson : Vaste sujet. L'énorme apport du XML concerne l'échange des données et non le stockage ! Un flux XML est mort. Il faut récréer la sémantique. SDO essaie de standardiser l'intégration des flux XML dans le système d'information. N'oubliez pas que XML n'est pas transactionnel. On ne connaît pas son modèle. Cela nécessite de réaliser du mapping dynamique. La plupart du temps, on fait du Java sans le moindre modèle métier Java. On se tourne vers XML pour les données, mais ce sont des données "mortes". C'est une tendance. La programmation Java demeure complexe. Cela peut fonctionner dans de petites applications. XML est un format d'échange génial, mais pas de stockage. Il est trop verbeux. Cependant, pour des choses simples, on peut voir des projets utilisant XML comme format de stockage, certains de nos clients l'utilisent par exemple dans des applications de domotique.

Programmez ! : Depuis le printemps dernier, de vives discussions tourmentent autour du modèle de mapping de Java. Certains voudraient imposer EJB 3.0 au détriment de JDO. Les autres voudraient pousser Hibernate. Qu'en est-il ?

Éric Samson : Cela a cristallisé un certain nombre de querelles du monde Java. L'approche d'EJB 3.0 valide finalement l'approche de JDO. Cependant, EJB 3.0 est limité à un fonctionnement par container. On reconnaît aujourd'hui que les EJB 1.0 et 2.0 furent des aberrations. JDO 2.0 répond aux nombreuses critiques faites dans les précédentes versions. Les choses sont moins évidentes que ce qui a été dit à la dernière JavaOne. JDO est un bon standard et on va se battre pour SDO.

■ *Propos recueillis par François Tonic*

✓ **L'actualité** en ligne
www.programmez.com

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

IBM devance Microsoft sur l'intégration de .NET à la base de données

Force est de reconnaître que Microsoft s'est royalement fait griller la politesse par IBM : la première base de données à incorporer le ".net framework", ou plus exactement son environnement d'exécution, la CLR n'est pas SQL Server 2005 (nom de code Yukon) mais DB2-UDB 8.2 (nom de code Stinger). Et Microsoft aura beau retourner ses arguments marketing, l'implémentation de la CLR par IBM dans DB2 n'a pas grand-chose à envier à ce qui était présenté comme l'une des innovations clés de Yukon.

Le principe d'intégrer un environnement d'exécution dans la base de données n'est pas nouveau, puisque Oracle et IBM, entre autres, ont depuis longtemps incorporé une machine virtuelle java à leur base. En adoptant la CLR, IBM ne fait donc que s'ouvrir à un nouvel environnement, celui de Microsoft. Très pratique pour écrire des fonctions exploitant des calculs arithmétiques complexes (en astronomie, navigation routière ou aéronautique, calculs financiers, architecture, etc.), l'ajout de la CLR permet de programmer en VB ou C# plutôt qu'en TSQL ou SQL/PL. Plus lisible, le code est plus facile à écrire et s'exécute plus rapidement. Dans des cas complexes de manipulations des curseurs de la base, l'utilisation de VB/C# et ADO.NET est aussi plus appropriée que le langage SQL qui a davantage été conçu pour la manipulation d'ensembles de données. Enfin, en intégrant la CLR dans la base de données, Microsoft et IBM offrent également un accès complet à .NET et donc à son jeu de classes plus riche que le langage SQL notamment.

Faire aussi convivial et intégré que Microsoft

Dans DB2-UDB 8.2, IBM a travaillé sur de nombreux aspects pour faire une base aussi conviviale et intégrée aux environnements de Microsoft que l'est SQL Server. Ainsi, les nouveaux outils de productivité (comme l'éditeur de script et de requêtes SQL ou le gestionnaire des connexions et objets de la base) s'intègrent parfaitement à Visual Studio .NET 2003 et même Visual Studio 2005 (Whidbey) et adoptent la même philosophie que les fenêtres d'outils développées par Microsoft pour SQL Server. Par ailleurs, un nouveau fournisseur de

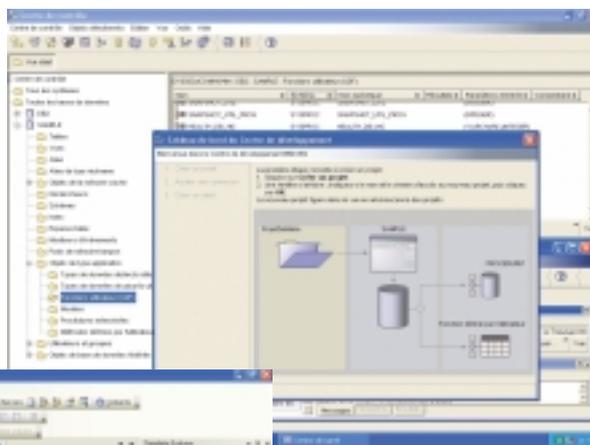
données natif (DB2.NET Data Provider) est prévu pour toutes les déclinaisons de DB2 (y compris sous Linux, AS400 et autres systèmes 390). Il favorise une bien meilleure intégration aux applications .NET que ne le permettait l'ancien driver ODBC et devrait contribuer à des meilleures performances.

Quelques différences significatives

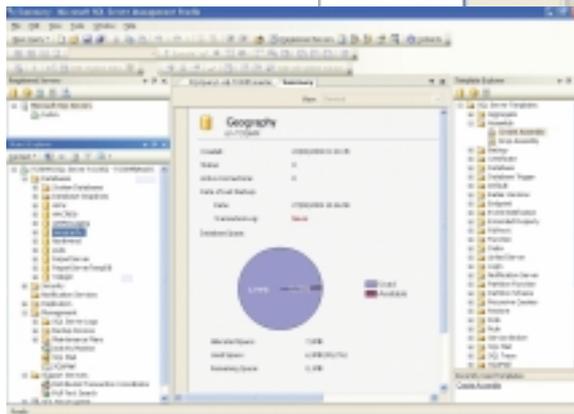
On constate toutefois quelques variations entre les deux implémentations. Ainsi, DB2 et SQL Server permettent l'écriture de procédures stockées et d'UDF (User Defined Functions) en VB ou C#. Mais seul SQL Server étend cette possibilité aux triggers. Cette limitation peut être contournée dans DB2, par un appel de procédure stockée depuis un trigger. De même, les UDTs (User Defined Types) ne peuvent être écrits qu'en SQL dans DB2. Dans SQL Server 2005, en revanche, ils peuvent être de types définis pour la CLR et, dès lors, être enrichis de notions d'objet et donc bénéficier des

méthodes .NET. A ces différences, il faut encore ajouter les fonctions de debugging avancées de SQL Server 2005 que le code soit en SQL, VB ou C# qui, chez IBM s'arrêtent au SQL ; des versions de CLR différentes (CLR 1.1 pour IBM, CLR 2.0 pour Microsoft, version optimisée en performances) ; et des manipulations plus simples chez Microsoft qui profite de la très forte intégration entre son atelier de développement et sa base de données pour, par exemple, limiter à quelques clics l'implémentation de code VB ou C# dans la base. La même opération chez IBM implique la création d'un script de linkage pour que le projet référence l'assembly IBM.Data.Db2.dll. La routine compilée doit être ensuite placée dans le dossier "SQLLIB\FUNCTION". Enfin, il faut enregistrer l'assembly finale dans la base de données, à l'aide des instructions SQL 'CREATE PROCEDURE' ou 'CREATE FUNCTION'. En d'autres termes, si le résultat au final est le même, l'opération est plus simple avec SQL Server 2005.

■ Marie Varandat



▲ IBM DB2 8.2 (Stinger) est attendu en version finale avant la fin de l'année.



◀ SQL Server 2005 (Yukon) est attendu en bêta 3 fin 2004 et en version finale vers le milieu du premier semestre 2005.

Panorama des bases de données

4D 4Th Dimension Server 2004 <i>Windows, MacOS</i>	Relationnel	Base faisant partie d'une plate-forme complète dotée des mécanismes XML (XSLT, CSS) originaire de l'Univers Mac et pour laquelle il existe de nombreuses solutions métier.
Borland InterBase <i>Windows, Solaris, Linux</i>	Relationnel	Cette base de données distribuée par et avec les outils Borland, elle respecte la norme SQL/92 à la lettre et reste une solution cross plateforme pratique pour tous ceux qui développent sous les outils Borland.
Computer Associate Ingres r3 <i>Windows, Linux, divers Unix</i>	Relationnel	La base Ingres est désormais en Open Source (dans une licence propre à CA, qui lui permet de retirer son logiciel de l'Open Source à tout moment). Méconnue, elle reste une base très moderne et plus évoluée que la plupart de ses concurrents Open Source : unicode, partitionning des tables, triggers, réorganisation en ligne, etc.
Gupta SQLBase 9.0 <i>Windows, Linux, Netware</i>	Relationnel	La plus ancienne (elle date de 88) des bases de données SQL pour PC est toujours là, ouverte à Java comme à .NET. Totalement autoadministrée, elle a une faible empreinte mémoire qui la rend bien adaptée à l'embarqué.
IBM DB2 Universal Db 8.2 <i>Windows, Linux, AIX, mainframes et z/OS</i>	Relationnel	Annoncé comme nouveau leader du marché SGBD (toutes plates-formes confondues) DB2 a su évoluer. Sa nouvelle version "Stinger" dame même le pion à Microsoft sur l'intégration de la CLR .NET. Elle permet surtout à IBM de rattraper son retard en matière d'automaintenance de la base et d'outils d'optimisations automatiques. Le problème d'IBM réside aujourd'hui dans la trop grande variété des solutions DB2 qui permet certes une couverture élargie sur tous les marchés, mais handicape la lisibilité de l'offre.
Intersystem Caché <i>Windows, OpenVMS, Linux, div. Unix</i>	Objet	Certainement la base "orientée objet" la plus connue, d'autant qu'elle demeure interrogeable en SQL. Elle expose même ses classes en Web Services en quelques clics.
Microsoft SQL Server 2000 / 2005 <i>Windows NT/2000/2003 Windows 98/2000/XP</i>	Relationnel	SQL Server a popularisé les principes d'automaintenance, d'optimisations automatiques et d'intégration "tout-en-un". Performante, apte à supporter les projets les plus lourds, SQL Server a également démocratisé la BI. La version 2005, actuellement en bêta, ajoute un type XML accessible en XQuery, un service-broker intégré, l'hébergement natif des Web-Services, l'intégration de la CLR .NET, et une approche novatrice des problématiques de Business Intelligence. Il existe une édition gratuite limitée dénommée MSDE, prochainement remplacée par SQL Server 2005 Express.
MySQL Standard & Max 5.0 <i>Linux, Windows, Solaris, FreeBSD, MacOSX, HP-UX, IBM AIX, QNX, Netware, SGI Irix, DEC OSF</i>	Relationnel	La base de données "Open Source" fétiche du monde Web/Apache ne veut plus jouer les seconds couteaux. Le support du clustering marque une avancée capitale en ce sens. La version 5 apporte des fonctionnalités très attendues : procédures stockées, des vues modifiables, et même des triggers rudimentaires. Malgré l'apparition de MySQL administration, les outils de gestion demeurent limités.
ObjectStore ObjectStore database 6.2 <i>Windows, Solaris, HP-UX, AIX, Linux</i>	Objet	Cette base de données objet, particulièrement adaptée au C++ et à Java, est réputée pour son support de Corba (et de Com/Dcom) ainsi que pour son architecture "Cache-Forward" et la flexibilité du modèle de programmation qu'elle induit.
Oracle Oracle DB Server 10g <i>Windows, Solaris, HP-UX, Linux, z/OS</i>	Relationnel	Avec la version 10g, Oracle se veut plus que jamais le SGBD des architectures hautement clustérisées (ce qu'Oracle appelle un Grid). Si elle reste la référence sur tous les projets d'envergure où montée en charge et haute disponibilité sont les éléments cruciaux, Oracle DB tente aujourd'hui de reconquérir ses parts de marché perdues auprès des PME/PMI avec des versions financièrement plus accessibles. L'intégration XML (avec un type dédié et interrogations en XPath) devrait également être améliorée dans la R2 par l'apparition des requêtes en XQuery.
Pervasive Pervasive SQL V8 <i>Windows, Netware, Linux</i>	Relationnel	SGBDR orienté TPE / PME / PMI, facile à déployer et optimisée pour les projets de taille réduite (sur architecture matérielle restreinte). Encore compatible DOS côté client.
PostgreSQL PostgreSQL 8.0 <i>Windows, Linux, divers Unix</i>	Relationnel	L'autre grande référence du monde "Open Source". S'adapte à tous projets de petites et moyennes importances. Bien que moins répandue et moins populaire que MySQL, elle n'a techniquement rien à envier à cette dernière.
Software AG Tamino 4.2 <i>Windows, Solaris, AIX, HP-UX, Linux</i>	XML	La plus connue des bases XML natives. Sérieusement concurrencée par les fonctionnalités XML d'Oracle, SQL Server 2005 ou DB2. Elle reste néanmoins la mieux adaptée au stockage de documents XML (davantage que des données XML).
Sybase ASE 15 <i>Windows, AIX, HP-UX, Linux, Solaris</i>	Relationnel	Sous la compétition, Sybase s'est réfugié sur les marchés niches. La récente gratuité de la version Linux ne devrait rien changer à l'affaire. Pourtant la v15 (attendue en 2005) supporte nativement XML et les Web Services. Sybase se focalise aujourd'hui sur la mobilité des données et propose en la matière des systèmes de synchronisation parmi les plus versatiles et les plus aboutis. La société dispose également d'une base orientée décisionnel qui permet d'éviter les moteurs OLAP (Adaptive Server IQ) dont la particularité principale est l'indexation des champs à la volée.
Versant FastObjects <i>Java, .Net</i>	Objet	Ce système de gestion de données objet, plutôt connu dans l'univers Java, s'est fait récemment remarquer parce qu'il offre aujourd'hui une vraie solution de persistance transparente pour .NET
Apache Xindice 1.1 <i>Windows, Linux, divers Unix</i>	XML	C'est la base de données "native XML" en OpenSource. Issue du projet dbXML, elle utilise XPath et XUpdate comme langage de requête.

Panorama des outils complémentaires

Dans la première partie de ce dossier (Programmez ! 68), nous vous avons parlé du fait que la base n'est pas (ou n'est plus) un élément isolé. Tout une panoplie d'outils permet de contrôler telle ou telle fonction, d'optimiser les requêtes, de surveiller telle relation ou tel objet, etc. Toutes les parties de la base de données peuvent être surveillées et

analysées. Voici un petit panorama sur ce genre d'outils, bien entendu, la liste proposée est loin d'être exhaustive. Vous trouverez des outils commerciaux et libres. Les SGBD leaders possèdent bien souvent toutes les fonctions que l'on peut trouver dans des outils annexes. À vous, comme toujours, de savoir si la fonction standard répond à vos attentes ou non.

Modèle / Développement

La plupart des SGBD ne possèdent aucun RAD ou IDE. L'offre de modélisation et de développement orientés données / SGBD est très large et couvre quasiment tout type de base. Cela comprend aussi les fonctions de tuning et d'optimisation.

Éditeur	Nom de produit	Commentaires
Allround automations	PL/SQL Developer	IDE pour développer en Oracle PL/SQL
Charonware	CaseStudio 2	Outil de modeling et de reporting (Oracle, SQL Server, MaxDB, MySQL)
Computer Associate	AllFusion Erwin Data Modeler	Outil de modélisation des bases
Datanamic	DeZign for databases	Outil de développement basé sur des diagrammes
Ecobilan	TEAM	Modélisation et gestion. Pour très grosses bases de données
Embarcadero	ER/Studio	Outil de cycle de vie pour SGBD, module de modélisation
FabFORCE.NET	DBDesigner	Conception visuel des bases de données. Open Source.
IDB Consultants	SQL Object Builder	Outil pour développer et déboguer Oracle PL/SQL
Oracle	HTML DB	Outil de développement Web pour base de données Oracle
SoftVelocity	Clarion ASP	Permet de publier sur le Web une base de données.
Sybase	PowerAMC	Outil de modélisation du système d'information, avec génération de la base
TheKompany.com	Data Architect	Modélisation de base. Supporte MySQL, DB2, InterBase, PostgreSQL.
VbeXpress	VbeXpress.NET	Solution complète .NET avec support SQL Server, DB2, Oracle, Access et des procédures stockées.
Visual Soft	Visual SQL-Designer	Construction visuelle de requêtes SQL

Administration / Performance / réplication

L'administration et tout ce qui gravite autour (réplication, monitoring, performance) est une des clés de la bonne intégration des données dans son environnement et de la gestion du cycle de vie des données.

Éditeur	Nom de produit	Commentaires
Compuware	DevPartner	Gamme d'outils de debug et d'optimisation pour Oracle, Sybase, SQL
DataMirror	High Availability Suite	Outil de haute disponibilité
Embarcadero	DB Artisan, ER/Studio, Rapid SQL	Pour l'administration des bases en entreprise
Gupta	Report Builder	Pour du reporting et quering de base de données.
Hit Software	DBMoto	Réplication de données entre DB2/400 et SQL Server – Oracle
Keptool	Keptool PL/SQL Debugger	Pour détecter et résoudre des erreurs dans une base de données (sous Oracle)
MySQL	MyDBA	Contrôle les bases, détection de problèmes, suivi au quotidien.
MySQL	MySQL Administrator	Administration graphique pour MySQL
Parasoft	Datarecon	Outil automatique de vérification et de monitoring afin de prévenir toute pollution des données
Parasoft	Data solution	Prévention d'erreur pour SGBD, s'intègre dans le cycle de vie des données.
Pervasive	Pervasive AuditMaster	Outil d'audit et de surveillance dédié à Pervasive SQL
Pervasive	Pervasive DataExchange	Outil de réplication pour Pervasive SQL
Progress Software	Trading Accelerator	Outil de haute disponibilité. Dédié au monde de la finance
Quest Software	Spotlight	Outil de diagnostics temps réel
Quest Software	TOAD	Gamme d'outils d'administration et de développement pour SGBD
Quest Software	Quest Central	Outil de gestion, de tests et de monitoring pour Oracle, SQL Server, Sybase
Quest Software	ShareFlex	Outil de réplication
TheKompany.com	dbRADER	Outil d'administration et de monitoring

Editeurs : A la pointe de la technologie

Même si quelques grands noms s'y taillent la part du lion, l'innovation dans les bases de données est loin d'être leur monopole. L'offre, multiple, inclut des noms comme 4D, File Maker, Caché, Pervasive etc... Nous avons demandé à 3 éditeurs de décrire leur vision de l'évolution de la technologie.

CACHÉ, l'alternative reconnue

Les SGBD traditionnels, limités par le modèle relationnel, se heurtent à deux écueils : le mapping O/R qui éloigne du modèle objet et engendre un surcoût de développement, et l'absence de scalabilité (volumétrie ou nombre d'utilisateurs) qui se traduit toujours par une augmentation des ressources matérielles. Avec CACHÉ, SGBD post-relationnel, ces problématiques disparaissent. Depuis 1978, InterSystems est le spécialiste de la persistance à très hautes-performances du RAD et de l'intégration. Reconnu mondialement comme le SGBD le plus performant du marché, CACHÉ gagne aussi du terrain sur les serveurs d'applications WEB grâce à son environnement de développement RAD intégré.

UML au cœur des données

Ses partenaires et utilisateurs apprécient notamment l'administration transparente de CACHÉ : ex. copier la base d'une plateforme à une autre, indépendamment des OS/architectures. La migration automatique de SGBDR est aussi assurée, procédures stockées comprises. Autre originalité, les index bitmap transactionnels offrent les performances OLTP au monde décisionnel, sans intermédiaire ETL. Orienté Objet, CACHÉ accepte une modélisation UML jusqu'à la persistance, interne ou externe (passerelle vers des SGBDR). La projection native de ce modèle vers des clients Java/.Net/EJB/C++ rend enfin possible un accès objet à un SGBD.

Téléchargement gratuit

L'arborescence XML est un excellent partenaire pour le moteur multidimensionnel CACHÉ : import/export XML par simple héritage. La publication de Services Web (SOAP/WSDL) devient instantanée. CACHÉ se télécharge gratuitement en version complète et illimitée sur www.intersystems.fr

PERVASIVE : l'intégration

La principale tendance émergente dans le domaine des bases de données est l'intégration. Aucune application n'est isolée ; les entreprises exigent toujours davantage de connexion des applications et un partage facile des données avec les systèmes existants. Les décideurs peuvent avoir une vision à 360° de leurs données. En réponse à cette tendance, certains fournisseurs de bases de données ont commencé à élargir leur offre pour y intégrer une technologie d'alimentation ETL (Extract/Transform/Load) de base.



Les atouts de Pervasive

Avec la génération de produits actuels, il subsiste encore un écart important entre ce dont le développeur a besoin et ce qui est disponible. Les outils actuellement disponibles conviennent bien pour les migrations sur la base de projets, mais ne sont pas conçus pour être intégrés dans les applications afin de fournir des capacités d'intégration complètes. Les développeurs se retrouvent encore souvent contraints de programmer du code spécifique, non lié à leurs activités principales et coûteux, pour créer des ponts entre les systèmes. Pervasive propose désormais sa suite intégrable de solutions d'intégration de données qui permet aux développeurs de créer des applications qui se connectent de façon transparente dans les systèmes critiques existants des entreprises. Pour la première fois, les développeurs peuvent accélérer le développement d'applications permettant l'intégration. Fini les ponts entre systèmes différents nécessitant la programmation de code spécifique coûteux.

<http://welcome.pervasive.com/telechargez>

EMBARCADERO : gestion du cycle de vie des données

Le marché des bases de données connaît aujourd'hui une profonde mutation, essentiellement expliquée par l'explosion des données hétérogènes au sein des entreprises. Ainsi, la croissance des données au sein des SI double tous les 2 ans, avec un taux annuel de 42% (IDC). Cette croissance s'explique notamment par l'augmentation des nouveaux types de contenus digitaux, des obligations de conformité, etc. Enfin, les fusions & acquisitions créent des environnements hétérogènes ou des technologies comme Linux, les services Web, XML qui font naître de nouvelles complexités dans la gestion des bases de données. Fort de ce constat, Embarcadero offre aujourd'hui une réponse unique à ces problématiques en proposant un Portfolio de produits entièrement dédié à la gestion du cycle de vie des données.

Une réponse unique

Plus que tout autre élément, l'expertise d'Embarcadero s'explique par la création d'applications innovantes développées de bout en bout par des professionnels utilisateurs de bases de données. Cet élément fondamental nous permet de proposer des solutions pouvant interpréter et influencer les données capitales de l'entreprise pour accroître son avantage concurrentiel tout en améliorant l'accessibilité et la pertinence des informations. Les solutions d'Embarcadero accompagnent quotidiennement les professionnels de bases de données en réduisant la complexité de leurs tâches de gestion, en améliorant leur productivité et en proposant des outils de veille préventifs permettant d'éviter la chute de systèmes. Enfin, pour continuer à proposer des développements industriels synonymes de fiabilité, Embarcadero investit plus de 30% de son CA en R&D, dans un secteur où la moyenne pratiquée est en général de 22%.

C-JDBC, une alternative Open Source de clustering

D'un point de vue architectural, un cluster est constitué d'un groupe de serveurs ou d'instances, utilisé pour former une sorte de serveur virtuel, vu comme une seule et même instance pour les applications qui l'utilisent.

Dans une infrastructure de type Internet, et plus spécifiquement J2EE, le clustering s'opère à différents niveaux. On peut, en effet établir des clusters au niveau des frontaux HTTP, des serveurs d'applications, mais aussi des serveurs de bases de données. Au niveau des serveurs de bases de données, une solution de clustering implique que :

- Les performances puissent évoluer en démultipliant les instances de bases de données ;
- Le tiers base de données soit évolutif, tolérant aux pannes (haute disponibilité et failover) ;
- Il n'y ait pas d'impact sur la partie cliente ;
- L'intégrité transactionnelle soit garantie.

Au niveau des serveurs de bases de données, la mise en oeuvre d'une architecture clusterisée s'opère généralement à deux niveaux : d'une part une infrastructure matérielle dédiée (réseaux locaux ultras rapides, solutions SAN) et d'autre part une solution logicielle dédiée (généralement une version Cluster d'un grand SGBDR du marché). Ces deux composants (matériels et logiciels) sont extrêmement coûteux.

Dans l'univers des SGBDR Open Source, il n'existe pas encore d'implémentation complète, capable de répondre aux exigences du clustering. En effet, MySQL propose un système de réplication des bases de données en mémoire reposant sur un module appelé NDB, mais celui-ci ne propose que du mirroring de données. De son côté, PostgreSQL propose une implémentation encore trop récente pour être évaluée.

C-JDBC se pose en alternative Open Source pour la mise en place de cluster de bases de données.

C-JDBC est un projet du consortium ObjectWeb, il a été développé à l'initiative de l'INRIA. Il se veut être aux bases de données, ce que le RAID est pour les disques.

L'objectif de C-JDBC est de fournir de façon transparente des mécanismes de clustering de

bases de données (réplication, partitionning, failover, etc.) à toutes les applications Java au travers de JDBC (Java Database Connection API). Il présente l'avantage de fonctionner avec n'importe quelle application Java (standalone, Web, etc.) sans qu'il soit nécessaire d'y apporter des modifications du point de vue du code, et, plus intéressant encore, avec n'importe quel serveur de bases de données. C-JDBC a été testé et fonctionne aujourd'hui avec :

- Tomcat, JBoss, JOnAS, IBM Websphere, BEA Weblogic pour les serveurs d'applications Java/J2EE,
- MySQL, PostgreSQL, Oracle, MS SQL Server, etc. pour les serveurs de bases de données.

C-JDBC est composé de deux éléments principaux :

- Un driver JDBC 2.0, appelé plus communément driver C-JDBC, qui est utilisé au niveau du serveur d'applications Java, ou d'une application Java standard,
- Un contrôleur C-JDBC reposant sur un composant qui se nomme RAIDb et qui a pour fonction d'assurer la scalabilité des performances, la haute disponibilité, la reprise sur panne, mais aussi des fonctions de gestion de cache, de logs de suivi, et de supervision.

RAIDb

C-JDBC repose sur un socle appelé RAIDb (Redundant Array of Inexpensive Databases).

RAIDb permet de combiner plusieurs instances de bases de données sous la forme d'un tableau logique de bases. Dès lors, le contrôleur RAIDb présente à l'application cliente la vue d'une unique base de données (virtuelle) et peut assurer ensuite la répartition de charge sur les différents serveurs qui composent le tableau logique.

La communication entre l'application cliente et la, ou les bases de données finales prend donc la forme d'une interaction au travers d'un middleware RAIDb. Le contrôleur RAIDb s'exé-

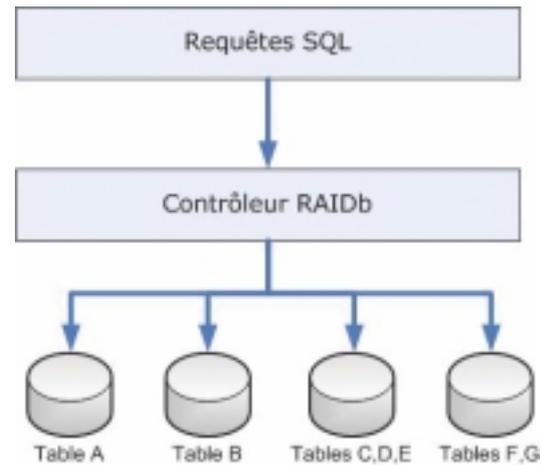


Fig 1 : Architecture C-JDBC en mode Raidb-o

cute sur un ou plusieurs serveurs dédiés. Il communique dans un premier temps, avec les applications, au travers du driver C-JDBC et assure, dans un second temps, la communication avec les serveurs de base de données qui composent le cluster.

RAIDb est une couche d'abstraction entre l'application et les bases de données qui est chargée d'assurer :

- la persistance des données,
- la scalabilité des performances,
- la haute disponibilité,
- le failover,
- la gestion du cache,
- des fonctions de log et de supervision.

A l'instar des solutions RAID pour les disques, il existe plusieurs modes de fonctionnement pour RAIDb :

- **RAIDb-o** propose uniquement le partitionnement des données. Dans cette configuration, C-JDBC se contente uniquement de partitionner les données, mais n'assure pas la duplication, ni la tolérance aux pannes. Cette configuration se compose au minimum de deux nœuds dans le cluster.

RAIDb-o permet, par exemple, de mettre en place un système de bases de données réparties, ou d'établir des liens entre différentes bases, sans que cela ait d'impact sur la configuration des serveurs d'applications, de bases de données ou des applications elles-mêmes. (Fig 1)

- **RAIDb-1** assure la réplication complète des données (mirroring), et a pour effet de garantir une meilleure tolérance aux pannes. Dès lors, il favorise la scalabilité de l'architecture, ce qui permet d'augmenter la dispo-

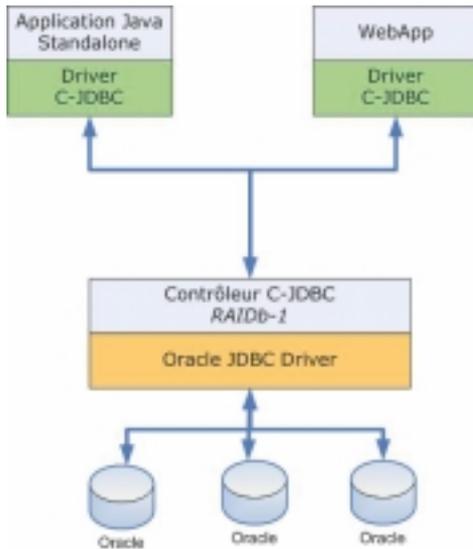


Fig 2 : Architecture C-JDBC en mode Raidb-1

nabilité des serveurs de données. Le contrôleur RAIDb assure la propagation des données qui doivent être ajoutées, modifiées, ou supprimées sur l'ensemble des instances de bases de données qui composent le cluster.

Cela permet ensuite à plusieurs applications d'accéder et d'effectuer des traitements qui peuvent être lourds sur ces données ; ces traitements se font directement sur des instances dédiées (cubes décisionnels, reporting, données de production, etc.) sans nécessairement nuire aux performances des applications qui pourront continuer à utiliser les autres instances. RAIDb-1 peut être mis en place à partir de deux noeuds dans le cluster. (Fig 2)

- **RAIDb-1ec** est similaire à RAIDb-1, si ce n'est qu'il propose en plus, des fonctions de contrôle d'erreurs reposant sur le principe suivant :

- Les requêtes qui doivent être envoyées aux différentes bases de données du cluster sont lues puis exécutées ;
- Les réponses aux requêtes envoyées sur chacun des serveurs de bases de données sont comparées par le contrôleur RAIDb ;
- Le contrôleur retourne un résultat, uniquement si une majorité de serveurs a apporté la même réponse.

Le cluster doit comprendre au moins 3 noeuds pour pouvoir utiliser ce mode.

- **RAIDb-2** permet de faire de la réplication partielle de données. Il autorise ainsi de

meilleures performances en répartissant la charge entre plusieurs instances qui peuvent être spécialisées par type de traitements.

Une configuration RAIDb-2 doit comporter au moins deux copies de chacune des tables à répliquer, et au moins trois noeuds. (Fig 3)

Il est aussi possible de monter des configurations composées d'associations entre les différents modes RAIDb présentés ci-dessus.

Au coeur du contrôleur C-JDBC

(Fig 4). N'importe quelle application Java peut utiliser les services de clustering de C-JDBC. Il suffit pour cela qu'elle utilise le driver JDBC fourni avec C-JDBC, en lieu et place du driver natif du SGBDR. (Fig 5)

D'un point de vue interne C-JDBC repose sur une notion de base de données virtuelle. Il s'agit d'une couche d'abstraction qui apporte ses mécanismes de "recovery log", de scheduling, de cache de requêtes ou de répartition de charge au dessus du serveur de bases de données cible.

Ces mécanismes sont implémentés au sein de différents modules de C-JDBC parmi lesquels :

- L'Authentification Manager assure la liaison entre le couple utilisateur/mot de passe utilisé par l'application et les couples utiliza-

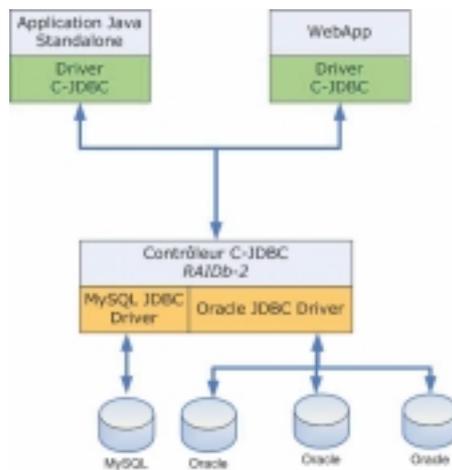


Fig 3 : Architecture C-JDBC en mode Raidb-2

teur/mot de passe utilisés par les bases de données. Il gère aussi la connexion et le login de l'administrateur qui assure la gestion des bases de données virtuelles C-JDBC,

- Le Scheduler est chargé d'effectuer les contrôles de concurrence. Il en existe différentes implémentations en fonction du mode d'utilisation de C-JDBC (une seule et unique

base de données, RAIDb-0, RAIDb-1 ou RAIDb-2). Le scheduler se positionne au niveau des requêtes et fonctionne selon une logique optimiste ou pessimiste (réordonnement des requêtes pour améliorer les temps de réponse, préférence pour les mises à jour massives, etc.) à partir des schémas récupérés automatiquement depuis les serveurs de base de données avec lesquels il collabore.

- Le Request Cache assure un cache sur le résultat des requêtes SQL transmises par le biais du contrôleur C-JDBC. Il permet de limiter les invalidations de cache (au niveau des tables, des colonnes, etc.) en analysant les requêtes SQL qui sont exécutées.

- Le LoadBalancer implémente la couche RAIDb.

- En mode RAIDb-0, les requêtes sont passées directement au serveur qui contient les tables nécessaires.

- En mode RAIDb-1, la lecture est effectuée par le thread courant. Les écritures sont effectuées en parallèle, par autant de threads qu'il existe de serveurs dédiés concernés dans le cluster. Le résultat est retourné, si la majorité ou la totalité des serveurs ont acquitté la transaction. Si l'un des noeuds du cluster était amené à faillir, mais que les autres aient répondu sans problème, le noeud défaillant serait désactivé automatiquement.

- En mode RAIDb-2, le comportement est identique au mode RAIDb-1, à l'exception du fait que l'instruction d'écriture est envoyée uniquement aux noeuds qui possèdent la, ou les tables demandées.

En complément, la répartition de charge en tant que telle peut aussi s'effectuer selon les algorithmes Round-Robin ou Weight Round-Robin, ou bien sur le serveur offrant la meilleure disponibilité (Least Pending Requests First). Ceci est intéressant si, et seulement si, tous les serveurs du cluster sont homogènes en terme de configuration et de charge.

- Le ConnectionManager assure le pooling des connexions au niveau des serveurs de bases de données. Il propose quatre modes de gestion des connexions :

- Simple, c'est-à-dire qu'aucun pooling n'est effectué. Une nouvelle connexion est obtenue pour chacune des requêtes.
- Random Wait, qui reste en attente de

connexions libres, à retourner si jamais toutes les connexions du pool devaient être utilisées.

- FailFast qui retourne une erreur si jamais le pool ne possède plus de connexions disponibles,
- VariablePool, dont la taille se redimensionne automatiquement si le nombre de connexions maximum est déjà atteint.
- Recovery Log chargé de la gestion des points de contrôle permettant de restaurer dynamiquement les données, lorsqu'une avarie s'est produite sur les serveurs de données, sur les instances du cluster qui ont été touchées, ou sur tout nouveau serveur ajouté au cluster. Recovery Log rejoue tout simplement l'ensemble des ordres SQL effectués depuis un checkpoint donné.

Scalabilité avec C-JDBC

C-JDBC permet d'assurer une architecture scalable, tant horizontalement que verticalement. La scalabilité horizontale qui peut être mise en place permet de redonder le contrôleur, pour que ce dernier ne constitue pas un Single Point Of Failure (point unique de défaillance) dans l'architecture, de distribuer la charge parmi les différents serveurs, et d'assurer la cohérence entre les différents groupes en place.

Les mécanismes de scalabilité verticale permettent d'entremêler différents niveaux de configuration RAIDb, et sont nécessaires lorsqu'un grand nombre de serveurs de bases de données sont mis en place dans l'architecture C-JDBC.

Conclusion

C-JDBC se pose comme une alternative originale dans le domaine du clustering de bases de données. En effet, l'architecture et la relative facilité de mise en œuvre dont il fait preuve, permettent d'implémenter un cluster de bases de données à partir de n'importe quel SGBD. A contrario des solutions classiques du marché, au-delà des aspects financiers que la constitution d'un cluster peut représenter, C-JDBC se veut surtout simple à mettre en œuvre au sein d'une architecture. Pour faire face, par exemple, à un besoin ponctuel de tenue à la charge, ou sur des projets pour lesquels le besoin n'a pas été identifié au préalable.

C-JDBC offre au niveau des serveurs de bases de données, de meilleures performances et

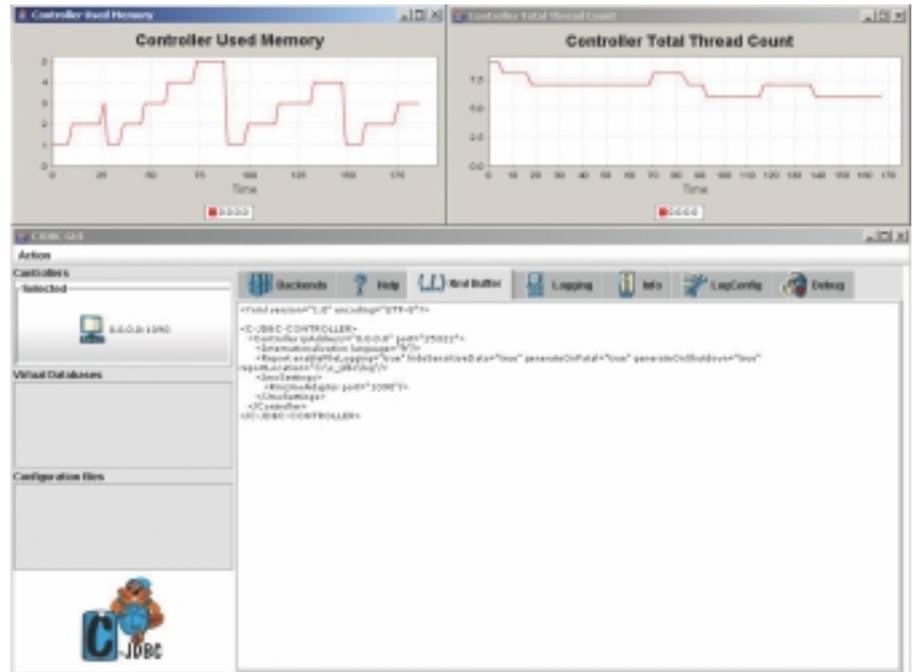


Fig 4 : Interface de configuration et d'administration de C-JDBC.

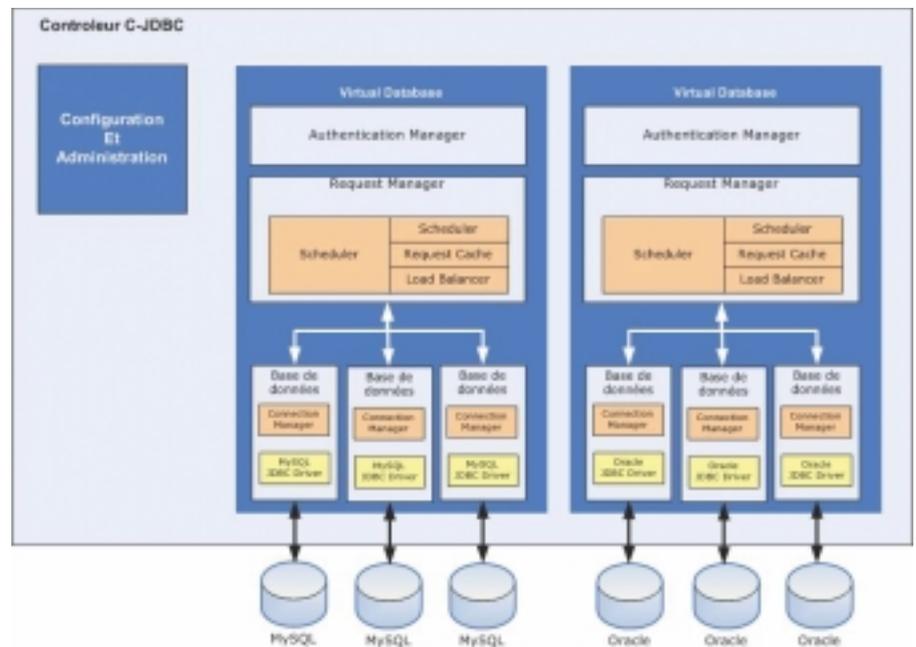


Figure 5 : Architecture interne du contrôleur C-JDBC.

une plus grande tolérance à la panne qu'une base de données unique, et ce, pour un coût réduit et sans modification de code au niveau des applications clientes, puisque C-JDBC repose sur le driver JDBC livré avec la base de donnée utilisée, et sur le moteur de cette dernière dans sa version la plus standard.

On pourra simplement regretter le fait que

cette solution alternative soit aujourd'hui confinée au monde Java. A quand un driver OLEDB pour C-JDBC ?

■ Mickael MAINDRON

Architecte chez NEOXIA, cabinet de conseil en architecture des systèmes d'information.

mickael.maindron@neoxia.com



Optimisez vos codes sources C/C++ avec le dernier-né des compilateurs d'Intel



Vous pouvez essayer gratuitement la dernière version 8 du compilateur d'Intel sous Linux. Il s'agit en quelque sorte d'une vitrine pour Intel, et le seul outil opérationnel si vous désirez tester les dernières technologies du fondeur.

Sous Windows, si le programmeur C++ décide de ne pas investir lourdement dans un compilateur (en terme de monnaie sonnante et trébuchante), il sera quand même gâté en matière de choix. En effet, tout d'abord un des meilleurs compilateurs qui soit, et disponible gratuitement, est celui de Microsoft. Celui-ci est livré par l'intermédiaire d'un Toolkit. Il s'agit en fait, à l'identique, du compilateur de Visual C++ : le compilateur est analogue à la version payante, mais évidemment vous ne disposez pas de l'environnement Visual Studio . Net. La dernière version 1.01 téléchargeable date de juillet 2004.

C builder v5.5 de Borland est gratuit et conforme aux derniers standards ANSI/ISO C/C++. De plus, il tourne sous Windows 9x, ce qui est loin d'être négligeable. Soulignons aussi qu'un code source compilé sous Windows, avec la version 6 de ce Borland C builder est portable sous Linux avec Kylix.

Nous avons aussi le compilateur de Jacob Navia, lcc-win32, gratuit pour un usage non commercial, et mingw32 le GCC, qui est spécifique à l'environnement Windows et également complètement gratuit.

Sous Linux, il y a Kylix précédemment cité, GCC bien entendu, qui est la suite de compilateurs de référence, et enfin le compilateur Intel C++ dont nous allons parler.

Ce qu'il y a sous le capot

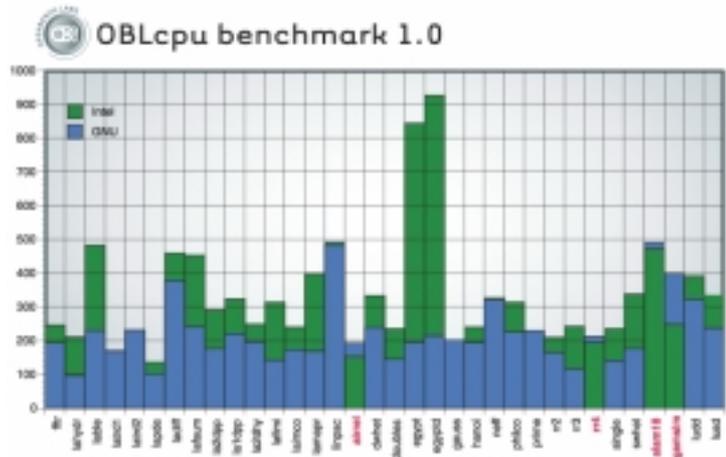
Le compilateur d'Intel C++ sous Windows est un outil commercial. Il s'intègre dans les environnements Visual studio . NET. Mieux : il est compatible avec le compilateur Visual C++ . NET, et vous pouvez recompiler des projets Microsoft Visual Studio .NET. De plus, l'Intel C++ Compiler Windows est fourni avec l'Intel C++ Compiler for eMbedded Visual C++,

```

Session Edition Affichage Signets Configuration Aide
[Icons]
Début des calculs : Wed Sep 29 15:09:33 2004
Fin des calculs: Wed Sep 29 15:09:37 2004
différence : 4.000000 secondes
Début des calculs : Wed Sep 29 15:09:39 2004
Fin des calculs: Wed Sep 29 15:09:41 2004
différence : 2.000000 secondes
[root@localhost ~]# clear ; ./single ; sleep 2 ; ./single

```

(*) ICC ne supporte pas, par exemple, les fonctions imbriquées ; un document rédigé par Intel et intitulé "Intel Compilers for Linux, Compatibility with GNU Compilers" reprend l'ensemble de ces incompatibilités qui, avec cette version 8, sont finalement fort peu nombreuses.



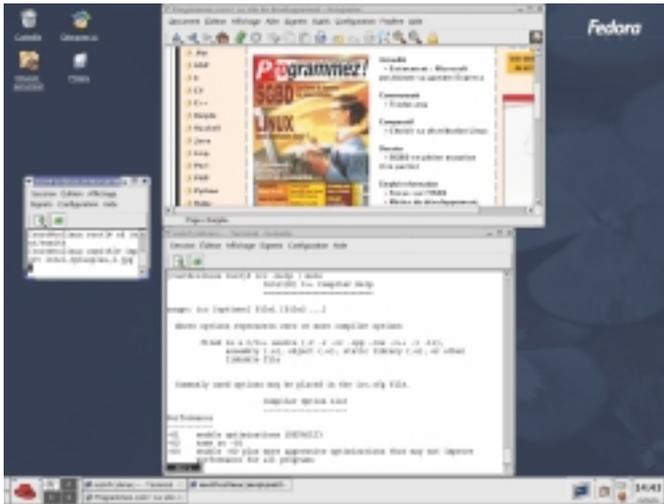
qui vous permettra de développer des applications destinées à des assistants personnels ou des téléphones mobiles.

Le compilateur d'Intel est téléchargeable gratuitement sous Linux, à condition de ne pas l'employer dans un but commercial. Sous Linux, la dernière version téléchargeable d'ICC (Intel C++ Compiler) est la 8.0 qui se décline en deux versions : une version 32 bits et une version 64 bits. Comme GCC, Intel a choisi d'adhérer aux standards, tels qu'ABI et C99. Sa compatibilité avec GCC est grande (*) mais non complète : il est possible de compiler un noyau Linux, mais en lui appliquant préalablement un patch.

Au mois d'août 2004, Red Flag, un éditeur Linux qui se destine au marché chinois, avait déjà annoncé que sa distribution avait été entièrement compilée avec ICC. Quelques ingénieurs Intel, en collaboration avec les développeurs du kernel, se sont en effet attelés à pouvoir réussir le test ultime de compilation du noyau avec ICC. C'est ainsi que vous pouvez trouver sur Internet un patch récent se rapportant à la compilation des noyaux 2.6.5 et 2.6.6 avec ICC. Certaines routines seront, de ce fait, accélérées jusqu'à 40 %, mais globalement, le gain en performance ne dépassera pas les 10 %.

Nous en arrivons au nœud gordien : les performances. D'après le vice-président de MySQL AB, le compilateur Intel C++ 8.0 offre un gain d'environ 20% sur l'exploitation du serveur MySQL. En réalité, gcc est un formidable compilateur, mais n'offre pas certaines optimisations, telles que le support d'OpenMP 2.0 et du jeu d'instructions HyperThreading spécifiques aux processeurs Intel.

D'autres caractéristiques permettent d'optimiser au maximum les applications : il s'agit du Profile Guided Optimization (PGO), de l'Interprocedural Optimization (IPO), du chargement de données en pile (prefetching) et enfin de l'optimisation floating-point.



Les optimisations

Le compilateur Intel C++ 8.0 emploie la totalité de la zone de mémoire floating-point pour une exécution plus efficace des instructions de type flottant. Les gains de performance sont dus aux instructions dites "d'overlapping", qui stockent leurs résultats de calcul dans n'importe quelle zone de mémoire.

L'IPO permet d'améliorer fortement la performance dans des programmes qui contiennent beaucoup de fonctions de petites ou moyennes tailles fréquemment utilisées, spécialement pour les programmes qui contiennent des appels dans des itérations.

Le PGO est un processus de compilation, qui permet au compilateur Intel C++ de tirer avantage de la microarchitecture du processeur, d'utiliser plus efficacement les instructions de pagination et la mémoire cache, et ainsi de réaliser de meilleures prédictions. Il améliore la performance de l'application en réduisant la suppression des instructions cache, en réorganisant la disposition du code, diminuant la taille du code et réduisant les mauvaises prédictions de branches.

Le prefetching de données est une technique qui vise à minimiser les temps d'attente d'accès à la mémoire. Le prefetching de données consiste à baliser d'instructions prefetch des références de données pré-sélectionnées, afin qu'elles soient déplacées en mémoire cache, avant même leurs utilisations.

Enfin, le compilateur ICC supporte le standard OpenMP 2.0 pour C/C++. Ce qui signifie la prise en charge de fonctionnalités d'autoparallélisation, selon le modèle de programmation SMP (symmetric multi-processing), à condition que le processeur (Intel) le supporte. Le compilateur détectera une itérative capable d'être exécutée en parallèle, et génère automatiquement du code multi tâches pour celle-ci !

Installation et configuration

Si vous êtes sous gentoo, vous pouvez essayer un "emerge icc" pour compiler icc, mais il vous faudra quand même vous inscrire sur le site d'Intel, afin de rapatrier le fichier de licence pour un usage non commercial. Attention : l'archive, téléchargeable après inscription sur le site d'Intel, renferme des RPMs. Et même s'il est théoriquement possible d'employer ce format universel avec la plupart des distributions (même avec gentoo), une distribution à base de Red Hat est préférable. Lorsque vous exécutez le script d'installation, le shell se plaindra de ne pas trouver ce fameux fichier de licence (qu'Intel vous expédie par courrier électronique).

```
#!/install.sh
```

```
A valid FLEXlm license is required to install this product, but none were found
in $INTEL_LICENSE_FILE (./opt/intel/licenses:/opt/intel_cc_80/licenses).
Where is a valid FLEXlm license for this product?
Enter directory or file, or just 'Enter' to exit.
```

Vous devez par conséquent recopier cette licence sous l'arborescence demandée :

```
#ls /opt/intel/licenses
noncommercial_cpp_l_51182202.lic
```

Le script le repère, déclare ne détecter aucun compilateur Intel, puis vous demande ce que vous désirez installer :

```
RPM shows no Intel packages as installed.
```

```
Which of the following would you like to install?
```

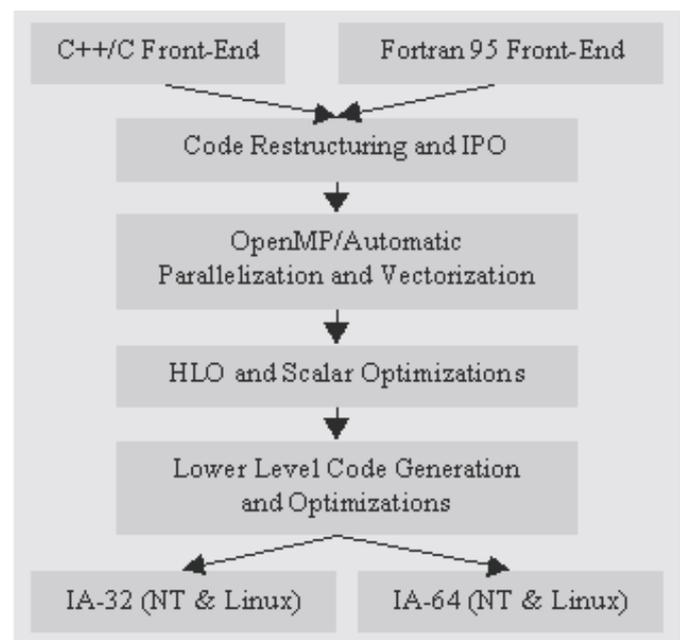
1. Intel(R) C++ Compiler for 32-bit applications, Version 8.0
2. Linux Application Debugger for 32-bit applications, Version 7.3.1
- x. Exit

Ensuite vous devez approuver la licence, et indiquer un chemin d'installation :

```
Where do you want to install to? Specify directory starting with '/'.
[/opt/intel_cc_80]
What rpm install options would you like? [-U -replacefiles]
```

```
Intel(R) C++ Compiler for 32-bit applications, Version 8.0
Installing...
Installation successful.
```

Vous devez maintenant installer l'environnement du compilateur. D'abord en ajoutant le chemin aux exécutables :



```
#PATH=/opt/intel_cc_80/bin:$PATH
```

Puis en exportant les variables nécessaires :

```
#export CC=icc
#export CXX=icpc
#export LD=ld
#export LD_LIBRARY_PATH=/opt/intel_cc_80/lib:$LD_LIBRARY_PATH
```

Tests et compilations

Globalement, certains outils se recompilent sans aucun changement, comme `xmms-1.27-r19`,

`bzip2-1.0.2-r2`, `libogg-1.0` ou `openquicktime-1.0-r1`. D'autres ne passent pas tels quels, comme `jpeg-mmx-1.1.2-r1` ou `xvid-0.9.1`.

Passons maintenant à la pratique. Éditez le code source suivant (`sinicc.c`). Il s'agit d'une routine de calculs de sinus.

sinicc.c

```
#include <math.h>
#include <stdio.h>
#include <time.h>

int calculsine(double *a, double *b, int N) {
    int i;
    for (i=0;i<N;i++){
        b[i]=sin(a[i]);
    }
}

int calculsine(double *a, double *b, int N);

int main() {

    const int N = 120000;
    double a[N], b[N];
    int i;

    time_t t1,t2;

    t1 = time(NULL);

    printf("\nDébut des calculs : %s", ctime(&t1));

    for (i=0;i<N;i++)
        a[i] = i;

    for (i=0;i<300;i++)
        calculsine(a,b,N);

    t2 = time(NULL);

    printf("\nFin des calculs: %s", ctime(&t2));
    printf("différence : %lf secondes\n", difftime(t2, t1) );
}
```

Avec `gcc`, nous devons effectuer une édition des liens avec la bibliothèque mathématique (`-lm`), ce qui donne :

```
#gcc sinicc.c -o singcc -lm -O3
```

```
#!/singcc
```

```
Début des calculs : Tue Sep 14 08:52:10 2004
```

```
Fin des calculs: Tue Sep 14 08:52:14 2004
```

```
Différence : 4.000000 secondes
```

La précision du temps écoulé n'est pas très grande, mais peu importe, elle est suffisante pour détecter un gain avec la bibliothèque mathématique d'Intel (`-O3` indique un haut niveau d'optimisation).

Modifions notre code source pour pouvoir le compiler avec ICC :

```
#ifndef _INTEL_COMPILER
#include <mathimf.h>
#else
#include <math.h>
#endif
```

Ce qui permet d'inclure `mathimf.h` lorsque le compilateur `icc` est sollicité :

```
#icc sinicc.c -o sinicc -axW -O3
```

```
...
```

```
sinicc.c(32) : (col. 2) remark: LOOP WAS VECTORIZED.
```

```
sinicc.c(20) : (col. 12) remark: main has been targeted for automatic cpu
dispatch.
```

```
...
```

Remarquons au passage que le compilateur `icc` a détecté une optimisation possible et a généré du code en conséquence. En effet, les options `-axW` (*) mettent en place la vectorisation automatique pour un processeur Pentium 4 (SSE2). Pour un Pentium 3, ce sera `-axK` (SSE).

Si vous voulez que le compilateur vous donne des détails, utilisez les options `-v` (verbose) et `-opt_report` (affiche le détail des optimisations) :

```
...
```

```
6 FUNCTIONS HAD VALID STATIC PROFILES
```

```
IPO CURRENT QUALITY METRIC: 50.0%
```

```
IPO POSSIBLE QUALITY METRIC: 50.0%
```

```
IPO QUALITY METRIC RATIO: 100.0%
```

Exécutons :

```
#!/sinicc
```

```
Début des calculs : Tue Sep 14 08:58:19 2004
```

```
Fin des calculs: Tue Sep 14 08:58:21 2004
```

```
Différence : 2.000000 secondes
```

Nous pouvons séparer le code source de la fonction et de "main", puis générer des codes objets avec `icc` et `gcc` que nous lierons ensemble, afin de créer un exécutable. ICC utilise le standard ABI (Application Binary Interface) pour décrire le format des fichiers objets générés. GCC étant compatible ABI depuis la version 3.2, un fichier objet généré par un des compilateurs peut faire l'objet d'une édition des liens avec un autre.

```
#gcc -version
```

```
gcc (GCC) 3.3.2 20031022 (Red Hat Linux 3.3.2-1)
Copyright (C) 2003 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
```

```
#gcc -c main.c
```

```
(ce qui donne main.o)
```

```
#icc -c fonction.c -axW -O3
```

```
(ce qui donne fonction.o)
```

```
#gcc main.o fonction.o -o executable -L/opt/intel_cc_80/lib -lsvml -limf
ce qui donne "executable" :
```

```
#!/executable
```

```
Début des calculs : Tue Sep 14 09:03:43 2004
```

```
Fin des calculs: Tue Sep 14 09:03:45 2004
```

```
différences : 2.000000 secondes
```

Maintenant prenez garde au fait que, par défaut, certaines bibliothèques spécifiques à Intel sont liées dynamiquement (l'option `-i_dynamic`). Pour les lier statiquement, et ainsi déployer le programme sur des machines où le compilateur `icc` n'a pas été installé, vous devez utiliser l'option `-static`. Pour vous donner une idée, l'exécutable `sinicc` généré avec l'option `-static` occupe plus de 400 Ko, tandis que l'exécutable `singlecc` (dynamiquement lié) occupe moins de 6 Ko...

Conclusion

C'est une évidence, mais le compilateur `icc` ne s'avère vraiment intéressant qu'à la condition de développer sur base de processeurs Intel, et plus encore, à partir d'une nouvelle architecture du style de celle du Pentium 4. Vous pourrez ainsi bénéficier d'optimisations telles que

Un benchmark ?

SCIMARK2 est un benchmark qui se compile en un tour de main, et pour le compiler avec `icc` il suffit d'éditer le `Makefile` et d'y indiquer "CC=icc" à la place de "CC=cc".

```
#make
```

```
...
```

```
icc -O -o scimark2 scimark2.o FFT.o kernel.o Stopwatch.o Random.o
SOR.o SparseCompRow.o array.o MonteCarlo.o LU.o -lm
```

```
#!/scimark2
```

```
Using 2.00 seconds min time per kenel.
```

Composite Score:	386.90	
FFT	Mflops: 299.70	(N=1024)
SOR	Mflops: 467.71	(100 x 100)
MonteCarlo:	Mflops: 63.01	
Sparse matmult	Mflops: 347.67	(N=1000, nz=5000)
LU	Mflops: 756.42	(M=100, N=100)

Ne tombez pas dans le piège du benchmark : `gcc` n'est pas globalement plus lent de x% que le compilateur d'Intel, car il s'avère parfois plus lent pour un test, tandis qu'il est plus rapide dans d'autres.

ICC

Points positifs :

- Meilleures performances en exploitant des optimisations spécifiques aux processeurs Intel (en matière de benchmarks, il faut rester extrêmement prudent);
- Bonne compatibilité avec GCC (compilation du kernel possible, si un patch est appliqué);
- Support 64 bits (pour Intel Itanium seulement).

Points négatifs :

- Le package est livré sous la forme de RPM (compatible avec la famille Linux Red Hat);
- Le débogueur `idb` (Intel Application Debugger) est uniquement livré en ligne de commande sans interface graphique;
- Licence propriétaire.

GCC

Points positifs :

- GCC est libre;
- GCC (GNU Compiler Collection) représente une suite de compilateurs (C, C++, Ada, fortran77, Java);
- GCC est disponible sur de très nombreuses plates-formes;
- GCC peut cross-compiler;
- GCC peut recompiler entièrement une distribution GNU/Linux, sur toutes les plates-formes supportées, sans patch;
- Les développeurs GCC sont très réactifs.

Point négatif :

- Les dernières technologies d'optimisation d'Intel ne sont pas toutes incluses dans `gcc`.

l'OpenMP ou la vectorisation automatique, que GCC n'offre pas encore. Les développeurs de GCC ont démontré par le passé, à quelle vitesse ils pouvaient implémenter des options de ce type, et nous ne serions pas étonnés d'apprendre une prochaine annonce dans ce sens.

Mais attention : si vous compilez votre code pour un Pentium 4 (l'option `-tpp7 (*)`), celui-ci ne pourra pas s'exécuter sur un Pentium 2 ou 3, et encore moins sur un AMD.

N'oublions pas non plus qu'ICC est distribué selon les termes d'une licence propriétaire. En pratique, dans les cas où `icc` est meilleur (...), le gain de vitesse ne saurait compenser les efforts qu'il faudrait déployer pour changer de compilateur suivant la plate-forme cible...

Liste des liens :

Le compilateur de Borland C builder v5.5:

http://www.borland.com/products/downloads/download_cbuilder.html

Le compilateur de Jacob Navia lcc-win32 :

<http://www.cs.virginia.edu/~lcc-win32/>

Mingw32 : <http://www.mingw.org/>

Le compilateur gratuit de Microsoft :

<http://msdn.microsoft.com/visualc/vctoolkit2003/>

Compilation de Red Flag avec ICC :

<http://www.pcmag.co.uk/news/1157196>

Patch kernel pour ICC :

<http://www.pyrillion.org/downloads/linux-2.6-icc-1.0.tar.bz2>

Téléchargement d'ICC pour un usage non commercial :

<http://www.intel.com/software/products/noncom/>

Standard ABI : <http://www.codesourcery.com/cxx-abi>

Le bench SCIMARK2 : <http://math.nist.gov/scimark2/scimark2c.zip>

■ Xavier Leclercq - Xavier.Leclercq@programmez.com

(*)

- axW : Nouvelles instructions du Pentium 4 SSE2;
- axM : Intel MMX extensions;
- axK : Intel Streaming SIMD Extensions (SSE);
- axi : Intel Pentium Pro/Pentium II instructions.

La programmation du shell de Windows

SOURCES DE L'ARTICLE
WWW.PROGRAMMEZ.COM

Programmer les menus contextuels de Windows.

L'interface graphique de Windows, le Shell, est en grande partie extensible. Nous allons nous adonner à cette intéressante activité en nous appuyant sur nos connaissances relatives aux serveurs COM, acquises le mois dernier.

L'interface utilisateur de Windows, dite le Shell, est largement construite à partir d'une technologie phare de Microsoft : COM. Ainsi, le programmeur Windows peut voir le Shell Windows comme un agrégat de composants logiciels, ou serveurs COM. Il est parfois possible de piloter, côté client, les composants de Windows. Ainsi avons-nous fait avec le planificateur de tâches (c.f. programmez! 67). Mais plus souvent, c'est Windows lui-même qui est le client. Le mécanisme est simple. Pour bon nombre d'éléments fonctionnels du Shell, tels que menus contextuels ou autres 'Tool Bands', Windows, alias le Shell, examine la base de registre, avant d'afficher quoi que ce soit à l'écran à l'usage de l'utilisateur. Si lors de cet examen, des composants logiciels COM in-process (autrement dit des dll) sont trouvés, alors les méthodes qu'ils exposent seront invoquées. Ces méthodes doivent être l'implémentation d'interfaces prédéfinies par Microsoft. Ces interfaces prédéfinies sont autant de possibilités d'extension offertes au program-

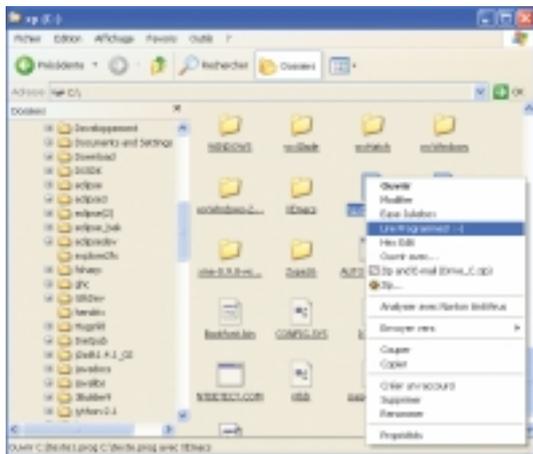
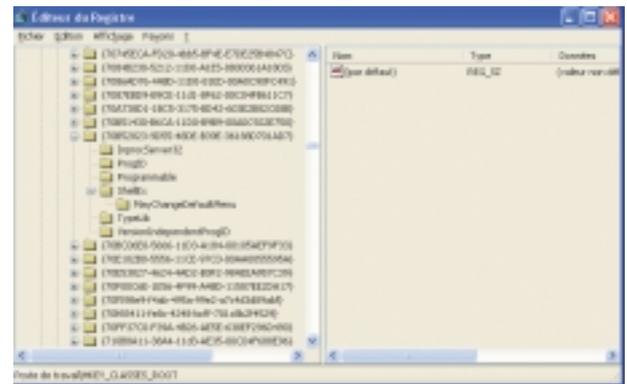


Fig 1 : Nous ajoutons une entrée au menu contextuel de Windows, au moyen d'un petit composant COM.

meur, qui peut ainsi créer des sortes de plug-ins, enrichissant le comportement natif de Windows. Nous allons voir ensemble comment écrire un composant COM qui ajoute une entrée au menu contextuel de Shell, quand le pointeur de la souris survole des fichiers dont l'extension est .prog. (Fig 1) Avant de commencer, considérons deux choses. D'abord, ce que nous nous proposons de faire est un exemple simple. Si simple, que nous pourrions atteindre notre but en ajoutant des clés dans le registre, sans écrire de composant COM. Mais si nous voulions, par exemple, que le menu contextuel soit adapté au contenu du fichier .prog concerné, alors le composant COM est le seul moyen. A vous de faire travailler votre imagination, et de créer des extensions d'enfer à partir de l'exemple rudimentaire donné dans cet article. Ensuite, deuxiè-

Fig 2 : Notre serveur COM est inscrit dans la base de registre.



me point, la description du mécanisme de 'plug-ins', que nous avons donnée plus haut, est loin de couvrir la totalité de Shell/COM. Ce n'est qu'un aspect des choses. Le Shell est beaucoup plus riche et complexe que cela, ce que nous découvrirons peut-être ensemble ultérieurement.

La base de registre

Nous savons déjà que nous devons enregistrer les serveurs COM dans la base de registre de Windows. Cet enregistrement est facilité par les outils de développement qui utilisent le programme regsvr32.exe de Microsoft (ou tregsvr.exe de Borland). L'opération consiste à écrire le CLSID du composant sous HKEY_CLASSES_ROOT\CLSID (Fig 2). Puis sont écrites des sous-clés, informant le système de la localisation de la dll et du modèle de threading. Toutes ces informations sont écrites automatiquement par la routine d'enregistrement de la dll, le rôle de regsvr32 consistant essentiellement à appeler la dite routine. En outre, dans le cas qui nous occupe, Microsoft recommande d'ajouter la sous-clé ShellExt\MayChangeDefaultMenu. Ensuite, il faut indiquer au système que nous avons créé, une extension, et que celle-ci doit s'activer pour les fichiers .prog. Pour Windows, des fichiers à l'extension commune font partie d'une classe. C'est donc tout naturellement que nous écrivons une clé sous la racine HKEY_CLASSES_ROOT. (Fig 3) Son nom sera :

```
ShellExt\ContextMenuHandlers\ProgrammezExt
```

Le terme ProgrammezExt est de notre choix. En revanche Les termes ShellExt et ContextMenuHandlers sont obligatoires. Enfin, la valeur de la clé sera notre CLSID de serveur. Ainsi, la classe de fichiers et l'extension du Shell sont mises en correspondance. Nous verrons plus loin comment réaliser par programmation cette écriture dans le registre, sous Visual C++ et sous C++Builder.

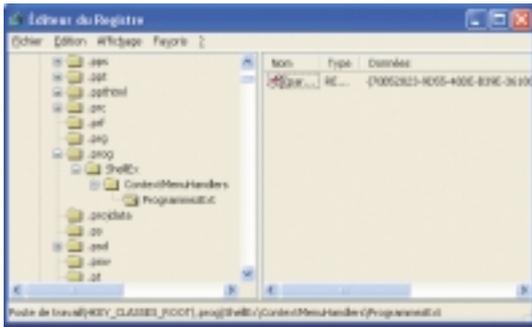


Fig 3 : Notre composant COM doit être associé avec une classe de fichiers.

Les interfaces

Celles-ci sont au nombre de deux : IShellExtInit et IContextMenu. La première ne contient qu'une méthode, qui est aussi la plus difficile à implémenter. Si cette méthode, Initialize, échoue ou est absente, le Shell ne plante pas et ne tient tout simplement pas compte de l'extension. Implémenter Initialize est ardu, car cela nous plonge (un petit peu) dans les arcanes du Clipboard et du Glisser-Déposer de Windows. En effet, le menu contextuel apparaît quand le pointeur de la souris a sélectionné un fichier, ou autrement dit, quand un glisser-déposer pourrait être lancé (et le glisser-déposer travaille implicitement avec le Clipboard). Parmi plusieurs paramètres que reçoit Initialize, le pointeur sur une interface IDataObject nous intéresse au plus haut point. Cette interface contient le, ou les noms de fichiers actuellement sélectionnés par la souris. Le tout est de savoir les récupérer... Ceci se fait en 5 temps.

1. Remplir une structure FORMATETC. Cette structure sert à définir à la méthode GetData de IDataObject sous quel format elle devra nous retourner les données. Remarquez que l'on demande un format DROP bien que nous ne fassions pas du tout de glisser-déposer.
2. Invoquer GetData de IDataObject. Ceci a pour effet de remplir une structure STGMEDIUM. Cette structure est une sorte de conteneur générique. À ce stade, les données ne sont pas encore utilisables.
3. Invoquer DragQueryFile une première fois pour connaître le nombre de fichiers contenus dans le STGMEDIUM.
4. Invoquer autant de fois que nécessaire DragQueryFile, afin d'obtenir les noms de fichiers un à un. Noms que nous stockons dans un vecteur à cette occasion.
5. Libérer la mémoire occupée par le conteneur STGMEDIUM.

On a déjà vu des mécanismes plus simples... Le lecteur voudra bien se reporter au code d'exemple sur le Cd-Rom accompagnant la revue (ou sur www.programmez.com) pour plus d'informations, ainsi qu'à la documentation officielle MSDN (<http://msdn.microsoft.com>). Nous préférons donner ici le schéma général qui ne saute pas forcément aux yeux dans la documentation Microsoft.

L'interface IContextMenu

Cette interface est plus simple à implémenter et elle concerne le menu surgissant proprement dit. Si l'on veut dessiner dans le menu, on implémentera plutôt IContextMenu2 ou IContextMenu3, ce qui ne présente pas de difficulté particulière.

Dans la première méthode 'QueryContextMenu' nous invoquons l'API système InsertMenu, pour insérer l'entrée de menu comme on le fait dans une application classique, à ceci près qu'en guise de quatrième paramètre nous donnons l'offset, pour positionner notre entrée de

menu. Notez bien l'usage de la macro MAKE_HRESULT pour construire la valeur de retour de la fonction.

Vient ensuite GetCommandString dans laquelle nous construisons une chaîne qui apparaîtra dans la barre de statut de l'explorateur, lorsque le pointeur de la souris survole notre entrée de menu. Le point difficile ici est que nous devons travailler soit en Ascii soit en Unicode, à la demande du système. Pour nous aider avec l'Unicode nous utilisons la macro à la sauce Microsoft USES_CONVERSION. Cette macro ATL doit apparaître au début du corps de la fonction. Moyennant quoi, nous pouvons employer la macro ATL A2W qui nous fabrique un pointeur sur une chaîne Unicode à partir d'un pointeur sur une chaîne Ascii. Notons encore que dans cette méthode nous filtrons les fichiers .prog. En effet bien que nous ayons fait le nécessaire avec le registre, des fichiers étrangers peuvent apparaître dans le DataObjet... C'est notamment le cas si on sélectionne d'abord un fichier .prog avec la souris, puis un ou plusieurs fichiers quelconques... Pour terminer, la méthode InvokeCommand nous permet de réagir à la sélection de notre menu par l'utilisateur. Ici nous ouvrons nos fichiers dans l'éditeur XEmacs, via l'API CreateProcess. Bien évidemment, vous devez adapter le code à votre convenance, surtout si XEmacs n'est pas présent sur votre système ;-)

Voyons maintenant comment procéder pratiquement avec Visual C++ 6.0 et C++Builder 6.0. Nous partons du principe que le lecteur a lu Programmez! 68 et nous ne revenons pas sur les maniements de base. Même si vous travaillez avec C++Builder, lisez le paragraphe relatif à Visual, cela vous sera utile.

Visual C++

Créez un projet ATL en veillant bien à demander la création d'une dll (Fig 4). Puis ajoutez un composant ATL sous le nom (par exemple) de ProgrammezExt et conservez les attributs par défaut proposés par Visual. (Fig 5) Compilez le projet, pour que tous les fichiers soient générés. A ce stade, nous avons une CoClass contenant une interface totalement vide et qui le restera. Cette interface pourrait même être supprimée. Seule la présence de la CoClass est requise, afin que le système puisse accéder aux interfaces IShellExtInit et IContextMenu que nous devons implémenter. La question est maintenant : comment faire pour implémenter une interface prédéfinie ? Nous ne travaillerons pas avec le fichier IDL cette fois. Tout ce que nous avons à considérer est que nous

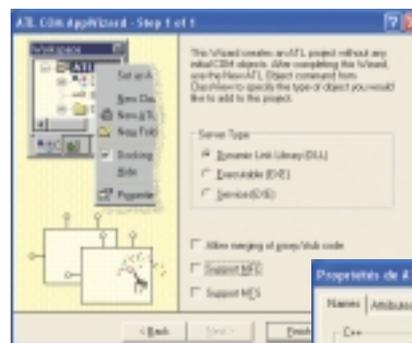


Fig 4 : Création du composant COM sous Visual C++. Etape 1.

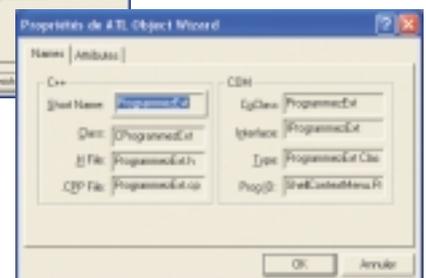


Fig 5 : Création du composant COM sous Visual C++. Etape 2.

devons écrire une classe, notre CoClass, et que celle-ci va dériver les interfaces. Avant tout, le compilateur doit connaître les définitions des interfaces. Pour cela, le fichier ProgrammezExt.h doit inclure des entêtes système

```
#include <Shlobj.h>
#include <comdef.h>
```

Regardons ensuite la déclaration de la CoClass telle qu'elle a été générée par le compilateur :

```
class ATL_NO_VTABLE CProgrammezExt :
    public CComObjectRootEx<CComSingleThreadModel>,
    public CComCoClass<CProgrammezExt, &CLSID_ProgrammezExt>,
    public IDispatchImpl<IProgrammezExt, &IID_IProgrammezExt, &
LIBID_SHELLCONTEXTMENUlib>
{
public:
    CProgrammezExt()
    {
    }

DECLARE_REGISTRY_RESOURCEID(IDR_PROGRAMMEZEXT)

DECLARE_PROTECT_FINAL_CONSTRUCT()

BEGIN_COM_MAP(CProgrammezExt)
    COM_INTERFACE_ENTRY(IProgrammezExt)
    COM_INTERFACE_ENTRY(IDispatch)
END_COM_MAP()

// IProgrammezExt
public:
};
```

Ce que nous devons faire est simple. D'abord, nous devons ajouter l'héritage public des interfaces qui nous intéressent. Au passage, si nous le souhaitons, nous pouvons retirer l'héritage de IDispatchImpl. Le début de la classe devient donc :

```
class ATL_NO_VTABLE CProgrammezExt :
    public CComObjectRootEx<CComSingleThreadModel>,
    public CComCoClass<CProgrammezExt, &CLSID_ProgrammezExt>,
    public IShellExtInit,
    public IContextMenu
{ // etc., etc.
```

Ensuite nous devons modifier la table de pointeurs des méthodes. Cette table est déclarée dans un sandwich de macros :

```
BEGIN_COM_MAP(CProgrammezExt)
    COM_INTERFACE_ENTRY(IProgrammezExt)
    COM_INTERFACE_ENTRY(IDispatch)
END_COM_MAP()
```

Les entrées de cette déclaration disparaissent et sont remplacées par les interfaces prédéfinies :

```
BEGIN_COM_MAP(CProgrammezExt)
    COM_INTERFACE_ENTRY(IShellExtInit)
    COM_INTERFACE_ENTRY(IContextMenu)
END_COM_MAP()
```

Il ne nous reste plus qu'à déclarer les méthodes des interfaces dans la classe. Les déclarations sont, bien sûr, issues de la documentation Microsoft :

```
// IContextMenu
STDMETHOD(GetCommandString)(UINT, UINT, UINT*, LPSTR, UINT);
STDMETHOD(InvokeCommand)(LPCMINVOKECOMMANDINFO);
STDMETHOD(QueryContextMenu)(HMENU, UINT, UINT, , UINT, UINT);

// IShellExtInit
STDMETHOD(Initialize)(LPCITEMIDLIST, LPDATAOBJECT, HKEY);
```

Et bien entendu, le corps de ces méthodes sera donné dans le fichier ProgrammezExt.cpp. À ce stade, nous pouvons compiler. Le composant s'enregistre, mais le menu n'apparaît toujours pas, car nous n'avons pas encore mis le composant COM en correspondance avec la classe .prog. Pour cela, modifiez le fichier ProgrammezExt.rgs, comme indiqué (Encadré 1). Ceci fait, recompilez (Build), ça fonctionne. Non ? Alors peut-être devrez-vous ouvrir une fenêtre de l'explorateur Windows, afin que la base de registre soit relue. De même, si vous modifiez le code après que cela ait fonctionné, l'éditeur de liens émettra peut-être une erreur, car le fichier dll est verrouillé par le système et ne peut être réécrit. Vous pouvez contourner le problème en fermant les fenêtres de l'explorateur Windows pointant sur des fichiers .prog. Mais le mieux est aussi de désinscrire le composant avec regsvr32, avant de recompiler.

C++Builder

Créez une librairie ActiveX (Fig 6) sous le nom (par exemple) ShellContextMenu. Retournez dans le Wizard et depuis l'onglet ActiveX, sélectionnez un 'Com Object'. (Fig 7) Baptisez le :

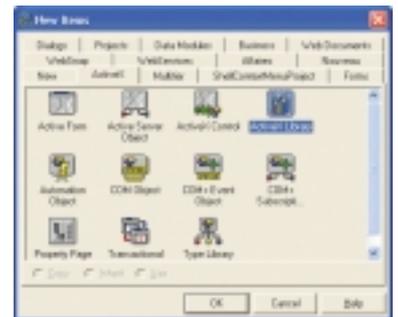


Fig 6 : Sous C++Builder, nous développons notre composant dans une librairie ActiveX. Etape 1



Fig 7 : Ajout du composant COM dans la librairie ActiveX sous C++Builder. Etape 2

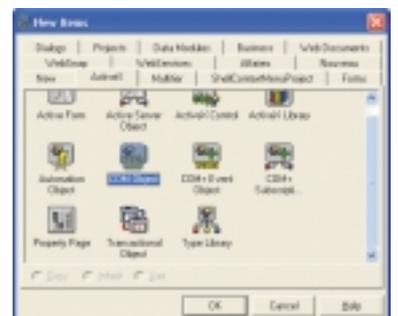


Fig 8 : Définition du nom de la CoClass sous C++Builder. Etape 3

ProgrammezExt. (Fig 8) Nous ne pouvons, hélas, nous servir du dialogue à ce stade, pour dire que nous souhaitons implémenter des interfaces Shell. Compilez le projet. Ajoutons maintenant nos modifications. Dans ProgrammezExtImpl.h, ajoutez l'héritage multiple, modifiez la table de pointeurs et donnez la déclaration des méthodes, comme expliqué pour Visual. N'oubliez pas les en-têtes système. Donnez le corps des méthodes dans ProgrammezExt.cpp. Il nous reste à nous occuper de l'enregistrement. Pour cela, revenez au fichier ProgrammezExtImpl.h et modifiez, dans la déclaration de la classe, la méthode UpdateRegistry, comme indiqué ci-contre (Encadré 2). Faisons plusieurs remarques. d'abord, nous devons inclure l'en-tête registry.hpp pour compiler le code. Par sécurité ce code crée des clefs, mais n'en supprime pas. Modifiez-le selon vos besoins. Le CLSID se trouve dans le fichier ShellContextMenu_TLB.h. Enfin, n'oubliez pas que C++Builder n'enregistre pas le composant à la compilation. Vous devez le faire manuellement depuis le menu 'Run\Register ActiveX Server'

Un problème ?

Il se peut que C++Builder émette une flopée d'erreurs abscones à la compilation. Par exemple :

```
[C++ Error] shlobj.h(1422): E2238 Multiple declaration for 'FVSHOWINFO'
```

Par défaut, C++Builder utilise des en-têtes.hpp pour la librairie VCL écrite en Pascal. Pour le cas qui nous occupe, le fichier shlobj.hpp est donc inclus automatiquement. Or, ce fichier redéclare des types qui sont en conflit avec ceux de shlobj.h. Nous ne pouvons omettre d'inclure ni ce dernier, ni le premier. Par contre, votre esprit affûté n'a pas manqué de remarquer que nous n'utilisons pas du tout la VCL dans notre projet. Nous pouvons donc demander au préprocesseur de C++Builder de n'inclure que les fichiers en-têtes Windows lorsqu'il traitera shlobj.hpp. Ceci se fait en ajoutant :

```
NO_WIN32_LEAN_AND_MEAN
```

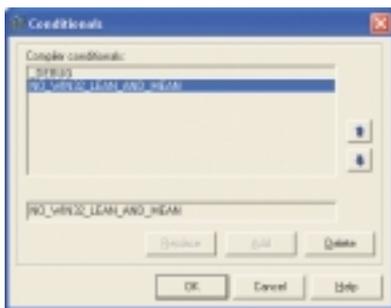


Fig 9 : En cas de problème de compilation sous C++Builder, définir une directive à l'attention du préprocesseur.

dans les directives conditionnelles du projet (Fig 9). À bientôt, pour de nouvelles aventures au coeur de Windows !

encadré 1

Extrait du fichier ProgrammezExt.rgs (Visual) pour l'inscription du composant dans le registre

```
NoRemove CLSID
{
    ForceRemove {70B52823-9D55-48DE-B39E-36188D701AD7} = s 'CProgrammezExt Class'
    {
        ProgID = s 'ShellContextMenu.CProgrammezExt.1'
```

```
VersionIndependentProgID = s 'ShellContextMenu.CProgrammezExt'
ForceRemove 'Programmable'
InprocServer32 = s '%MODULE%'
{
    val ThreadingModel = s 'Apartment'
}
NoRemove ShellEx
{
    NoRemove MayChangeDefaultMenu
}
'Typelib' = s '{A2C83B6F-E715-4153-92CC-EA6E569D530D}'
}

NoRemove .prog
{
    NoRemove ShellEx
    {
        NoRemove ContextMenuHandlers
        {
            ProgrammezExt = s '{70B52823-9D55-48DE-B39E-36188D701AD7}'
        }
    }
}
```

encadré 2

Routine pour enregistrer le composant COM sous C++ Builder.

```
static HRESULT WINAPI UpdateRegistry(BOOL bRegister)
{
    TRegistry *rg = new TRegistry();
    TTypedComServerRegistrarT<TProgrammezExtImpl>
    regObj(GetObjectCLSID(), GetProgID(), GetDescription());
    if(bRegister)
    {
        AnsiString cle;

        HRESULT hr = regObj.UpdateRegistry(bRegister);

        rg->RootKey = HKEY_CLASSES_ROOT;
        cle = "\\.\prog\ShellEx\ContextMenuHandlers\ProgrammezExt";
        rg->OpenKey(cle, true);
        rg->WriteString("", "{49AB3241-D983-430B-9D4EDA939A0D35C5}");
        rg->CloseKey();

        cle = "\\CLSID\{49AB3241-D983-430B-9D4E-DA939A0D35C5}\ShellEx\MayChangeDefaultMenu";
        rg->OpenKey(cle, true);
        rg->CloseKey();

        return hr;
    }
    return regObj.UpdateRegistry(bRegister);
}
```

■ Frédéric Mazué - fmazue@programmez.com

Sérialisation XML sous Mono 1.0.1 (Linux) et .NET (Windows), et convertisseur VB.NET vers C #.

La sérialisation est une technique qui consiste à sauvegarder un objet sur disque ou vers le réseau. Autrement dit, un objet possède un certain état en mémoire que l'on désire conserver pour le communiquer à un autre objet (d'un autre processus pouvant, par exemple, tourner sur une autre machine distante). L'objet passe du stade volatile (mémoire) à persistant (stockage sur disque), et on dira que la sérialisation "aplatit" l'objet (en le sauvegardant). Le phénomène inverse de désérialisation charge ce flux sériel de données, pour le transmettre à un autre objet (opération de chargement).

.NET propose différents moyens pour sérialiser/désérialiser un objet. Nous allons en voir trois : le format binaire, le format SOAP et le format XML.

Le format binaire

Imaginons qu'un développeur désire sauvegarder temporairement un objet : il peut utiliser une base de données locale (à condition que l'objet en question soit facilement représentable dans une structure de base de données), ou encore créer un fichier XML comme nous le verrons plus loin. Cependant, une autre possibilité s'offre à lui : la sauvegarde binaire en employant les attributs personnalisés [Serializable] et [NonSerializable]. Dans l'exemple suivant, si vous ne marquez pas la classe Nuancier comme étant d'attribut [Serializable], à l'exécution (à la compilation cela passe) vous obtiendrez un message similaire à celui-ci :
`Unhandled Exception: System.Runtime.Serialization.SerializationException: Type Nuancier is not marked as Serializable and does not implement ISerializable.`

L'attribut SerializableAttribute (que l'on peut indiquer par <Serializable()>) indique au compilateur que cette classe peut être sérialisée. Notre application remplit la collection "MaListe" et sauvegarde ces informations dans un fichier, à l'aide de la classe System.Runtime.Serialization.Formatters.Binary.BinaryFormatter.

L'avantage de créer une collection est ici que nous héritons de tous ses membres publics tels que Count (pour déterminer le nombre d'éléments), ou Clear (vide la collection), ou RemoveAt(Index) qui extrait un élément, etc.

XML_1.cs

```
using System;
using System.IO;
using System.Collections;
using System.Runtime.Serialization;
using System.Runtime.Serialization.Formatters.Binary;

[Serializable()] public class Nuancier {
    private string Couleur;
    private int CodeRAL;

    public Nuancier(string couleur, int codeRAL) {
        this.Couleur = couleur;
        this.CodeRAL = codeRAL;
    }
}
```

```
public override String ToString() {
    return "[" + this.Couleur + " = " + this.CodeRAL + "];"
}

public class Serialisation {

    public static void Charge() {
        FileStream stream = File.OpenRead("Nuancier.bin");
        BinaryFormatter formatter = new BinaryFormatter();
        ArrayList MaListe = (ArrayList)formatter.Deserialize(stream);
        for(int i=0; i<MaListe.Count; i++) {
            Console.Out.WriteLine("\t" + MaListe[i].ToString());
        }

        stream.Close();
    }

    public static void Sauve() {
        ArrayList MaListe = new ArrayList();
        MaListe.Add(new Nuancier( "Beige vert", 1000 ) );
        MaListe.Add(new Nuancier( "Beige", 1001 ) );
        MaListe.Add(new Nuancier( "Jaune Sable", 1002 ) );
        MaListe.Add(new Nuancier( "Jaune Or", 1004 ) );

        FileStream stream = File.Create("Nuancier.bin");
        BinaryFormatter formatter = new BinaryFormatter();
        formatter.Serialize(stream, MaListe);
        stream.Close();
    }

    public static void Main() {
        Sauve();
        Charge();
    }
}
```

Ce qui donne à l'exécution :

```
#mono XML_1.exe
[Beige vert = 1000]
```

```
[Beige = 1001]
[Jaune Sable = 1002]
[Jaune Or = 1004]
```

Le fichier binaire créé, portant le nom "Nuancier.bin", est éditable à l'aide d'un éditeur hexadécimal comme hexedit (gratuit) sous Windows ou la commande hexdump sous un Shell Linux :

```
#hexdump Nuancier.bin -C
00000000 00 01 00 00 00 ff ff ff 01 00 00 00 00 00 00 00 |.....|
00000010 00 04 01 00 00 00 1c 53 79 73 74 65 6d 2e 43 6f |.....System.Co|
00000020 6c 6c 65 63 74 69 6f 6e 73 2e 41 72 72 61 79 4c |llections.ArrayL|
00000030 69 73 74 03 00 00 00 05 5f 73 69 7a 65 06 5f 69 |list....._size_|
00000040 74 65 6d 73 08 5f 76 65 72 73 69 6f 6e 00 05 00 |tems._version...|
...
```

Cependant, attention : un fichier binaire créé avec Windows n'est pas compatible avec Mono.

Notez que sous Windows, si vous désirez que le flux de données soit sauvegardé de manière absolue sous l'arborescence "C:\ " vous devrez indiquer

```
FileStream stream = File.Create(@"C:\Nuancier.bin");
```

Le format SOAP

Nous avons choisi un exemple encore plus simple qui sérialise la classe Nuancier. Tout comme c'était déjà le cas du format binaire, nous devons ajouter l'attribut [Serializable]. Sous Windows, l'espace de nom System.Runtime.Serialization.Formatters.Soap doit être ajouté comme référence au projet Visual Studio. Sous Linux avec Mono 1.01 il est nécessaire de spécifier de manière explicite l'assemblage de référence lors de la compilation :

```
#mcs XML_2.cs -r:System.Runtime.Serialization.Formatters.Soap
```

Le fichier généré est lisible, car il s'agit en fait d'un format XML, avec une balise enveloppe SOAP :

```
<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema
chema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENC="http://schemas.xmlsoap
.org/soap/encoding/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org
/soap/envelope/" xmlns:clr="http://schemas.microsoft.com/clr/" SOAP
-ENV:encodingStyle="http://
schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
  <a1:Nuancier id="ref-1" xmlns:a1="http://schemas.microsoft.com/clr
/assem/XML_2%2C%20Version%3D0.0.0.0%2C%20Culture%3Dneutr
al%2C%20PublicKeyToken%3Dnull"
  >
    <Couleur id="ref-2">Beige Vert</Couleur>
    <CodeRAL id="ref-3">1000</CodeRAL>
  </a1:Nuancier>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Et voici le code source (minimaliste) :

XML_2.cs

```
using System;
```

```
using System.IO;
using System.Collections;
using System.Runtime.Serialization;
using System.Runtime.Serialization.Formatters.Soap;
```

```
[Serializable]
public class Nuancier
{
  public string Couleur;
  public string CodeRAL;

  public void Charge(string fileName)
  {
    FileStream s=new FileStream(fileName,FileMode.Open);
    SoapFormatter formatter = new SoapFormatter();
    Nuancier MaListe = (Nuancier)formatter.Deserialize(s);

    Console.Out.WriteLine("\t" + MaListe);
    Console.Out.WriteLine("\t" + MaListe.Couleur);
    Console.Out.WriteLine("\t" + MaListe.CodeRAL);
```

```
    s.Close();
  }
  public void Sauve(string fileName)
  {
    FileStream s=new FileStream(fileName,FileMode.Create);
    SoapFormatter sf=new SoapFormatter();
    sf.Serialize(s,this);
    s.Close();
  }
  static void Main(string[] args)
  {
    Nuancier nuancier=new Nuancier();
    nuancier.Couleur="Beige Vert";
    nuancier.CodeRAL="1000";
    nuancier.Sauve("Nuancier.SOAP.txt");
    nuancier.Charge("Nuancier.SOAP.txt");
  }
}
```

Le format XML

Nous allons utiliser l'espace de nom System.XML.Serialization qui nous permet de sérialiser et désérialiser des objets dans des fichiers lisibles (non binaires).

À la lecture du code source ci-dessous nous pouvons constater qu'il y a finalement très peu de modifications au code source.

XML_3.cs

```
using System;
using System.Collections;
using System.IO;
using System.Xml;
using System.Xml.Serialization;
using System.Runtime.Serialization;
```

```
[Serializable]
```

```

public class Nuancier
{
    public string Couleur;
    public string CodeRAL;

    public void Sauve()
    {
        XmlSerializer s = new XmlSerializer( typeof( Nuancier ) );
        TextWriter w = new StreamWriter( "Nuancier.xml" );
        s.Serialize( w, this );
        w.Close();
    }
    public void Charge()
    {
        XmlSerializer s = new XmlSerializer( typeof( Nuancier ) );
        Nuancier Liste2 = new Nuancier();
        TextReader r = new StreamReader( "Nuancier.xml" );
        Liste2 = (Nuancier)s.Deserialize( r );
        r.Close();

        Console.Out.WriteLine("\t" + Liste2);
        Console.Out.WriteLine("\t" + Liste2.Couleur);
        Console.Out.WriteLine("\t" + Liste2.CodeRAL);
    }
    static void Main(string[] args)
    {
        Nuancier nuancier=new Nuancier();
        nuancier.Couleur="Beige Vert";
        nuancier.CodeRAL="1000";
        nuancier.Sauve();
        nuancier.Charge();
    }
}

```

Et voici le résultat XML :

```

<?xml version="1.0" encoding="utf-8"?>
<Nuancier xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Couleur>Beige Vert</Couleur>
  <CodeRAL>1000</CodeRAL>
</Nuancier>

```

Comme on le voit, il est très facile de sérialiser/désérialiser avec .net. Une fois le fichier XML généré, d'autres traitements seront envisageables, par exemple, la transformation du fichier XML en un fichier HTML (via XSLT), ou la recherche d'un élément au sein de ce fichier XML (via X-PATH)...

Convertisseur C SHARP

En aparté, nous avons testé pour vous un convertisseur de code VB.NET vers C#, C-Sharpener for VB, que nous propose la société Elegance Technologies. Après la phase d'installation, celui-ci s'attache à Visual Studio sous la forme d'un plug-in que vous pouvez exécuter à partir du menu Outils. (Fig. 1)

Nous allons le tester avec un projet très simple qui consiste à déterminer quel est l'utilisateur courant, ainsi que ses privilèges. Voici le code initial en VB.NET :

Nous avons vu que le format SOAP (ou XML) n'est qu'un protocole implémenté par un "formatter" dans .NET (SoapFormatter dans le cas de SOAP).

Si vous vous passez de SOAP ou de XML, le traitement sera plus rapide. En effet, l'exécutable .net perd beaucoup de temps en formatage/ déformatage des données SOAP/XML. En clair, cela coûte "toujours plus cher" en terme de temps de traitement que de passer par une sérialisation binaire. Le gain est certain, mais revers de la médaille, si vous formatez en binaire, vos applications Windows .net ne pourront plus communiquer qu'avec d'autres applications Windows .net.

De même, dans le cas par exemple d'un service Web, vous pouvez vous passer de http, et utiliser directement TCP, mais dans ce cas, de nombreux pare-feu ne vous laisseront plus passer...



Fig. 1

Roleutil.vb

```

Imports System
Imports System.Threading
Imports System.Security
Imports System.Security.Permissions
Imports System.Security.Principal

Module Module1
    Sub Main()
        ' Vérification de l'identité de l'utilisateur et du rôle attribué...
        AppDomain.CurrentDomain.SetPrincipalPolicy(PrincipalPolicy.WindowsPrincipal)
        Dim user As WindowsPrincipal = CType(Thread.CurrentPrincipal, WindowsPrincipal)
        Console.WriteLine("Vous êtes {0}", user.Identity.Name)
        If user.IsInRole(WindowsBuiltInRole.Administrator) Then
            Console.WriteLine("Vous jouez le rôle d'administrateur")
        End If
        If user.IsInRole(WindowsBuiltInRole.Guest) Then
            Console.WriteLine("Vous jouez le rôle d'invité (minimum de privilèges)")
        End If
        Console.ReadLine()
    End Sub
End Module

```

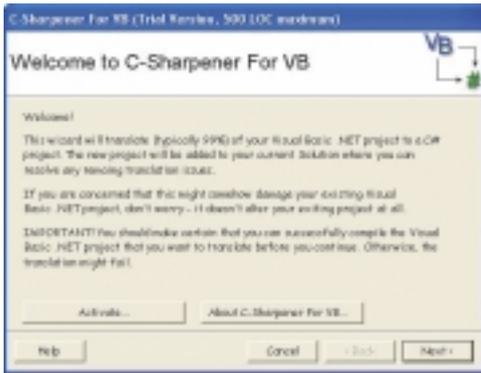


Fig. 2

Ce qui donne à l'exécution (Avec Visual Studio sous Windows) :
 Vous êtes DEIMOS\XL

Vous jouez le rôle d'administrateur

Au lancement, le convertisseur nous avertit que le taux de réussite de la transformation d'un projet VB.NET en C# est de 99 %. Avec la version d'évaluation, seulement 500 lignes pourront être traitées. (Fig. 2)

L'assistant sauvegarde le code source en XML pour analyse et conversion. Vous avez aussi la possibilité d'insérer chaque ligne VB en commentaires pour pouvoir comparer de visu le code traduit. (Fig. 3)

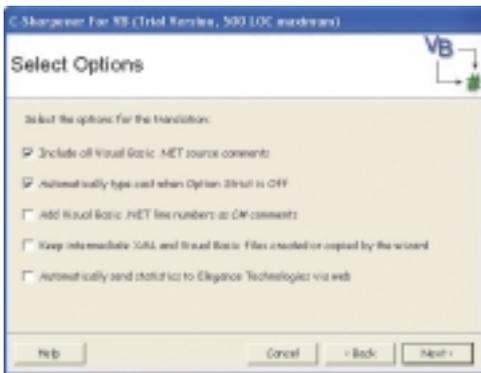


Fig. 3

Voici le code obtenu au final :

```
using System;
using System.Threading;
using System.Security;
using System.Security.Permissions;
using System.Security.Principal;

using Microsoft.VisualBasic;
using System.Collections;
using System.Data;
using System.Diagnostics;
namespace IdentificationUtilisateurVB {
    // following class was VB module
    [Microsoft.VisualBasic.CompilerServices.StandardModule]
    sealed public class Module1 {
        public static void Main() {
            // Vérification de l'identité de l'utilisateur et du rôle attribué...
            AppDomain.CurrentDomain.SetPrincipalPolicy( PrincipalPolicy.
            WindowsPrincipal );
            WindowsPrincipal user = ( ( WindowsPrincipal ){ Thread.
            CurrentPrincipal } );
```

```
Console.WriteLine( "Vous êtes {0}", user.Identity.Name );
if ( user.IsInRole( WindowsBuiltInRole.Administrator ) ) {
    Console.WriteLine( "Vous jouez le rôle d'administrateur" );
}
if ( user.IsInRole( WindowsBuiltInRole.Guest ) ) {
    Console.WriteLine( "Vous jouez le rôle d'invité (minimum de
privileges)" );
}
Console.ReadLine();
}
```

(Fig. 4)

Comme on le voit, il s'agit d'un module VB écrit en C#. En outre, le convertisseur a ajouté des lignes inutiles. Après nettoyage du code, nous avons essayé une compilation sous Linux avec Mono 1 (sans entrer dans les détails sans nettoyer le code la compilation échouera).

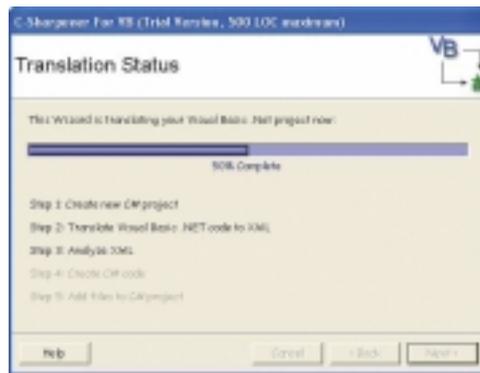


Fig. 4

```
using System;
using System.Security.Principal;
using System.Threading;
namespace RoleUtil
{
    class Class1
    {
        static void Main(string[] args)
        {
            ...
        }
    }
}
```

#mcs roleutil.cs

Compilation succeeded

#mono roleutil.exe

Vous êtes root

Vous jouez le rôle d'administrateur

Nous pouvons constater que le convertisseur fonctionne convenablement. Il pourra à l'occasion rendre service aux développeurs qui doivent migrer un projet vers C#.

■ Xavier Leclercq - Xavier.Leclercq@programmez.com

code a été écrit avec JBuilder, mais sans utiliser le générateur de code et pour cause. Nous commençons avec l'exemple le plus simple : un utilisateur clique sur un bouton. (Encadré 1). Dans le code, nous créons un bouton qui est incorporé à une fenêtre. La ligne de code la plus importante est

```
b.addActionListener(this);
```

Par cet appel, nous ajoutons un écouteur à la liste (auparavant, vide) des écouteurs du bouton. La classe JButton gère entièrement cette liste. Nous donnons "this" comme écouteur du bouton, ce qui dans le contexte, signifie que c'est l'instance de notre classe dérivée de JFrame qui écoute. Nous devons déclarer notre classe comme implémentant l'interface ActionListener. Comment savoir dans chaque cas quelle interface implémenter ? Il suffit de rechercher, dans la Javadoc, les méthodes de nom addXXXXListener appartenant au composant source. La doc renvoie alors à celle de l'interface, et celle-ci à son tour renvoie sur le type d'objet événement que la JVM construit. Une interface ActionListener ne comporte qu'une méthode, actionPerformed, que nous devons implémenter. Dans l'implémentation nous invoquons une méthode de l'objet événement, getActionCommand qui nous retourne une chaîne correspondant à l'intitulé du bouton. Examinons ceci de plus près.

Remonter à la source

Une fenêtre ou un panneau peuvent contenir plusieurs boutons. Or l'interface ActionListener ne sera implémentée qu'une seule fois, c'est bien évident. Comment savoir quel bouton a été cliqué, sachant que quel que soit le bouton c'est une seule et même méthode, actionPerformed, qui est invoquée ? Les générateurs de code résolvent le problème en instanciant autant de classes anonymes qu'il y a de boutons. Nous verrons plus loin ce que sont ces classes anonymes. Le procédé marche fort bien, mais conduit à un code très lourd et particulièrement illisible. En outre, la création d'objet demande du temps et de la mémoire en

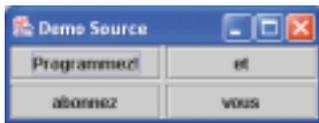


Fig. 2 : Quand un conteneur contient plusieurs boutons, la méthode getSource de l'objet événement permet de savoir qui a émis l'événement.

Java et lorsque l'on recherche l'économie de mémoire, il peut être judicieux de procéder différemment. Les objets événements ont tous une méthode, getSource, qui renvoie une référence sur le composant source. On procède alors ainsi, comme démontré dans l'exemple Demo3. (Fig. 2)

```
public void actionPerformed(ActionEvent ae) {
    Object source = ae.getSource();
    if(source == bouton1) {
        System.out.println("Clic bouton1");
        return;
    }
    if(source == b3) {
        //etc
    }
}
```

Ceci fonctionne toujours, la JVM garantissant l'unicité de la référence sur le bouton et c'est sans doute la meilleure approche, à condition que

l'on ait conservé les références quelque part dans le code. Sinon, on invoquera getActionCommand comme dit plus haut. Le problème majeur de cette approche est la maintenance du code ou, pire, son internationalisation. Une modification du libellé du bouton ou sa traduction, oblige à modifier le code en conséquence, et donc à commettre des oublis qui seront autant de bugs. Un moyen terme consiste, comme montré dans le code Demo2, à découpler le libellé et la chaîne décrivant la commande. Il suffit pour cela d'invoquer la méthode setActionCommand des composants concernés. Ceci permet au moins de centraliser les chaînes devant éventuellement être traduites.

Les adaptateurs

Implémenter l'interface ActionListener c'est très bien, il n'y a qu'une méthode. Dans le cas de WindowListener, par exemple qui contient 7 méthodes, le travail devient vite fastidieux. C'est là qu'interviennent les Adaptateurs. Pour toute interface XXXXListener, il existe un adaptateur de nom XXXXAdapter. Un adaptateur consiste en l'implémentation de l'interface correspondante, avec des méthodes qui ne font strictement rien, mais dont la présence à le mérite de satisfaire le compilateur. Un générateur de code va alors instancier une telle classe tout en la dérivant, pour spécialiser ses méthodes, sous la forme d'une classe anonyme. Le programme Demo4, dont voici un extrait, procède ainsi :

```
public Demo4() {
    addWindowListener(new java.awt.event.WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            OnClosing(e);
        }
    });
}

void OnClosing(WindowEvent e) {
    System.exit(0);
}
```

À notre humble avis, le code mettant en jeu les classes anonymes est fait pour ne pas être relu, tant il est laid. C'est pourquoi le code Demo5 (sur le Cd) propose une autre organisation du même principe, avec une classe non plus anonyme, mais plus simplement interne.

Découplage de la gestion des événements de l'interface graphique

Le lecteur transposera sans peine ce qui vient d'être dit à tous les types d'événements. Nous sortons maintenant des sentiers battus. Dans les cas simples, un JFrame ou un JPanel est en général l'écouteur des composants qu'il contient. Pour les applications complexes, il peut être pertinent de s'organiser autrement et profiter de l'opportunité offerte par le fait que tout objet peut être écouteur, pour découpler le traitement des événements de l'interface graphique. Il suffit d'écrire une classe à cet effet. Bien entendu l'idée n'a pas échappé aux concepteurs de Swing qui ont créé une interface nommée "Action", dont l'implémentation est un objet tel que nous venons de le décrire. Swing propose une implémentation de base, "AbstractAction" qu'il suffit de spécialiser. Cette interface propose de regrouper les événements simples, à savoir clic sur

bouton, sélection de menu et appui sur une touche. En effet, il n'est pas rare qu'une action réagisse à l'identique à chacun de ces événements et l'interface Action est tout ce qu'il y a de pertinente dans ce cas. L'exemple Demo6 change la couleur (rouge, vert ou bleu) d'un panneau suite à un clic sur bouton, une sélection de menu ou une frappe sur les touches R, V, ou B.

Voici notre classe action :

```
class ColorAction extends AbstractAction
{ public ColorAction(String name, Color c, Component co)
  { putValue(Action.NAME, name);
    putValue("Color", c);
    target = co;
  }

  public void actionPerformed(ActionEvent evt)
  { Color c = (Color)getValue("Color");
    target.setBackground(c);
    target.repaint();
  }

  private Component target;
}
```

On peut en apprécier la concision, chose assez rare dans le monde des GUI. Le programme d'exemple crée une instance de cette classe par bouton et bien sûr chaque instance est enregistrée comme écouteur d'un bouton. Les entrées de menu sont ensuite créées comme ceci :

```
JMenu menu = new JMenu("Menu couleur");
menu.add(rougeAction);
menu.add(vertAction);
// etc
```

Automatiquement, les libellés de menu se voient attribuer le membre name de l'action. Il ne reste plus qu'à connecter les touches du clavier à nos actions. Nous employons registerKeyboardAction de JComponent. La Javadoc signale cette méthode comme obsolète, mais nous l'utilisons dans un souci de concision dans le contexte de cet article qui n'est pas un traité sur les ActionMap et InputMap.

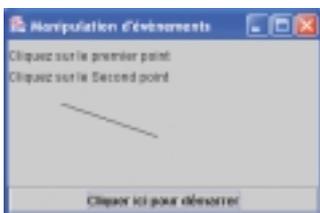


Fig. 3 : Exemple demo7. Nous pouvons directement les événements souris dans la file d'attente de la JVM.

les clics de souris correspondant aux extrémités du trait. Le problème est que nous sommes à l'intérieur d'une méthode de rappel. Tant que celle-ci n'a pas rendu la main, la JVM ne dispatche plus les événements,

ce qui est normal. Mais nous pouvons le faire nous-mêmes comme le montre l'échantillon de code ci-dessous, extrait de Demo7.

```
Point getPoint() {
  EventQueue q =
    Toolkit.getDefaultToolkit().getSystemEventQueue();
  while (true) {
    try {
      AWTEvent event = q.getNextEvent();
      if (event.getID() == MouseEvent.MOUSE_PRESSED) {
        MouseEvent mouse = (MouseEvent) event;
        Point p = mouse.getPoint();

        Point cadre = getRootPane().getLocation();
        p.x -= cadre.x;
        p.y -= cadre.y;

        return p;
      }
    }
    catch (InterruptedException ie) {}
  }
}
```

Tout simplement, nous obtenons une référence sur la file d'événements de la JVM. Ensuite, nous piochons joyeusement dedans jusqu'à rencontrer les événements qui nous intéressent, les autres passant à la trappe.

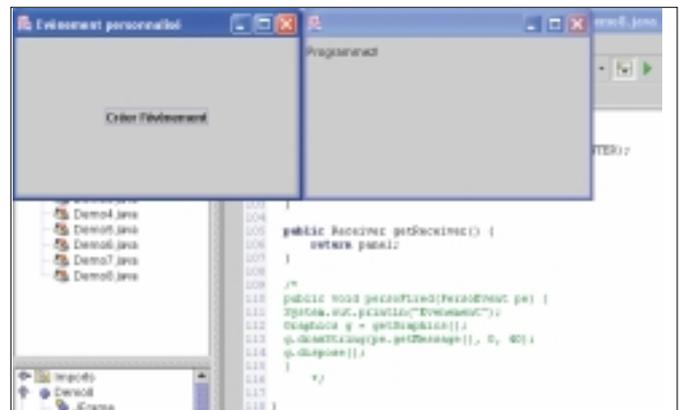


Fig. 4 : Exemple Demo8. Nous insérons un événement personnalisé dans la file pour passer des données à une fenêtre.

Une file d'événements secondaires

Nous pouvons éprouver le besoin de réagir à des événements à l'intérieur d'une méthode de rappel. Supposons que nous avons un logiciel de dessin (Fig. 3). Nous avons cliqué sur le bouton "tracer une ligne" de la boîte à outils.

Maintenant, nous voulons suivre les clics de souris correspondant aux extrémités du trait. Le problème est que nous sommes à l'intérieur d'une méthode de rappel. Tant que celle-ci n'a pas rendu la main, la JVM ne dispatche plus les événements,

Un événement personnalisé

De même que nous pouvons retirer des événements de la file, nous pouvons en insérer. De plus, l'objet événement peut être de notre cru. Ce procédé peut être un bon moyen, par exemple, de passer d'un seul coup des données à une ribambelle d'objets, dont on ne détient pas forcément les références. Demo8, le dernier exemple fait cela. Un événement est émis, qui sert de véhicule à une chaîne de caractères qui sera reçue par l'écouteur recensé. (Fig. 4). Pour parvenir à notre but, nous devons suivre trois étapes. Premièrement, nous définissons une interface écouteur (PersoEventListener) dotée d'une seule méthode (persoFired) que tout écouteur devra implémenter. Deuxièmement, nous définissons notre classe événement (PersoEvent). (Fig. 5) celle-ci dérive

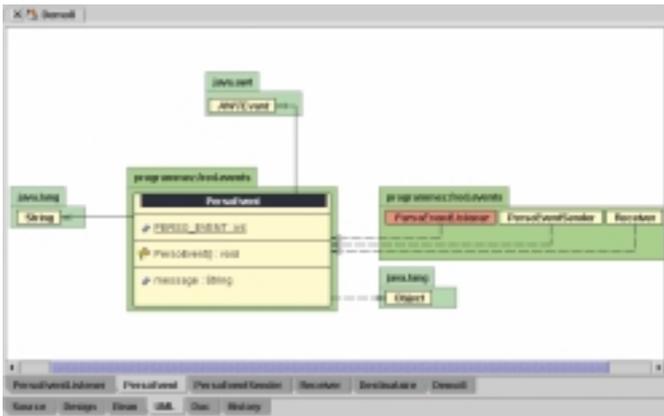


Fig. 5 : Le diagramme UML de notre classe événement personnalisé.

tout bêtement d'AWTEvent. Le point important à noter concerne son ID. Tout événement Java en a un et nous ne devons pas marcher sur les plates-bandes de la JVM en employant un ID réservé. Pour éviter cela, nous donnons :

```
public static final int PERSO_EVENT = AWTEvent.RESERVED_ID_MAX + 1;
```

Troisièmement, nous définissons une classe qui sera la source de l'événement. (Fig. 6) Cette classe doit dériver de JComponent (ou à la rigueur de Component) car si tout objet peut écouter, seul un composant peut poster un événement de type AWTEvent. La méthode postEvent de notre classe instancie un objet de type PersoEvent et le place dans la file. La méthode processEvent sera automatiquement invoquée par la JVM quand celle-ci traitera l'événement que nous avons posté. C'est dans le corps de cette méthode que nous devons invoquer tous les écouteurs

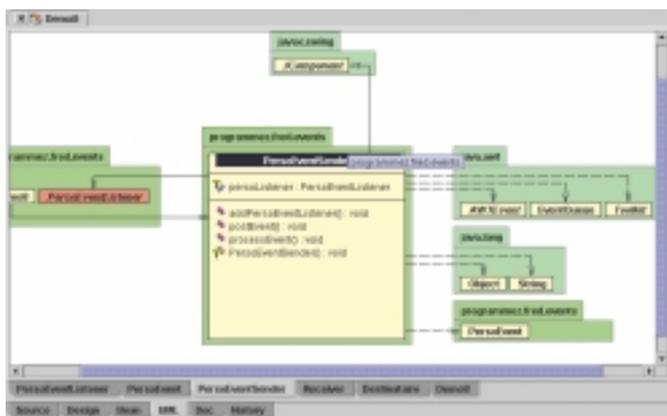


Fig. 6 : Le diagramme UML de notre classe événement personnalisé.

recensés. Attention! Par souci d'alléger le code, notre source ne peut avoir qu'un seul écouteur, mais une programmation complète demande la gestion d'une liste au lieu du simple membre persoListener, point que nous laissons au lecteur, qui devra notamment modifier la méthode addPersoEventListener et écrire une méthode removePersoEventListener. Signalons un dernier point important avant de nous quitter, afin d'éviter un bug sournois. Si l'événement à poster est instancié dans la source, le constructeur de PersoEvent reçoit donc "this" comme source, ce qui est naturel. Dans le cas où l'on souhaite instancier l'objet événement à un autre endroit, ne pas commettre l'erreur de donner "this" (c'est-à-dire n'importe quoi) comme source, mais bien l'objet PersoEventSender. A bientôt pour de nouvelles aventures dans le monde de Java.

encadré 1

Ecoute d'un clic sur un bouton.

```
import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.*;

public class Demo1 extends JFrame implements ActionListener{
    BorderLayout bl = new BorderLayout();
    JButton b = new JButton();

    public Demo1() {
        b.setText("Programmez!");
        b.addActionListener(this);
        getContentPane().setLayout(bl);
        getContentPane().add(b, BorderLayout.CENTER);
        pack();
    }

    public static void main(String[] args) {
        Demo1 demo = new Demo1();
        demo.setDefaultCloseOperation(EXIT_ON_CLOSE);
        demo.setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        System.out.println("Clic !!");
        System.out.println(ae.getActionCommand());
    }
}
```

■ Frédéric Mazué - fmazue@programmez.com