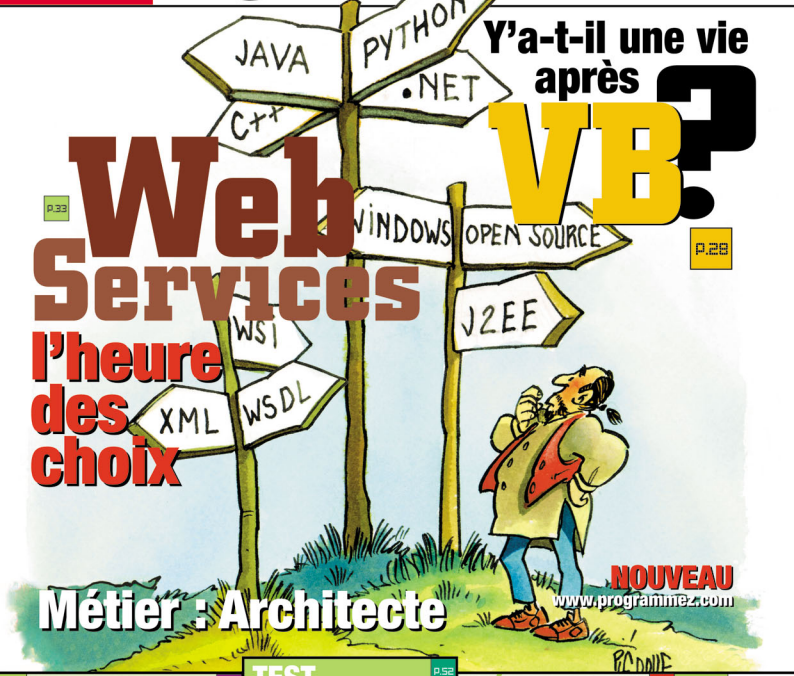
ENQUÊTE : Ces SSII qui recrutent p.24

LE MAGAZINE DU DÉVELOPPEMENT

Programmez !

Mensuel • Septembre 2003 • N°56 • 5,95 €



Y'a-t-il une vie après VB ?

p.28

Web Services

p.33

l'heure des choix

Métier : Architecte

NOUVEAU
www.programmez.com

LUDIQUE

p.79

Jeux 3D : Modélisation en relief

Jeu de Puissance 4 avec **LISP**

TEST

p.52

Quoi de 9 dans **JBUILDER?**

p.64

Programmez avec **Eclipse**

MÉTHODE

p.06

UML pour une meilleure productivité

FRAMEWORK de mapping O/R

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

M 04319 - 56 - F: 5,95 €

Placez vos EJB sous haute surveillance avec PerformaSure

PerformaSure est un outil de suivi permettant de localiser les chutes de performances d'Entreprise Java Beans en situation de production.

La plate-forme J2EE n'est pas la plus simple qui soit. Sa complexité rend difficile la localisation des causes de pertes de performances, mal endémique à Java.
Quest Software propose un outil de suivi des performances des JSP et autres EJB de votre site, construit sur la technologie J2EE. Ceci, aussi bien en développement, qu'en phase de production. Le produit utilise une technologie novatrice dite 'Tag and Follow'.
Voici comment Ed Lycklama, un des co-fondateurs de Quest, nous décrit cette technologie : "Le tag and Follow permet de suivre une (ou plusieurs) transactions dans le serveur d'applications. Les données sont collectées au niveau du composant, de la servlet à la requête JDBC, en passant par le bean session et le bean entité".

Figure 1 : Suivi d'un serveur WebSphere avec PerformaSure.

L'organisation du produit

PerformaSure s'articule autour de trois composants :

- **Les "Agents".** Ceux-ci collectent les informations spécifiques aux objets devant être observés et transmettent les données au composant Nexus. Cela implique qu'il doit y avoir un agent par serveur.
- **"Nexus"** qui est une sorte d'agent de liaison. Il collecte les informations en provenance des agents et les transmet aux "workstations" ;
- Ces **"Workstations"** sont des moniteurs visualisant les résultats, et qui offrent un grand choix de possibilités (figure 1). Chacune de ces possibilités apparaissant dans une fenêtre indépendante (figure 2).

Ce que nous avons apprécié

PerformaSure tourne sur de nombreuses plates-formes, à savoir Solaris, AIX, HP-UX, Linux et Windows (notre plate-forme de test). Outre le serveur d'applications WebSphere, la version 2.0 supporte maintenant BEA Web Logic. Les serveurs Web Sun One, Apache IIS, WebLogic et WebSphere sont supportés. Tout système SGBDR est en principe supporté. PerformaSure permet de tout espionner dans les moindres détails. Il est par exemple possible de suivre les invocations RMI, ou de mettre en évidence un problème d'équilibrage de charge, ou encore de constater les fuites dans un pool de connexions. Les prises de mesures peuvent être effectuées à des fréquences choisies par l'administrateur. Bref, le produit remplit bien son rôle. En condition de production, on a l'impression de voir vivre le serveur.

Ce que nous regrettons

Les serveurs d'applications de Borland (BES) ou de Sun (Sun One Application server) ne sont pas supportés. Par ailleurs si le produit s'installe facilement, sa configuration post-installation est plus ardue. Il faut en passer par l'édition manuelle de fichiers de configuration, que ce soit sur PerformaSure lui-même, ou sur le serveur d'applications ou le serveur Web sur lequel on souhaite opérer. Des outils graphiques auraient été bienvenus. Ceci peut laisser une impression de manque de finition, compte tenu du fait que le produit est onéreux.

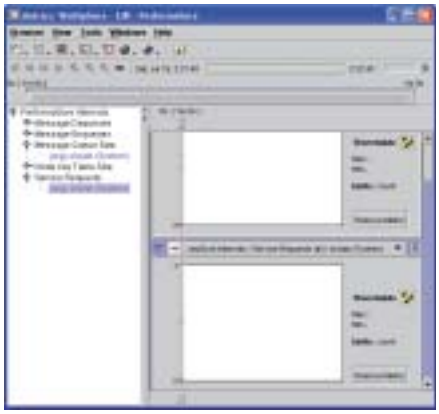


Figure 2 : Suivi de la charge du système.

PerformaSure

- **Editeur :** Quest Software
- **Prix :** environ 33.000€ pour un poste serveur non clusterisé.
- **Configuration requise :** CPU 750Mz avec 1 Go de Ram.
- **Points forts :** Remplit son rôle et permet de localiser les problèmes.
- **Point faible :** Ne supporte que deux serveurs d'applications.

Pour conclure

Bien pris en main PerformaSure vous donnera certainement satisfaction. Toutefois, c'est un investissement qui semble à conseiller dans des cas bien précis : sites importants ou entreprises spécialisées en développement d'EJB.

Frédéric Mazué - *fmazue@programmez.com*

Commentaires de L'éditeur

Le coût de la solution de base (33.000 euros) inclut 1 PerformaSure Nexus, 1 Workstation pour l'administration, 1 licence CPU pour le serveur d'applications, 1 année de support, maintenance, patches et mises à jours.
Nous estimons que performaSure s'amortit en 3 à 6 mois :

- Le produit réduit la durée et le coût de la phase de test, de diagnostic et d'analyse.
- Il réduit également le C.O.Q ("Cost of Quality"), en supprimant les défauts avant la mise sur le marché.

Kulvinder Hari, *European Product Manager - PerformaSure. kulvinder.hari@uk.quest.com.*

Quoi de 9 dans JBuilder ?

Certes, pas d'innovations majeures dans cette "9 mouture ", mais beaucoup d'améliorations, qui font que JBuilder touche à la perfection.

Nous testons aujourd'hui un JBuilder 9 Entreprise en version américaine. L'installation se déroule sans difficultés. Dès la prise en main, on sent que l'on a affaire à un produit soigné. Impression qui se confirme, lors d'une utilisation soutenue. JBuilder 9 est nickel. Pour cette version 9, il semble que l'intention de Borland ait été d'améliorer encore la productivité du développeur, à défaut d'apporter des innovations majeures. D'abord il y a le support de nouveaux outils. Ainsi la prise en charge des systèmes de contrôle de versions CVS, Visual SourceSafe et Clearcase a été mise à niveau. La prise en charge de StarTeam a été ajoutée. Il est maintenant possible d'écrire et de déployer des EJB sur :

BEA WebLogic Platform Server 8.1.
BEA WebLogic Server 6.1 service pack 4
BEA WebLogic Server 7.0, service pack 2
IBM WebSphere Application Server 5.0.
Borland Enterprise Server 5.1.1 - 5.2.1
Sybase EAServer 4.2.

Ce qui n'est pas rien. Nous notons encore l'intégration de la suite d'outils Optimizeit à l'environnement (figure 1).

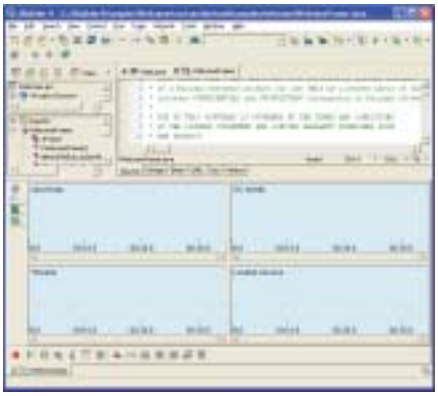


Figure 1 : La suite Optimizeit est désormais intégrée à JBuilder.

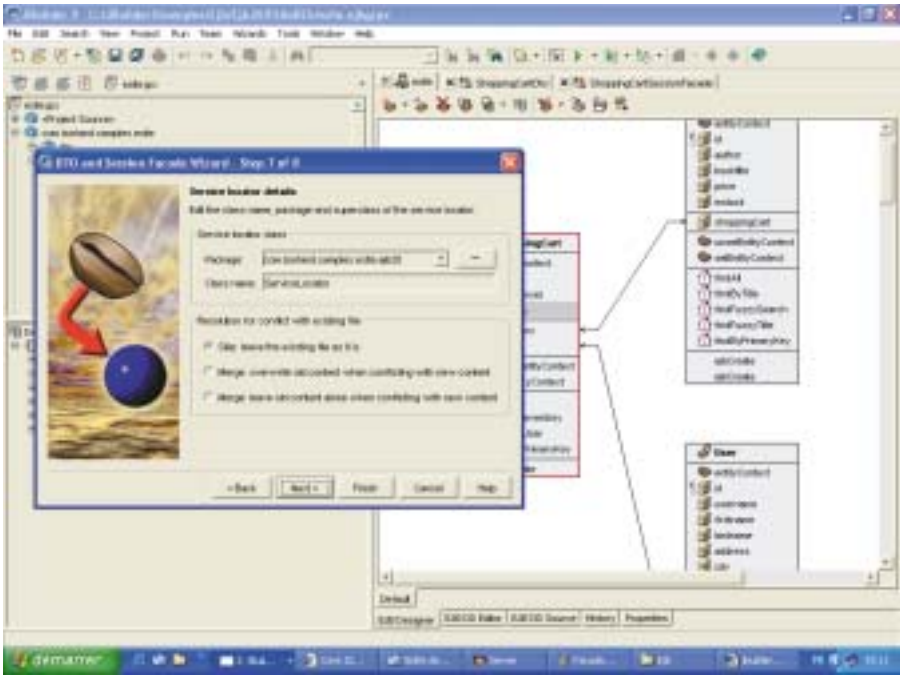


Figure 2 : Un générateur de beans de façades.

Une écriture de code facilitée

Nous avons particulièrement apprécié les améliorations de l'éditeur et de l'outil en général, en ce qui concerne l'aide à l'écriture du code. Cela peut sembler incongru d'en parler dans le test d'une version Entreprise, mais nous pensons que ce sont parfois (et même souvent) les choses les plus simples qui font qu'un outil se place au dessus des autres sur le plan de l'efficacité. Il est maintenant possible de faire du copier-coller entre projets et groupe de projets, ou encore depuis l'explorateur Windows et JBuilder. En outre, il est possible de manipuler les fichiers depuis le panneau de projet. Désormais JBuilder détecte vos erreurs de saisie, à la volée, comme Eclipse. Si vos modèles de code comportent des instructions import, celles-ci seront insérées au bon endroit dans le source. Vous pouvez encadrer du code avec une construction try/catch d'un clic de souris, ou encore remonter à la définition d'une méthode surchargée. Un appui sur la touche F1, alors que le caret est sur un nom de classe ou de méthode, vous renvoie immédiatement à la documentation (y compris la javadoc que vous pouvez avoir générée) dans une fenêtre d'aide, autre que le navigateur d'aide courant. Ainsi vous ne perdez pas votre page. Bref écrire du code Java avec JBuilder 9 est maintenant extrêmement confortable.

Les beans de sessions de façades

Les beans de sessions de façades sont des beans voués à découpler les beans clients des beans entity, ceci, à des fins de performances et de simplifications d'écritures des clients. JBuilder9 vient avec un wizard qui générera le code de tels beans (figure 2). Vraiment puissant !

Pour conclure

Nous n'avons pas décrit toutes les nouveautés du produit. Nous espérons toutefois vous avoir fait saisir l'esprit qui a guidé l'élaboration de cette version d'un produit que nous estimons être le meilleur outil de développement Java du marché.

■ **Frédéric Mazué** - *fmazue@programmez.com*

JBuilder 9 Entreprise

Editeur : **Borland**
Prix : 3 500 Euros
2 099 Euros (mise à jour)
Configuration requise : PII 233 avec 512 Mo de Ram (768 Mo recommandés)

Points forts : *Son efficacité à tous niveaux. Le développement et le déploiement des EJB.*

Points faibles : *Rien de notable.*

10ème TechEd, à Barcelone : 5000 développeurs réunis autour de .NET

Entre des conférences techniques d'un excellent niveau, les démos et annonces de l'éditeur et de ses partenaires, les 5000 développeurs présents à cette 10ème édition du TechEd n'ont vraiment pas eu le temps de visiter Barcelone où se tenait l'événement.
Véritable Université d'Eté des développeurs européens, le TechEd était aussi l'occasion pour Microsoft de dévoiler quelques annonces.

Tech-Ed : .NET, sécurité et migration

Donnant le coup d'envoi au Tech-Ed, Sanjay Parthasarathy, vice président plate-forme chez Microsoft, a lancé d'emblée que le modèle économique de l'Internet s'est déplacé vers le serveur. Microsoft se devait de posséder une plate-forme capable de relever ce défi. C'est tout l'enjeu du lancement tant attendu Windows Server 2003 (WS'03) qui a fort logiquement constitué un des principaux axes de ce 10ème Tech-Ed .

Sur scène, Microsoft a surtout fait l'éloge de la robustesse de WS'03, braquant les projecteurs sur son intégration à un serveur HP 9000 Superdome. L'objectif est de pouvoir rivaliser avec les systèmes Unix. Outre un support amélioré des configurations de grappes de serveurs, l'éditeur de Redmond, faisait ainsi grand cas d'une implémentation intégrale de l'architecture NUMA (Non-Uniform Memory Access), disponible pour la version " datacenter " de WS '03. " IBM nous a aidé en nous fournissant du code spécifique pour mettre en œuvre cette gestion distribuée multi-processeurs de la mémoire " révélait, à cet effet, Jean-Pierre Chabaud, chef de produits infrastructures chez Microsoft France. Et, ce dernier de renchérir que, grâce à l'hyper-threading des processeurs Intel, la parallélisation des opérations devient désormais possible. D'où, selon lui, des gains de performances et une réduction des goulots d'étranglement.

L'argument de la sécurité

La sécurité accrue est l'autre point fort de WS'03 mis en avant au Tech-Ed. " Nous avons réduit les vulnérabilités dans le code et encadrons le déploiement des mises à jour " souligne t-on chez Microsoft. Prochaine étape : l'évaluation WS '03 aux Critères Communs de la sécurité. Microsoft vante une meilleure intégration entre son serveur et Active Directory. L'émission et l'enregistrement en ligne de certificats X.509 est automatisé, mais ne comprend pas de serveur maison OSCP (Online Status Certificat Protocol) pour leur révocation. Les premiers outils de développement basés sur les spécifications de la NGSCB (New Generation Security Board) précédemment TCPA (Trusted Computed Plate-form) sont livrés cet été aux fabricants.

WS'03 leur fournit un environnement d'exécution optimisé, grâce à l'intégration du CLR au kernel. Ce qui n'avait pas été le cas jusque là. Microsoft a claironné les avantages du ciblage d'applications sur n'importe quelle plate-forme fixe et mobile, dont la nouvelle version PocketPC 2003, mais pas encore son "SmartPhone" qui ne supporte pas le CLR. En l'état actuel, WS'03 n'est livré qu'avec un système de fichiers et des services de serveur de terminal, comme le faisait remarquer un grand compte client. Et de poursuivre " Nous attendons la livraison d'un serveur de portails et continuons d'utiliser le serveur messagerie Exchange dans un environnements natif ". Et, d'autre part, sur la nécessité de conserver un environnement de développement dual pour WS'03, pour les applications natives et les applications .NET.

La lente migration des applications vers .NET
L'un des autres arguments choc martelé lors du Tech-Ed a été l'apparente facilité de migration

vers .Net. Preuves à l'appui, Keith Peas, architecte .NET chez Guided Design, a assuré que "La migration d'une application de 10,000 lignes de code n'a pris que cinq minutes et n'a généré que trois erreurs". Les développeurs présents se sont montrés un peu plus réservés. Le basculement vers .NET implique quasi systématiquement une ré- architecture complète des applications. Outre, les contrôles Active-X, les nombreux formats de fichiers d'accès aux données, de type RDO et DAO, sont incompatibles avec .NET. Se voulant à leur écoute, Keith Peas a légèrement nuancé son propos, concédant que pour ce qui concerne une grosse application client serveur de 100,000 lignes de code, programmée en Forms sur une base SQL, au moins 5% des lignes de code n'avaient pas pu être migrées.

.NET ouvre la voie à l'administration du code des applications. Microsoft enjoint désormais ses développeurs d' instrumenter le code, dès la phase de conception de l'application. Pour l'instant, l'interface d'administration WMI (Web Management Interface) est surtout optimisée pour le serveur web, IIS. L'administration de composants .NET au travers d'un API standard suscite quelques interrogations. Les développeurs pointent du doigt les différences de comportement des composants selon le langage employé. Et, à ce titre, de s'étonner "qu'une application développée en VB.NET soit recompilée en permanence pour vérifier des erreurs dans le code, ce qui provoque un ralentissement de l'environnement d'exécution, alors qu'une application écrite en C# n'a besoin d'être compilé qu'une seule fois".

SGBD

Yukon disponible en été 2004

Le successeur de SQL Server 2000 sera disponible vers la mi- 2004. Yukon est un tournant important dans la stratégie SGBD de Microsoft. Cette nouvelle mouture s'apparente à un SGBD traditionnel, enrichi d'un framework spécifique et du CLR. L'outil de mapping relationnel-objet ObjectSpace fera également partie intégrante du framework Yukon. Ne souhaitant pas, à priori, proposer d'outil d'Enterprise Information Integration autonome, c'est Yukon qui sera chargé d'agréger les sources de données structurées et non structurées de l'entreprise grâce au support d'XQuery. Yukon propose aussi un nouveau système de partitionnement qui n'impose plus d'arrêter les serveurs pendant les opérations de maintenance. ■ David Thévenon

INTÉGRATION

Biztalk 2004, clé de voûte de .NET

Cette nouvelle version, disponible d'ici la fin de l'année, a fait sensation. Biztalk 2004 est une évolution majeure du produit. L'outil d'intégration de l'éditeur mue vers un outil d'orchestration de processus, chargé d'assembler les composants .NET. Supportant BPEL4WS, l'éditeur est désormais complètement intégré à Visual Studio .NET 2003. La création des ports, des messages, des processus et leur exposition sous la forme de services web, s'effectue essentiellement par glisser-déplacer et par paramétrage. Les services de Biztalk sont également exposés, via une couche SOAP (Human Workflow Services) et interagissent avec les outils bureautiques de l'éditeur, InfoPath notamment. ■ D.T.

APPLICATIONS MOBILES

.NET compact framework et SQL Server CE à l'honneur

Face à la pression de la communauté Java, Microsoft a multiplié les démonstrations de son savoir faire en matière de développement pour périphériques mobiles. L'éditeur a notamment présenté les prochaines versions de ses Pocket PC et SmartPhone lors de la conférence " Next Generation Pocket PC and Smartphone Development Platform " ainsi que l'utilisation conjointe de SQL Server CE et du .NET compact framework pour bâtir des applications mobiles. Le compact framework est complètement intégré à Visual Studio .NET 2003 et repose sur le même paradigme que les win et webforms. Il est notamment possible d'utiliser des contrôles utilisateurs ASP.NET spécifiques pour construire des applications mobiles. ■ D.T.

Abandon progressif de COM et de MTS

Une question qui est restée sans réponse claire. Tout comme celle de la gestion des transactions dans .Net et la définition d'un modèle de conteneur, condition sine qua non pour pouvoir rivaliser avec les serveurs d'applications. Aujourd'hui, la programmation des transactions avec le framework .NET doit se faire manuellement, apprenait t-on ainsi au détour d'une session consacrée à ce Framework. Selon un représentant de Borland, les inconvénients qui en découlent sont, d'une part, une difficulté à faire évoluer l'infrastructure logicielle sous jacente et, d'autre part, une plus grande complexité dans la mise en œuvre des applications. Qui plus est, .NET marque aussi l'abandon du serveur MTS (Microsoft Transaction Server) ainsi que du modèle de composant COM. Une évolution timidement confirmée, WS'03 est toujours livré avec une version de Com. Cet abandon n'a guère ému les développeurs.

L'accent a surtout été mis sur les développements au niveau de la couche de présentation. La mise en œuvre des ASP.Net profite d'une nouvelle version du serveur IIS. Entièrement

Présentée en juin dernier la nouvelle version Visual Studio 2003, intègre la version 1.1 du Framework .Net, dont la version Compact pour les plates-formes mobiles.

réécrit, " son noyau a été expurgé afin de gagner en performance et en stabilité ", indique Jean-Pierre Chabaud. L'éditeur tchèque Portsight dévoilait ainsi un portefeuille de librairies réutilisables ASP.NET pour faire fonctionner les sites de commerce électronique. Contrastant avec les ASP, la déclinaison .Net fait l'esquisse d'une séparation entre la couche de présentation et la logique métier. Mais Microsoft est resté discret sur l'intérêt de cette approche. Tout en continuant à déconseiller les objets avec état. En effet, la gestion de la logique métier au niveau de la base de données tient toujours le haut du pavé, comme en témoigne l'enrichissement des fonctions métiers de SQL Server 2000.En attendant l'arrivée en 2004 de Yukon, son alter ego .net.

■ Samuel Cadogan

COMMUNAUTÉS

Microsoft a également accueilli de nombreuses communautés françaises telles que DotNetGuru, C2i, CodeS-SourceS, etc. et présenté les lauréats de l'Imagine Cup - un concours international de programmation réservé aux universités et écoles d'ingénieurs. Les applications développées par les étudiants avaient en commun l'utilisation de .NET et de terminaux mobiles (Pocket PC). Particulièrement réussies, elles ont été mises en avant lors du keynote d'ouverture. Une façon pour l'éditeur de démystifier l'apparente complexité de sa plate-forme.

■ D.T.

L'habillage de Windows Server 2003 :

une feuille de route étalée sur trois ans.

➔2003 : applications compatibles CLR :
- Exchange Server 2003 (mise en production)
- SharePoint Portal Server 2003 (1ère version : Share Point Portal Services)
- Content Management Server 2003 (4ème trimestres)
- Identity Integration Server 2003 (4ème trimestre)
- Metadirectory Services 2003 (4ème trimestre)
- Small Business Server 2003

➔2004 : intégration et administration :
- Jupiter/ Biztalk 2004 (bêta 1 disponible cet été)
- Systems Management (disponibilité NC)
- Microsoft Operations Manager (disponibilité NC)
- Yukon – successeur de SQL Server (bêta 1 disponible cet été)
- Whidbey – (prochaine version de Visual Studio .NET)

➔2005/6 : système et développement :
- Blackcomb – successeur de Windows Server 2003
- Longhorn - successeur de Windows XP.
- Orcas – (nouvelle version de Visual Studio .NET)

La migration de Microsoft vers .NET, qui a de quoi donner le vertige, occasionne déjà une succession de décalages dans le calendrier de lancement des applications pour Windows Server 2003. Ce chantier ne devrait s'achever qu'en 2006.

[*] liste non exhaustive

www.programmez.com
www.programmez.com
www.programmez.com
www.programmez.com
www.programmez.com
www.programmez.com

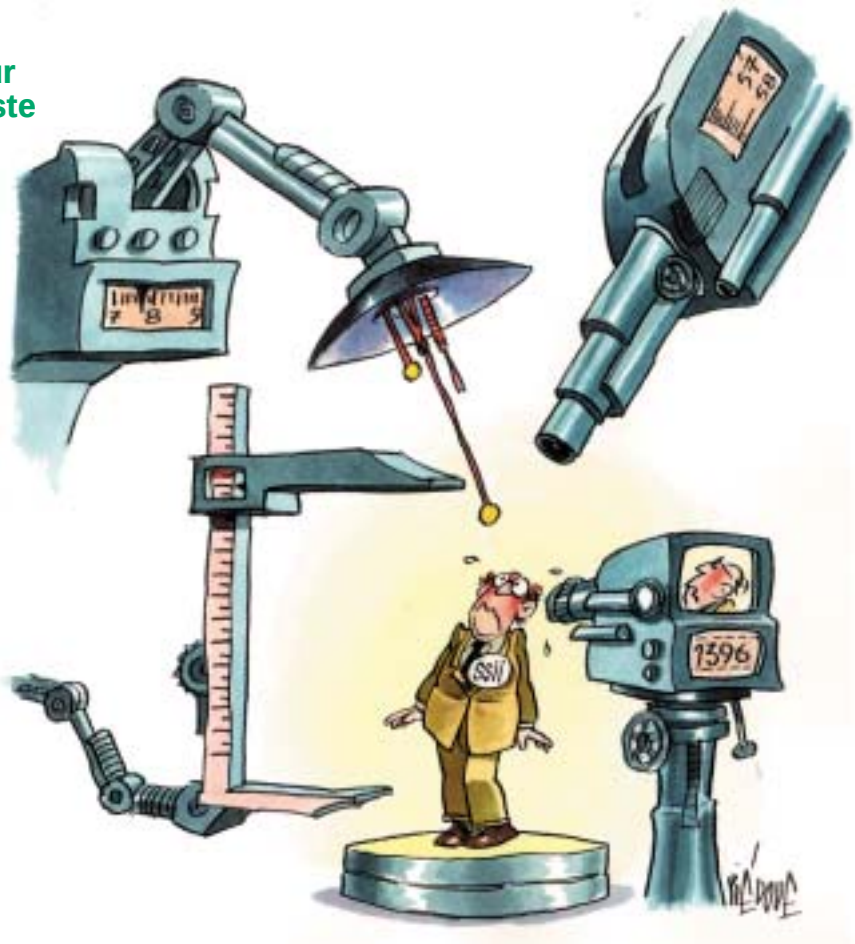
L'univers du développement :

• l'actualité •
• Le téchargement •
• **Le forum technique : questions réponses** •
• L'emploi •
• Les archives du magazine •

Ces SSII qui recrutent :
Profil recherché : cadre de 2 à 5 ans d'expérience

1ère partie

Les SSII qui ont survécu restent extrêmement prudentes quant à leur recrutement. La crise a sans conteste tiré le marché vers le haut. L'exigence des donneurs d'ordre entraîne de facto celle des futurs employeurs. Une montée en compétence qui favorise les candidats spécialisés en informatique et plus expérimentés.



La santé du secteur des SSII est si étroitement liée à la conjoncture économique et géopolitique que nul se s'aventure aujourd'hui à avancer des pronostics sur l'emploi. Après une déprime entre 1991 et 1996, le secteur connaît une embellie jusqu'en 1999. Si l'an 2000 cristallise toutes les euphories, dès 2001 le ralentissement se fait sentir pour atteindre une décroissance en 2002. Mis à part certaines prestations d'externalisation qui ont connu des croissances à deux chiffres jusqu'à fin 2002, les prestations d'assistance, de conseil, d'intégration ont subi une forte décroissance. *"Le développement pur et dur de programmation est l'activité qui a le plus souffert sur ces trois dernières années, précise Lionel Terral, pdg de Ajilon France. La dynamique provient aujourd'hui des activités de maintenance de système informatique et d'infogérance. Les entreprises sont dans une logique de besoins quotidiens, comme la gestion des parcs ou celle des grands applicatifs, plutôt que dans le développement de nouveaux programmes. La maintenance informatique et l'infogérance tirent donc leur épingle du jeu."* Selon le cabinet PAC, la mauvaise surprise de fin d'année vient du monde scientifique et technique, dont la dynamique encore positive début 2002 a franchement viré au négatif. Les entreprises ont une vision très financière et privilégient encore le court terme au long terme.

Quelles perspectives pour 2003 ? La reprise serait-elle à la porte de ce secteur ? Certains prestataires commencent à ressentir un léger frémissement du marché. Il faudra sans doute attendre début 2004 pour se prononcer plus franchement sur une véritable fin de cycle. Pour le recrutement, la conséquence sur ce marché attentiste est double. D'une part, les salariés en poste ont l'impression de stagner car les budgets formation sont réduits. Ils sont aussi envoyés sur le même type de mission afin de capitaliser sur leur expertise. D'autre part, le niveau de recrutement s'est accru : expérience professionnelle, compétences techniques mais aussi fonctionnelles sont très recherchées.

La polyvalence et l'expérience sont largement plébiscitées

La demande porte souvent sur des profils de jeunes cadres qui justifient de 2 à 5 ans d'expérience. *"Le jeune cadre compétent, efficace et pas trop cher est le profil le plus recherché sur le marché, estime Patrick Girodroux, consultant en ressources humaines à l'APEC. Les profils comme ceux de consultants en système d'information ou consultants AMO passent moins bien car les entreprises sont dans une logique de résultats court terme. Or, ce sont des fonctions dont la valeur ajoutée et l'expertise se mesure dans le temps."* Avec une visibilité de 15 jours à deux mois, les SSII sont entrées dans une logique de projets d'urgence à court terme. Certains postes à très forte valeur ajoutée, comme ceux du milieu des systèmes et réseaux, offrent des perspectives. Ils restent cependant ouverts à des candidats expérimentés. L'accès à l'emploi pour les jeunes diplômés est plus restreint. Il en va de même pour la population des quadras. Ces derniers sont souvent d'excellents fonctionnels mais restent "chers" du fait de leur âge.

Il y a encore peu, les SSII devaient séduire les candidats. Aujourd'hui, avec le retournement

de la conjoncture ce sont les candidats qui doivent convaincre les employeurs. Il y a pléthore de candidats, et rareté des offres. La première conséquence pour les SSII est de moins subir la pression des salaires. Globalement, le niveau des salaires dépend des postes, des profils, de l'expérience. Actuellement, il est beaucoup plus en phase avec la valeur des personnes et des expériences. Un jeune ingénieur peut espérer commencer à 27 Keuros/30 Keuros par an. Il ne doit pas cependant s'attendre à une évolution aussi rapide que sur la période 97 à 2001. Avec 2 à 3 ans d'expérience, les salaires peuvent atteindre 35 Keuros Certains professionnels notent cependant une certaine dissension entre les attentes des jeunes diplômés et la réalité du marché à ce niveau-là. N'oublions pas que des profils plus expérimentés se retrouvent sur le marché à la suite de licenciements et se vendent moins chers.

■ **Anne-Françoise Moal**

(suite au prochain numéro)

SSII : Premier recul depuis 10 ans

■ **Syntec :** En 2002, le marché des sociétés de services informatique a connu son premier recul depuis dix ans. Le Syntec informatique estimait ainsi pour l'ensemble de l'année, un recul du marché de 2% à 5%. Le secteur reste cependant hétérogène : l'infogérance qui représente 25% du secteur devrait croître de 10%, alors que le conseil, l'intégration de systèmes et l'ingénierie devraient décliner de 5%. La situation économique s'explique par plusieurs facteurs. D'une part, le retournement du marché des télécoms. D'autre part, la réduction des budgets d'investissement des clients, notamment par une pression sur les prix (baisse de 5% à 10%). Dans ce contexte, les SSII devraient améliorer leur productivité. En 2002, les rapprochements des sociétés informatique avec les grands du conseil se sont multipliés. En 2003, les phénomènes de concentration devraient se poursuivre.

■ **APEC :** Quant aux perspectives 2003, l'association pour l'emploi des cadres, ne mise pas sur un regain d'optimisme. Selon cet organisme, l'optimisme des entreprises continue de s'amoindrir. Alors qu'en 2001, 58% des entreprises comptaient embaucher plus de cadres, en 2003, elles ne sont plus que 29%. La tendance est la même concernant l'augmentation des effectifs salariés (voir graphiques). Parallèlement, la proportion d'entreprises souhaitant réduire leurs effectifs cadres est passée de 2% en 2001 à 8% en 2003. Il en va de même pour les effectifs salariés.

Analyse du marché

Le marché du logiciel et des services informatique "frémit"

Malgré les avatars sociaux, le mois de mai et le début de juin semblent confirmer la tendance exprimée en avril : le marché du logiciel et des services informatique "frémit". Plusieurs éléments, encore fragiles cependant, sont à noter à l'appui de cette thèse :

- le flux commercial (propositions, cahiers des charges) est nettement supérieur à celui du 1er trimestre.
- la baisse des prix semble en voie d'accalmie, et une stabilisation semble proche
- après 2 années de gel, certains utilisateurs semblent prêts à moderniser leurs parcs de matériels et d'applications

Ce "mieux" ne concerne toutefois pas l'ensemble du marché. Le "frémissement" est surtout visible en conseil et sur le marché des projets (intégration de systèmes) et des progiciels applicatifs. Par contre, on voit peu de signes encourageants dans l'assistance technique. D'autre part, la concurrence est de plus en plus tendue sur les marchés d'externalisation (TMA, infogérance). Il faudra attendre les prochains mois, et surtout le climat de la rentrée de septembre pour savoir si le marché français est revenu durablement sur la pente ascendante.

Source : Pierre-Audoin Consultants – Analyse de Jean-François Perret

AJILON :

Une logique qualitative de recrutement

Lionel Terral, pdg d'Ajilon France : "Le recrutement monte en gamme"

Ajilon a été créée en 2002. Société du groupe Adecco, elle est spécialisée dans l'externalisation de services et la gestion de projets et de compétences. Elle est dotée de 4 entités : I.T. consulting, Engineering, Sales&Marketing, Recruiting. Ses activités informatiques regroupent des services utilisateurs, des services de production informatique (maintenance de système ou d'application) et de l'assurance qualité logicielle, actuellement en développement, du conseil. Depuis le début de l'année, le nombre de salariés a considérablement évolué. La société est passée de 1089 collaborateurs à 1414 en France de janvier à juin 2003 (27 000 collaborateurs à travers le monde). Sur 2003, Ajilon envisage près de 600 recrutements, dont 100 en informatique, 60 en ingénierie, 300 en Sales & Marketing. Ces recrutements correspondent au turnover ordinaire et aux objectifs de croissance de l'entreprise.

"Nous recrutons moins de jeunes diplômés que de trentenaires qui justifient de 5 ou 6 ans d'expérience. Leur évolution sur des métiers différents les rends adaptables et représente un atout de taille, précise Lionel Terral, pdg de Ajilon France. Face à un même niveau d'expertise technique, la différence se joue sur la capacité d'adaptation aux nouvelles organisations et l'expérience de différents métiers. Pour répondre aux attentes des clients, nous apportons un service beaucoup plus qualitatif. Il y a encore 2 ou 3 ans, ceux-ci nous demandaient avant tout du volume. Aujourd'hui, la demande a évolué du volume vers le qualitatif."

"Nous maintenons le cap avec une vision à long terme"

Les entreprises qui recrutent aujourd'hui sont celles qui ont gardé la tête froide lors des envolées de l'an 2000. Homsys a su miser sur une politique de recrutement très constante qui, aujourd'hui, lui permet de continuer à recruter.

Yannick Gonneau, pdg de Homsys Group

Crée en 1991, Homsys est une société spécialisée dans les services et conseils informatiques architecture et business. La société comptabilise plus de 230 salariés dont 227 effectifs cadres, et recrute quasi exclusivement des ingénieurs de grandes écoles ou niveau DESS. "Nous n'engageons que des ingénieurs qui ont une spécialité informatique, précise Yannick Gonneau, PDG de Homsys Group. Il y a encore 2 ou 3 ans, les entreprises n'hésitaient pas à engager des ingénieurs travaux publics, chimie, etc. Notre politique de recrutement a été de ne pas s'éloigner de notre axe informatique : architecture, business intelligence, développement web et objet. "Selon les métiers, les profils varient mais restent complémentaires. L'architecture requiert des profils seniors avec 15 ans d'expérience, le développement web et objet des jeunes diplômés et des ingénieurs confirmés. Le business intelligence nécessite une maîtrise technique et une expertise fonctionnelle (RH, management, etc).

Homsys a misé sur la constance de sa politique de recrutement. En restant sélectif tant sur les compétences recherchées que sur le niveau de salaire, la société n'a pas cédé aux sirènes de la période bulle Internet. "Nous n'avons jamais fait de surenchère au niveau des salaires, affirme-t-il. Aujourd'hui, de nombreuses sociétés ont disparu. Nous étudions de près le dossier de certaines d'entre elles afin de les intégrer au groupe. Nous refusons les dossiers, lorsqu'il n'y a pas de cohérence dans le savoir-faire par rapport à nos métiers. Nous misons sur une cohérence formation initiale et salaire."

Homsys dissocie ses recrutements de l'affectation sur un projet, alors que la plupart des SSII recrutent souvent une personne pour un projet donné. La société se comporte plus en industriel qu'en SSII, qui vise avant tout le court terme. "Cotée en bourse depuis le 11 juillet 2003, la société part sur des bases saines et un recrutement qui se poursuivra sur la base de 80 personnes en 2003."

Unidirect mise sur la valeur humaine

Valérie Vital, directrice du département service et marketing chez Unidirect.

Créée en 1995, Unidirect est spécialisée dans l'intégration, l'administration et la sécurité des systèmes d'information. Depuis deux ans, elle développe également une activité de consulting et services (délégation de personnel). Ce pôle fonctionne comme une SSII. "Nous recherchons des profils Bac +4/5 avec une spécialisation informatique, explique Valérie Vital, directrice du département service et consulting chez Unidirect. Nous cherchons des candidats ouverts et souples, capables de s'adapter aux différentes missions de nos clients. Nous prenons le temps de sélectionner les candidats par un cabinet de recrutement. "Quant à la conjoncture, Valérie Vital constate un petit frémissement. Alors qu'en janvier c'était le calme plat, certains clients reviennent à la délégation. "Il est encore trop tôt pour se prononcer, estime-t-elle. La situation reste tendue car on ne sait pas si les contrats seront reconduits. Début 2004, la visibilité sera sans doute meilleure." Les postes proposés sont classiquement ceux d'administrateurs réseau et système mais aussi et de plus en plus, des fonctions gravitant autour de la qualité de service, du reporting, de la sécurité ou encore des bases de données.

Avanade :

Expérience professionnelle ouverture internationale font la différence

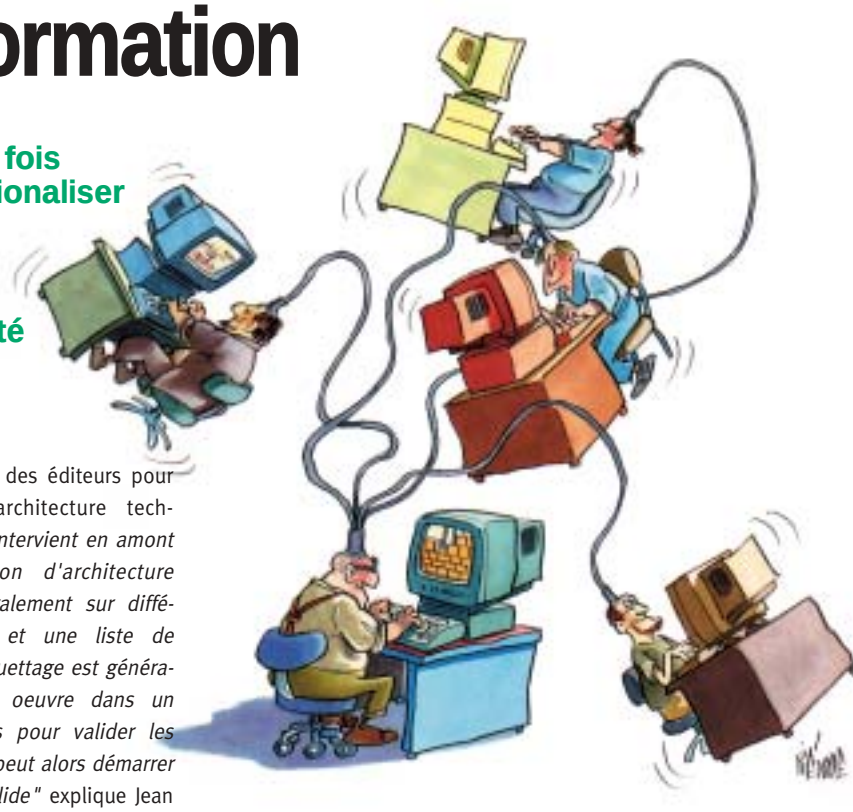
Avanade a été Créée en avril 2000 par Accenture et Microsoft aux USA. Une implantation internationale l'installe également en France dès août 2000. Son activité principale est l'intégration et spécifiquement celle des solutions basées sur les technologies Microsoft. Elle comptabilise 1.500 collaborateurs dans le monde dont 60 en France. Avec un effectif quasi exclusivement de cadres, la société reste exigeante sur les profils des candidats. "Nous recherchons d'abord une très forte compétence technique, précise Valérie Vezinhet, DRH chez Avanade France. Une expérience dans le conseil et la relation client au sens large est un deuxième critère de recrutement. Quant aux profils seniors, une expérience dans le management de projet est indispensable."

Objectifs : doubler les effectifs

A sa création, Avanade recrutait des profils seniors. Aujourd'hui, la société recrute à tout niveau de "seniorité", l'objectif étant de doubler les effectifs. Les portes sont ainsi ouvertes aux jeunes diplômés." Ils doivent cependant pouvoir s'appuyer sur des stages probants, précise-t-elle. Notre activité exige une grande mobilité géographique car nos projets sont plurinationaux, européens au moins. Sans oublier une pratique courante de l'anglais. Ces trois critères associés réduisent considérablement le nombre de profils disponibles. Quant aux salaires, ils sont ceux du marché. Je constate d'ailleurs qu'une meilleure compréhension du niveau des salaires commence à se faire jour chez les jeunes qui arrivent sur le marché."

Architecte : au cœur de la mutation du système d'information

L'architecte propose des approches à la fois techniques et méthodologiques pour rationaliser les développements et assurer la cohérence globale du système d'information. En prenant de la hauteur, il apporte flexibilité, réactivité et pérennité au système d'information.



A la fois guide et gendarme

"Comme dans le BTP, l'architecte conçoit la structure d'un ouvrage. Parfois il peut même être le responsable du chantier, mais ce n'est pas le cas général. On distingue deux types d'architectes. L'architecte fonctionnel est un "super-concepteur" tel qu'un analyste UML, mais avec une dimension SI. L'architecte technique est un "super développeur", capable de choisir les matériaux de base, de créer de la méthode, et de la faire appliquer" résume Pierre Pezziardi, Directeur technique d'Octo Technology.

L'architecte fonctionnel formalise et optimise les processus métier de l'entreprise. Il cartographie l'existant, factorise les nouveaux besoins, et propose un plan de projet à long terme, qui apporte une cohérence globale aux applications des utilisateurs. En bref, si l'architecte joue son rôle, plutôt que d'avoir 10 applications qui se recoupent, les utilisateurs disposent de 3 applications parfaitement complémentaires. L'architecte technique assure la jonction entre l'architecte fonctionnel (quand il existe) ou les directions fonctionnelles et se charge de traduire les besoins métiers en une architecture technique cible. Le cahier des charges technique traduit les "drivers" fonctionnels par des propriétés techniques. "Flexibilité" devient "découplage", "pérennité" devient "standard", etc. L'architecte technique conçoit dans un second temps l'architecture applicative dont il assure la cohérence et l'évolution : doit-on utiliser un framework ou plutôt recourir à des progiciels métier, combien y aura-t-il de couches et quel sera le niveau de granularité des composants, etc. C'est également lui qui est chargé de choisir les technologies des éditeurs pour implémenter l'architecture technique cible. *"Il intervient en amont car une mission d'architecture débouche généralement sur différentes options et une liste de risques. Un maquettage est généralement mis en oeuvre dans un deuxième temps pour valider les choix. Le projet peut alors démarrer sur une base solide"* explique Jean Pierre Lorre, consultant senior chez Valtech. Garant des bonnes pratiques, il édicte aussi les " règles du jeu " que doivent respecter les développeurs.

Compétences : un mélange subtil entre expérience, rigueur, pédagogie et curiosité

Quelle que soit sa formation initiale, l'architecte est avant tout un homme d'expérience qui connaît toutes les facettes d'un projet informatique et qui possède une connaissance approfondie du fonctionnement de l'entreprise. Cette expérience lui apporte une crédibilité indispensable auprès de ses interlocuteurs.

Un architecte n'est ni un gourou, ni un expert en blouse blanche. Il met régulièrement les mains dans le code pour expliquer, et au besoin démontrer, aux équipes de développeurs, l'intérêt d'une architecture plutôt qu'une autre. *"Les architectes de F.R.A. puis ceux de Business Interactif sont toujours des gens qui ont pratiqué le développement et qui ont une véritable reconnaissance de la part des développeurs. Je me méfie des théoriciens qui ne pratiquent pas... Les mains dans le cambouis, ça aide toujours à garder les pieds sur terre, et tous nos architectes sont au cœur des projets"* confirme Jean-Louis Bénard, ancien directeur technique de Business Interactif et fondateur de Brainsonic.fr, un site de formation destiné aux architectes.

Pour transmettre son savoir aux équipes, *"la pédagogie et le sens de l'écoute sont très importants"* ajoute Raphaël Mansuy, consultant Senior et fondateur de Business Design Consulting (BDC).

"Curiosité technique et rigueur sont indispensables pour devenir architecte" complète Jean Pierre Lorre.

Ses outils ? Bon sens, expérience et méthodologie

"D'un point de vue méthodologique nous nous appuyons sur une démarche formalisée de définition d'architecture fonctionnelle et technique (two-track process) s'inscrivant au sein du processus projet basé sur UP (Unified Process). Cette démarche formalise les différentes étapes nécessaires à l'élaboration des architectures du système à étudier" explique Jean Pierre Lorre. Bien que tous les architectes utilisent des outils de modélisation, c'est surtout la connaissance des technologies, des

(suite page 36)

Exemples de sites destinés aux architectes

- www.martinfowler.com
- hillside.net/patterns
- www.posa.uci.edu
- www.phppatterns.com

Cinq formations pour devenir Architecte

Type	Formation	Université
DESS	Architecture des systèmes d'information et de communication	Toulouse 1 et 3
DESS	Audit et conception de système d'information	Nancy 1
DESS	Système d'information et de connaissance	Paris 1
DESS	Systèmes d'information avancés	IAE Lyon
Ecole d'ingénieur	Architecture des systèmes d'information	Insa Rouen

"Une activité pilotée par les risques"

L'architecte effectue les choix structurants (architecture et technologie) en collaboration avec le client et le chef de projet. En tant que prestataire externe, l'un de nos rôles est aussi de rassurer le client quand les choix sont délicats. Cette tâche à haut risque demande de l'expérience à la fois technique et métier. C'est pour cette raison que la majorité des processus liés à la fonction d'architecte sont pilotés par les risques. Il est en effet impossible de construire un immeuble de 30 étages sans de solides fondations.

Architecte : *Jean-Pierre Lorre*
Fonction : *consultant senior*
Entreprise : *Valtech Toulouse*
Formation initiale : *Ingénieur ENSI Caen*
Expérience : *15 ans*
Salaire : *entre 45 et 60K€*
Outils utilisés au quotidien : *méthodologie "two-track process" basée sur UP (Unified Process)*

"Utiliser son expérience pour faire réussir un projet et une équipe"

Le métier d'architecte c'est comprendre le métier de son client, son contexte et ses contraintes. C'est ensuite, bien connaître le potentiel et les limites des outils. Enfin, c'est connaître les bonnes et mauvaises pratiques en matière de construction de logiciels, afin de tracer les plans et définir le cadre d'un système complexe. Un architecte intervient au début d'un projet pour assister le chef de projet et aider les concepteurs à choisir un modèle d'architecture.
Il assiste ensuite l'équipe pour définir le cadre du système et en valide les grands principes par la réalisation d'une maquette. Il intervient tout au long du projet pour veiller au respect des règles du jeu et pour affiner la stratégie définie au départ. Etre architecte c'est avant tout être pédagogue, être à l'écoute et utiliser son expérience pour faire réussir une équipe.

Architecte : *Raphaël Mansuy*
Fonction : *Consultant Senior*
Entreprise : *Business Design Consulting (BDC)*
Formation : *licence-maîtrise-DESS bases de données et intelligence artificielle après un DEUG de mathématiques, MCAD.NET, MCSD.NET, MCDBA, Rational Certified Consultant*
Expérience : *8 ans*
Salaire : *entre 50 000 et 70 000 € par an*
Outils utilisés au quotidien : *Rational XDE, UML, RUP, Visual Studio .NET, Biztalk Server, SQL Server, HIS, etc.*

"Une fonction en mutation"

Le rôle de l'architecte prend de plus en plus d'importance. Après avoir tenté d'exercer son métier dans la sphère technique pour limiter la prolifération technologique des années "downsizing", on lui confie maintenant des missions de gouvernance des frontières et de la donnée. Beaucoup de DSI se réorganisent pour donner plus de pouvoir aux architectes, et réalisent que la fonction ne s'exerce pas d'en haut (équipe d'urbanistes en blouse blanche), mais d'en bas, dans chaque brique du système d'information.

Architecte : *Pierre Pezziardi*
Fonction : *directeur technique*
Entreprise : *Octo Technology*
Formation : *Centrale Lyon*
Expérience : *10 ans (l'important, c'est d'avoir traversé toutes les phases d'un projet et d'avoir travaillé avec des mentors rigoureux)*
Salaire : *entre 70 et 120 K€*
Outils utilisés au quotidien : *PowerPoint, Visio. L'architecte d'application utilise des outils de case plus proches du code : Rational, etc.*

" L'architecte industrialise les développements "

L'émergence du métier d'architecte correspond à l'industrialisation des développements. On peut avoir les meilleurs développeurs, si l'application n'a pas été correctement conçue au départ et si cette conception n'a pas été réellement mise en œuvre, il est probable que les résultats seront catastrophiques. Un architecte ne conçoit pas simplement en dehors de toute implémentation, il doit être au cœur des projets et de l'opérationnel. Je pense que la satisfaction d'un architecte logiciel est proche de celle d'un architecte de bâtiments : celle d'avoir contribué à la réalisation d'applications opérationnelles, maintenables, réutilisables, reposant sur des schémas de conception dont l'efficacité fait la beauté.

Architecte : *Jean-Louis Bénard*
Fonction : *fondateur et directeur technique*
Entreprise : *Brainsonic.fr*
Formation : *Centrale Paris, Option Systèmes d'Information*
Expérience : *10 ans*
Outils utilisés au quotidien : *outils de modélisation*

Des prescripteurs courtisés par les éditeurs

Parce que les choix d'architectures qu'ils effectuent influencent largement les technologies mises en production, les architectes sont de plus en plus courtisés par les éditeurs de logiciels.
Marc Gardette, Responsable de groupe architecte, Microsoft France, MSFT : "Du point de vu de l'éditeur, l'architecte est un prescripteur qui traduit des besoins métiers en design

standards et des principaux "design patterns" qui sont utiles : *"Les outils que j'utilise au quotidien sont des outils de modélisation et de conception mais une grande partie du travail – la plus difficile – se fait avec une simple feuille de papier. C'est là que l'on accouche des schémas essentiels"* confirme Jean-Louis Bénard, fondateur de Brainsonic.fr. Appelés aussi "design patterns", ces schémas synthétisent les bonnes pratiques d'architecture à des niveaux de granularité différents. Service Oriented Architecture (SOA) propose, par exemple, une architecture globale, applicable à l'entreprise, tandis que Model Vue Controler (MVC) se limite à la couche de présentation des applications. *"Les design patterns sont le compagnon essentiel de l'architecte, que j'élargirais même aux patterns tout court : schémas reproductibles de ce qui fonctionne, de ce qui ne fonctionne pas (anti-patterns), que ce soit sur le plan applicatif, hardware, méthodologie, gestion de projet... Enfin Internet et tous les outils de veille sont un complément indispensable"* précise Jean-Louis Bénard.

Un métier enfin reconnu mais en continuelle mutation

"La fonction d'architecte commence à être reconnue et les pratiques se formalisent peu à peu au travers de concepts tels que les Design Patterns, les frameworks, etc. L'architecte est un acteur nécessaire de tout projet informatique d'envergure" estime Raphaël Mansuy. Le développement d'applications demande un champ de compétences de plus en plus important : architecture applicative, intégration, etc. Si bien que les architectes pourraient être peu à peu, amenés à se spécialiser. *"On peut penser qu'une certaine classe d'architectes, plutôt dédiés à un pan de l'offre, verra le jour"* prédit Jean Pierre Lorre. *"C'est tout le métier du développement applicatif qui est en train de connaître sa révolution industrielle, son passage de l'artisanat à l'industrie. Cela implique une véritable prise de conscience de la nécessité d'aller vers plus de valeur ajoutée, car demain je crains qu'il n'y ait plus beaucoup de travail en France pour du développement "de base". En revanche on aura toujours besoin de gens proches du client pour le comprendre (avec sa culture), pour formaliser et pour concevoir l'architecture applicative qui répondra à ces besoins"* conclut Jean-Louis Bénard.

■ **David Thévenon**

technique et fait des choix d'implémentations. Sa démarche consiste souvent à éliminer itérativement des architectures et technologies potentielles, pour mettre au point une plate-forme cible dans un cadre cohérent avec l'existant. Nous avons donc fortement investi envers ce public pour expliquer nos technologies et les meilleures pratiques associées. C'est l'objectif du site Patterns & Practices (msdn.microsoft.com/practices) qui guide les architectes dans leur choix d'architecture, puis d'implémentation. Visual Studio .NET Architect Edition leur permet également de matérialiser leurs choix d'architecture directement dans l'IDE - via les Entreprise Template – pour les communiquer plus facilement aux développeurs. Nous avons également chez Microsoft une centaine d'architectes ayant pour unique mission d'établir un dialogue constructif avec les architectes de nos clients et partenaires."

Alexis Moussine-Pouchkine, consultant Java et Sun One, SUN, SUN-A. : "J2EE définit clairement les rôles des différents acteurs d'un projet : le développeur, l'assembleur, le "déployeur", l'administrateur, etc. Malgré ses atouts, cette séparation des responsabilités impose qu'un architecte orchestre l'ensemble. L'architecte est donc pour nous un homme clé, qui décuple le potentiel d'une technologie. C'est également le garant de la pérennité des applications car l'architecture est la composante la plus pérenne d'une application distribuée (plus que le code, les technologies, les produits...). Conscient des difficultés à maîtriser l'ensemble des technologies en même temps que tous les risques d'un projet nous avons mis à leur disposition des " Blueprints " sur java.sun.com/blue prints. Ces documents traitent des problématiques d'applications d'entreprise, de la bonne utilisation des Web Services, ou encore de la performance globale d'une architecture. Ils sont complétés par des listes de bonnes pratiques, des design patterns, des exemples de code, ainsi que des listes de questions/réponses"

Un exemple de parcours

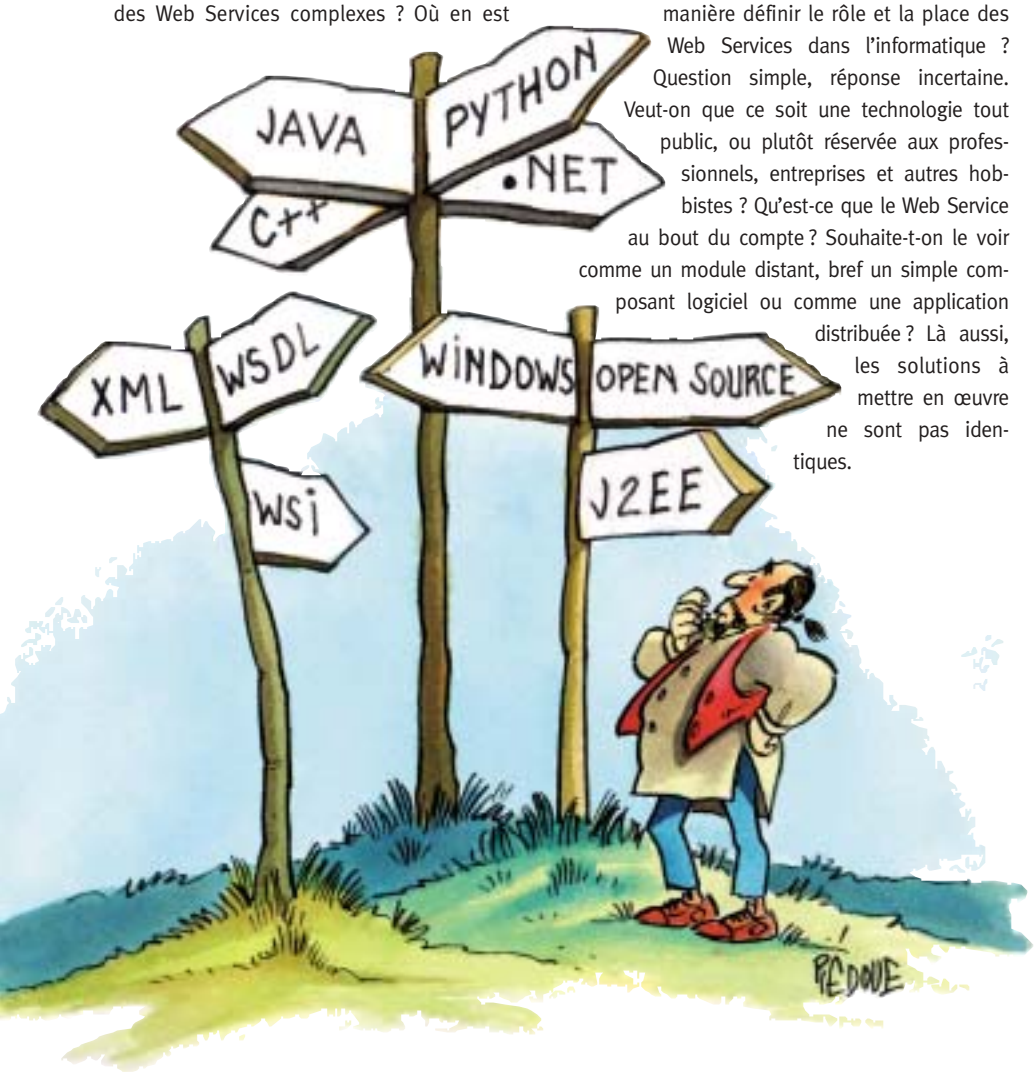
Raphaël MANSUY, Consultant Senior chez Business Design Consulting (BDC)

"Passionné d'informatique depuis mon adolescence, j'ai suivi des études d'informatique (licence-maîtrise-DESS bases de données et intelligence artificielle) après un DEUG de mathématiques. J'ai commencé ma carrière chez CERIUS informatique à Dijon (IBP Informatique Banque Populaire) en 1995 dans le département "Architecture Technique". C'était l'époque du client/serveur, des L4G, on ne parlait pas vraiment d'architecture multi-tiers, de serveur d'applications, et d'EAI. J'y ai conçu et développé un MOM (middleware orienté message) sous Windows NT, qui permettait aux postes Windows en agence, aux services applicatifs locaux de l'agence et au site central (mainframe IBM) de dialoguer en mode message. Nous avons également conçu un serveur d'applications sous NT, servant de cadre d'exécution à des services locaux en agence (standard de composants applicatifs enfichables, gestion des "threads", gestion de la synchronisation, file d'attente persistante, système de caches). Les contraintes étaient très fortes : lignes spécialisées de 9200 bits / secondes, 300 serveurs d'agence, 400 000 transactions par jour. J'ai rejoint ensuite la société Winwise (Microsoft Regional Director Paris) en 1997, en qualité de consultant expert, où j'ai exercé des missions variées d'expertise technique et de transfert de compétences pour de multiples sociétés comme Air Liquide, Valtech, Microsoft ou Thomson. Fin 1998 j'ai créé un cabinet de conseil en architecture spécialisé sur les technologies Microsoft au sein du Groupe Winwise. Business Design Consulting a ensuite été racheté par le groupe l'Air Liquide en 2000, dans le cadre de la création du pôle services Air Liquide. Le pôle services regroupe des sociétés spécialisées dans le domaine de la traçabilité, de la gestion de la "supply chain", l'optimisation, la télémesure, le télécontrôle. Le pôle voulait se doter d'un cabinet d'architectes pour assurer le succès des projets réalisés pour les clients du groupe. Business Design Consulting est aujourd'hui un cabinet de conseil de référence sur les architectures fondées sur les technologies Microsoft"

Web Services l'heure des choix

Il est temps maintenant de revenir sur cette technologie qui prend de plus en plus d'importance dans les stratégies d'entreprises et les offres des éditeurs.

Depuis un premier dossier (n°49 - Janvier 2003), deux mouvements sont à noter : l'arrivée d'un support natif dans Java et les projets Open Source. Cependant, certaines questions demeurent : où en est la sécurité ? Comment administrer des Web Services complexes ? Où en est l'interopérabilité ? Encore aujourd'hui, IBM et Microsoft demeurent les locomotives. Les raisons du retard de la sortie de J2EE 1.4 sont peu claires, même si certains voudraient y voir un problème avec le WS-I. Une question de fond demeure pour beaucoup. De quelle manière définir le rôle et la place des Web Services dans l'informatique ? Question simple, réponse incertaine. Veut-on que ce soit une technologie tout public, ou plutôt réservée aux professionnels, entreprises et autres hobbistes ? Qu'est-ce que le Web Service au bout du compte ? Souhaite-t-on le voir comme un module distant, bref un simple composant logiciel ou comme une application distribuée ? Là aussi, les solutions à mettre en œuvre ne sont pas identiques.



À l'heure où nous écrivons ces phrases, plusieurs problèmes sont à considérer. Faisons fi du pseudo problème de sécurité, qui n'est finalement qu'une bonne excuse pour ne pas expérimenter et/ou implémenter les Web Services. Non, le problème est ailleurs et en particulier dans toutes les notions d'administration des Web Services complexes. On ne peut pas dire que les protocoles et autres spécifications répondent aux attentes. Dans des environnements légers, le Web Service répond parfaitement aux besoins et ce, à moindre coût. Mais, essayez un peu de créer une intégration applicative en environnement lourd avec des Web Services. Je n'ose pas imaginer les difficultés.

Outre l'administration laborieuse, il va aussi se poser un problème de performances. Le Web Service n'est au bout du compte qu'un gros assemblage de couches que l'on superpose. Plus il y a de couches à traverser, plus les performances pâtissent du fait des multiples traitements. De plus, http n'est pas forcément le protocole de communication le plus performant. En interne, par exemple dans le cas d'un intranet, d'autres protocoles seraient plus appropriés, voire même en environnement inter entreprise unifié.

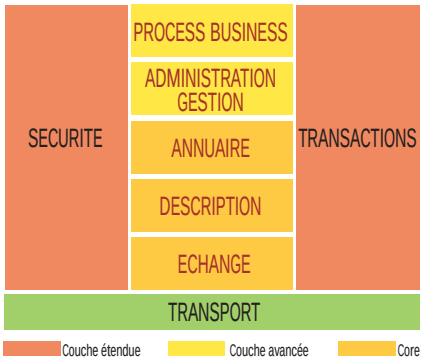
Si la technologie est bonne et représente une réelle avancée dans l'architecture applicative, il ne faut pas pour autant tomber dans le piège de la rhétorique des éditeurs. Il demeure, comme vous venez de le lire ci-dessus, des zones d'ombres que les consortiums et éditeurs doivent éclairer au plus vite.

François Tonic

Un labyrinthe technologique

Les Web Services intéressent les éditeurs et les projets indépendants. Dans le domaine Open Source / Logiciels Libres, il existe une offre convenable d'outils pour créer des Web Services. D'ailleurs, il n'est nullement besoin d'avoir un IDE .NET pour en faire. Que l'on soit en Java, Python ou C++, des produits sont maintenant disponibles. Mais l'environnement technique et technologique n'a jamais été aussi confus et diversifié.

ARCHITECTURE GLOBALE DES WEB SERVICES



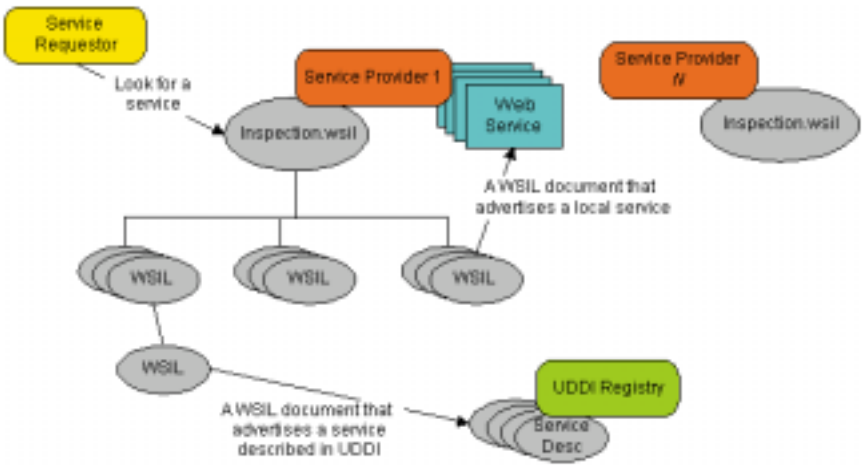
Les Web Services, on les voit à toutes les sauces et chez quasiment tous les éditeurs. Or, aujourd'hui encore, le Web Service a du mal à se positionner concrètement dans l'environnement applicatif de l'entreprise. Le plus gênant n'est pas là. Des problèmes existent sur les standards et les spécifications, la sécurité ou encore dans les choix des outils. Il est difficile de dresser un état de l'art du service web, tant la technologie change presque toutes les semaines.

Les standards sont-ils standard ?

Pour beaucoup, le principal problème des Web Services vient des standards, recommandations et spécifications. Dans le domaine des Web Services, il existe une telle profusion de standards (ou présentés comme tels), que l'on peut hésiter à utiliser telle ou telle solution. L'absence d'une réelle standardisation unique des Web Services (à tous les niveaux) empêche sans aucun doute une adoption plus massive de ceux-ci. Depuis quelques mois, on va jusqu'à s'interroger sur le rôle et la place du W3C dans les Web Services. Les diverses manœuvres autour de l'OASIS hypothèquent le rôle que tenait jusqu'à présent le W3C. OASIS est en passe de devenir la référence de standardisation. Le récent accord avec Rosettanet renforce OASIS dans les processus métiers. Cette alliance permettra aussi d'avoir une interopérabilité entre les protocoles des deux groupes. Preuve du poids important d'OASIS, de nombreux éditeurs ont demandé à cette organisation de s'occuper du BPEL4WS qui doit permettre une administration et gestion des Web Services complexes. Quoi qu'il en soit les délais d'adoption d'une spécification sont bien trop longs pour coller aux réalités du marché informatique. Dans ce contexte, il ne faut pas s'étonner de voir se multiplier les initiatives, quitte à passer en force et à imposer sa règle. D'ailleurs, on notera que plusieurs spécifications peuvent exister pour une seule et même fonction ! Dans ces conditions qui croire, qui suivre ? Faut-il suivre le consortium ayant le soutien des grands éditeurs ? Ou celui qui fait référence comme le W3C ? Le domaine de l'orchestration symbolise bien les difficultés d'harmoniser les différents camps. Quid de la fameuse universalité des Web Services ? Quid de la notion d'interopérabilité ? Il est encore trop tôt pour tirer des conclusions. Cependant, l'interopérabilité entre les différentes technologies risque de ne pas être une généralité, surtout entre les spécifications OASIS et W3C. Ce sont clairement deux camps rivaux, voir opposés. Bref, quand vous choisissez un outil de conception ou de gestion, vous faites un choix de facto dans l'un des camps et à vous de vérifier que votre chaîne Web Services reste parfaitement homogène et compatible. Espérons que les éditeurs comprendront l'intérêt d'une excellente interopérabilité. Avoir une multitude de protocole et de standards n'est pas en soi un obstacle si toutes les conditions d'une compatibilité optimale sont assurées. À l'heure actuelle ce n'est nullement le cas. Les retards du WS-I illustrent toutes les difficultés à imposer une interopérabilité. À ce jeu, les poids lourds de l'industrie informatique auront un rôle déterminant.

De toute manière, quand vous concevez et déployez des Web Services, de nombreux tests seront nécessaires : montée en charge, debug, interopérabilité, vérification SOAP, tests de sécurité, etc. Il faudra dans un avenir proche évoquer les coûts réels des Web Services.

Architecture de l'annuaire dans les services web



Les fondations des Web Services évoluent aussi...

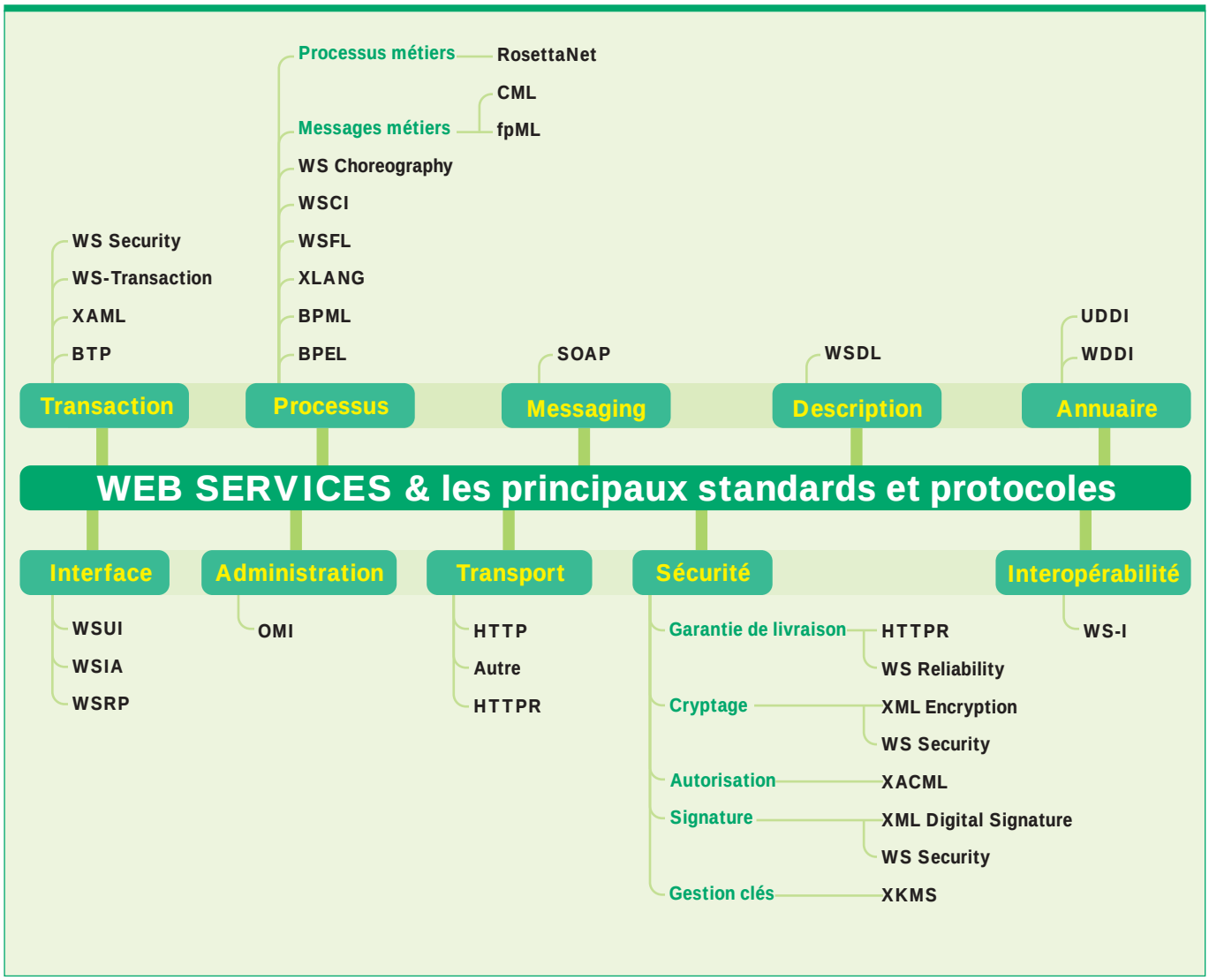
L'évolution des fondations des Web Services est une nécessité. Mais là aussi se pose un problème de suivi des spécifications par les éditeurs et les standards. La lenteur des travaux constitue aussi un handicap pour pouvoir faire évoluer plus rapidement les fondations et corriger les défauts.

1 - SOAP 1.2

Beaucoup attendent avec impatience la disponibilité finale de la version 1.2 de SOAP, qui règle les grandes faiblesses des versions précédentes. Plus de 600 problèmes de la 1.1 seront corrigés. Cette mouture doit aussi offrir de nouvelles fondations plus solides. Le W3C mise beaucoup sur cette évolution. Selon le W3C, il y aurait 9 raisons de passer à SOAP 1.2 :

- SOAP 1.2 fait tout ce que SOAP 1.1 fait, en mieux.
- SOAP 1.2 procure un modèle process clair
- Une meilleure interopérabilité : ce qui ne sera pas un luxe. Pas mal de problèmes avaient été constatés avec la 1.1.
- SOAP 1.2 est basé sur le XML Information Set. Cela signifie que la structure des messages SOAP est modifiée en adoptant l'idée d'infoset. Cela permet une plus grande souplesse et de meilleures performances (à vérifier dans la pratique).
- SOAP 1.2 procure un protocole indépendant et fournit un framework. W3C insiste sur le fait que SOAP 1.2 est un vrai protocole indépendant.
- Intégration d'un binding http pour accroître l'intégration au web.
- SOAP 1.2 fournit un modèle clair ET extensible.
- SOAP 1.2 implémente et supporte mieux les standards Web (XML Base, XML Schema 1.0, internationalisation ...)
- SOAP 1.2 devient la meilleure suite XML pour construire des applications distribuées via le Web. On pourra mettre en œuvre WSDL 1.2 (sortant début 2004).

Bref, pour le W3C, SOAP 1.2 est l'avenir et constituerait donc une meilleure base technique pour les Web Services. Il faut dire qu'il aura fallu attendre plus de deux ans et demi pour que cette nouvelle spécification soit finalisée ! Si en plus, la version 1.2 règle les problèmes d'interopérabilité, les responsables informatiques n'en seront que plus ravis...



SOAP 1.2 fait sauter la limitation à http / http/post.

2 – WSDL 1.2

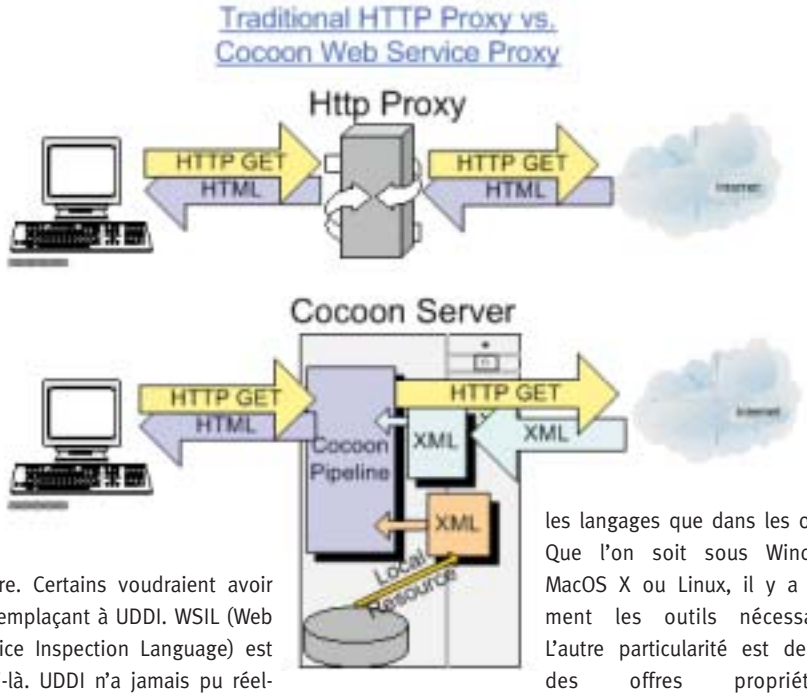
Tout comme la 1.2 de SOAP, la 1.2 de WSDL doit accroître la fiabilité de l'interopérabilité. Cette évolution apporte aussi un modèle mieux défini dans son langage. Elle s'appuie sur XML schéma et XML Information Set. Pour l'interopérabilité, le but du W3C est de supprimer les fonctions non interopérables de la version 1.1. Il s'agit aussi de suivre l'évolution du http et de SOAP. Comme les travaux sont toujours en cours sur cette spécification, il demeure quelques incertitudes sur les fonctions qui seront supprimées.

WSDL 1.2 est accompagnée de deux documents de travail : WSDL Bindings 1.2 et Usage Scenarios . WSDL Bindings 1.2 explique comment accéder à un fichier WSDL avec SOAP 1.2, HTTP 1.1 et MIME. Les scénarios d'usage listent l'ensemble des utilisations possibles de WSDL et notamment les différents types de modèles conversationnels utilisés : fire & forget, request -- response, etc.

3 – UDDI est-il l'annuaire du futur ?

Le référencement des Web Services demeure un point noir. Il faut pas mal de rigueur dans le travail de référencement. Dans un contexte inter entreprise ou purement interne, l'annuaire était-il indispensable ? Pas sûr. UDDI n'est surtout pas le seul candidat annoncé par l'annuaire. Certains voudraient avoir un remplaçant à UDDI. WSIL (Web Service Inspection Language) est celui-là. UDDI n'a jamais pu réellement remplir ses ambitions initiales. WSIL est une initiative d'IBM et Microsoft. Fondamentalement, WSIL peut être complémentaire à UDDI. Le WSIL est un simple fichier XML. Il facilite la découverte et la description des services web. L'approche de WSIL se veut simple : une approche document. L'approche de WSDI se veut aussi décentralisée. La description s'appuie directement sur WSDL. Les documents sont localisés en utilisant uniquement les standards et protocoles du Web. D'ailleurs le projet Apache Axis l'implémente. L'avantage du WSIL réside dans son utilisation des informations du WSIL et d'un simple fichier XML. Outre l'aspect totalement décentralisé de WSIL, il redonne le contrôle aux fournisseurs de services, ce qui n'était pas le cas avec UDDI. WSIL a des ambitions modestes par rapport à UDDI. On décrit son Web Service sur son site. Ainsi on crée un point d'entrée. À charge ensuite d'interroger un site pour obtenir les descriptions. WSIL définit bien entendu une série de règles pour structurer et présenter les informations.

Fonctionnement du projet open source Cocoon



De .NET à l'Open Source, de C# à PHP : un choix sans limite ou presque

Il faut faire une différence entre le développement et la plate-forme de déploiement des Web Services. Pour le côté développement pur, l'offre est désormais très large aussi bien dans les langages que dans les outils. Que l'on soit sous Windows, MacOS X ou Linux, il y a sûrement les outils nécessaires. L'autre particularité est de voir des offres propriétaires (WebSphere, WebLogic, Visual Studio.NET, JBuilder, etc.) et Open Source / Libre. Il est possible de réduire considérablement le budget alloué aux environnements de développement. Si vous êtes habitué à Python, vous pouvez mettre en œuvre Pywebsvcs. Il supporte les standards (SOAP, WSDL, UDDI).

Il existe des implémentations indépendantes des différents standards. Par exemple, pour SOAP, on peut utiliser SOAPpy ou Zolera SOAP Infrastructure. Pour WSDL, cela pourra être wstools. Si vous êtes plutôt Perl, SOAP ::Lite for Perl suffira amplement. Il supporte aussi bien SOAP 1.1 que 1.2. Cet outil a fait un gros effort sur les tests d'interopérabilité avec d'autres environnements. Il inclut aussi un parser XML ou encore XML RPC. Pour les amateurs de PHP, pas de souci non plus. Par exemple, pour faire du SOAP, il y a NuSOAP (nouvelle version réécrite de SOAPx4). Pour un IDE PHP, vous pouvez utiliser NuSphere. Même si vous restez en C#, il n'y a pas que Visual Studio .NET ou C# Builder (voir Programmez n°55). Côté C++, il n'y a pas non plus de difficultés. L'éditeur Rogue Wave est l'un des meilleurs spécialistes du domaine avec son outil LEIF. Il existe des outils Open Source tel que gSOAP. Pour les fans de Borland, il y a bien entendu C++ Builder. Le côté Java n'est pas non plus le parent pauvre, au contraire. Divers projets Open Source existent : Java Axis, WSDL2Java,

UDDI4Java... Les grands projets Open Source s'activent aussi autour des Web Services. Apache a publié son API Apache Web Services Invocation Framework. Cette API permet d'invoquer des services web décrits en WSDL. Avantage : on n'utilise rien que du standard, tout en se rendant neutre au niveau protocole de communication. Il fonctionne avec SOAP, EJB ou encore JMS. Aux frontières entre EAI / XML / Web Services, il y a des projets d'intégration de type Proteus et Babeldoc. Toujours du côté Apache, il existe Apache Cocoon. Cocoon est un environnement flexible pour le contenu, la logique et le style. Il permet d'avoir une solide base de serveur d'application XML. Cocoon est un framework pour construire et publier des applications web XML. Concernant tout particulièrement J2EE, il ne semble pas y avoir de problème. On peut parfaitement concevoir et déployer en environnement J2EE. Le seul réel souci sera de bien vérifier l'interopérabilité des services web. Les serveurs J2EE ont tendance à avoir quelques différences selon l'éditeur. Il est tout à fait possible de déployer des Web Services, via Apache Tomact, voire des serveurs d'application de type Jonas.

Et la sécurité ?

Où en est la sécurité autour des Web Services ? La situation depuis plusieurs mois a été notablement améliorée, même si les différents protocoles sont loin d'être finalisés et surtout implémentés. En usage interne, la situation n'a guère changé. Les risques ne sont pas plus grands qu'à l'accoutumée. Pour un extranet, il faut sécuriser les accès au réseau, par exemple, avec un bon VPN, comme on peut le faire entre réseaux et terminaux mobiles. Malgré tout, nativement, les Web Services ont toujours des carences. Plusieurs critères sont à retenir dans ce domaine : authentification, intégrité des données et messages, ou encore non-répudiation (= persistance de la session). Il faut que l'utilisation d'un service web et le service en lui-même soit dûment authentifié, pour lui donner l'autorisation d'ouvrir une session. Plusieurs travaux tournent autour de cela : SSO, SAML. L'intégrité des données et messages SOAP est un critère vital. Car, un intrus pourrait alors pénétrer sur le serveur, via le protocole SOAP. Spécifiquement sur SOAP, on peut mettre en place des firewalls SOAP (ex. : WestBridge, Reactivity, Xtradyne). Les éditeurs de sécurité informatique surfent sur une porteuse impressionnante tant que les standards, spécifications et protocoles ne seront pas figés pour de bon ! Un firewall seul ne suffit pas. Il faut aussi des technologies d'authentification forte, voire de signatures, pour garantir l'intégrité des données transportées. À cela, il faut bien entendu de puissants mécanismes de cryptage. Si vous pouvez utiliser d'autres protocoles de communication que http, allez-y, surtout s'ils sont mieux sécurisés. WS-Security annoncé dès 2001 n'est toujours pas entièrement disponible. Les discussions sont longues (on le voit bien avec WS-I). WS-Security propose 3 éléments : certification, intégrité et confidentialité. Cette spécification s'appuie sur SOAP. Tous les futurs produits Microsoft doivent l'inclure.

Attention cependant au prix des éditeurs concernant les outils de sécurité web services. La facture peut être impressionnante.

Orchestration : un obstacle de moins ?

Depuis l'annonce du BPEL4WS en 2002, ses promoteurs, BEA, Microsoft et IBM, semblaient patiner un peu pour l'imposer. Après avoir un temps pensé à le soumettre au W3C, c'est finalement OASIS qui a le droit de récupérer le BPEL4WS pour en faire un standard ! Cette annonce durant le printemps dernier, constitue une avancée importante dans l'orchestration des services web. Si pour certains, BPEL4WS est déjà un standard, est-ce LE standard d'orchestration ? Pas de réponse pour le moment. L'avantage de BPEL4WS est d'exploiter aussi bien WSFL que Xlang. L'un des objectifs de BPEL4WS est de fournir un moyen de définir un modèle et de synchroniser les transactions sous Web Services. À cela s'ajoutent WS-Coordination pour l'interaction entre différents web services et WS-Transaction pour la vérification de l'aboutissement des transactions (ou de l'échec). Au niveau de l'architecture, BPEL4WS se situe juste au-dessus de WSDL. Malheureusement, l'orchestration reste un terrain à risques avec une multitude d'offres : WSCI, BPML, XPDL ou encore BPSS. Pourquoi l'orchestration est si importante ? Tout simplement parce qu'il s'agit de définir un processus métier Web Services, capable de fonctionner en inter-entreprise, via les web services. En fait, on définit un modèle de scénario qui permet de piloter différents web services dans une architecture complexe. L'orchestration fournit donc le modèle de conception, de production et de maintenance des processus métiers.

Il faut posséder un standard reconnu par tous, implémenté par tous et surtout parfaitement interopérable entre web services de diverses origines. On en rêve encore.

■ François Tonic

GXA – WSE : le couple infernal de Microsoft

Microsoft a compris que l'un des freins aux Web Services est l'absence de tout package tout-en-un que l'on installe rapidement et qui permet de créer et de déployer facilement des services, tout en supportant les dernières versions des différents standards. Aujourd'hui, il faut bien avouer que l'on est souvent obligé de prendre des kits et implémentations ici ou là pour avoir un environnement complet. Microsoft a lancé deux projets : GXA et WSE. Le premier GXA (Global XML Web Services Architecture) est un ensemble de standards et protocoles. Pour Microsoft, GXA représente l'avenir du service web. GXA a l'avantage d'inclure une couche infrastructure. Souple et modulaire, GXA suit les évolutions de SOAP, UDDI, WS-I, etc. Mais, GXA n'est pas un environnement palpable. WSE (Web Services Enhancements) reprend les préceptes de GXA mais lui est palpable. Il propose un kit de fonctions avancées qui s'intègre directement à VS.NET. Il permet de bénéficier des dernières évolutions des standards Web Services sans attendre une mise à jour des couches .NET. WSE se compose principalement de 4 éléments : WS-Security, WS-Routing, WS-Attachments et DIME. WSE s'ajoute aux fonctions déjà implémentées dans .NET. Il ne les remplace pas.

L'avenir des Web Services est-il dans les concepts intégrés et modulaires et évolutifs afin de faciliter au maximum développement et déploiement ? Le débat est ouvert.

J2EE 1.4 en retard

Entretien avec Mark Hapner, ingénieur chez Sun Microsystems, en charge de la spécification J2EE et membre du conseil d'administration de la WS-I. Programmez ! a interrogé ce porte-parole de Sun sur l'avenir de J2EE.

Programmez ! : Comment expliquer que la spécification J2EE 1.4 n'ait pas encore été publiée ?

Mark Hapner : La spécification J2EE 1.4 est prête depuis six mois. Il n'y a aucun nouvel ajustement fonctionnel à y apporter. La principale raison de ce retard est imputable au besoin de synchroniser sa publication avec les travaux de la WS-I. J2EE 1.4 supporte déjà les API de services web. Mais, pour qu'ils puissent apporter une vraie valeur ajoutée, les services web ont besoin d'interopérabilité. Celle-ci dépend de la finalisation de la BP 1.0 (Basic Profil) par la WS-I.

Nous avions espéré que cela serait le cas pour ce printemps. Cette situation est frustrante pour tous les acteurs du marché J2EE car tout le monde est prêt à y aller.

« Le retard subi par J2EE 1.4, bride pour un temps l'évolution des fonctions des serveurs d'applications »

P ! : Quel est précisément le stade d'avancement de cette version 1.0 de la Basic Profil ?

M. H. : Une nouvelle version de la BP 1.0 vient d'être rendue publique. La plupart des modifications étaient d'ordre éditorial ou visaient à apporter des clarifications. Mais, le problème est que la BP 1.0 n'a pas encore été ratifiée par le conseil d'administration du WS-I.

Celui-ci devait procéder à cette ratification lors de sa dernière réunion, qui s'est tenue fin juin à Helsinki en Finlande.

La BP 1.0 devra ensuite faire l'objet d'une consultation publique et enfin être soumise à un nouveau vote. A cause du règlement interne de la WS-I, je ne suis pas autorisé à donner les détails de ce retard. Nous espérons qu'une ratification de la BP 1.0 pourra intervenir en août.

P ! : Quel est le niveau d'interopérabilité défini par la spécification Basic Profil 1.0 ?

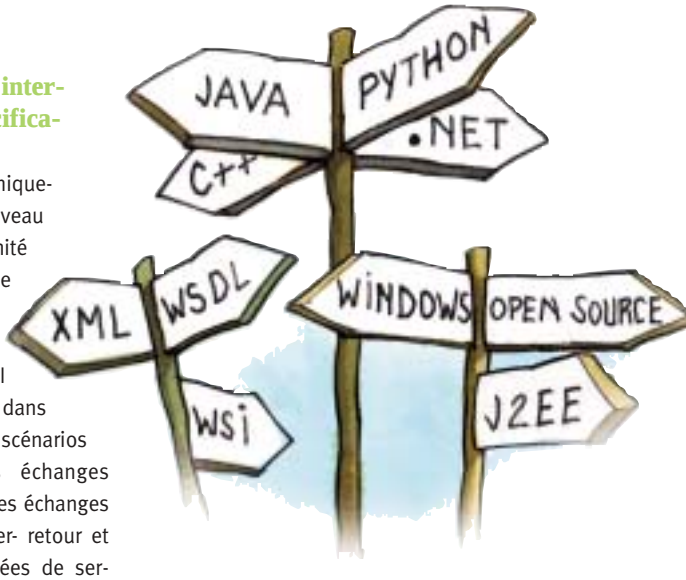
M. H. : L'interopérabilité porte uniquement sur les couches basses au niveau du réseau. Ce niveau de conformité n'est pas très strict. Il s'applique principalement à l'extrémité des connexions lors d'un échange de services web, via l'API JAX-RPC 1.1 définie dans J2EE. Trois différents scénarios sont décrits : Les échanges simples synchrones, les échanges asynchrones sans aller- retour et les invocations différées de services web. Ce n'est là qu'un début dans l'évolution des services web. La WS-I ne se focalise pas sur la validation technique au niveau des outils de développements, ou au niveau des applications.

Les échanges, par le biais des services web, ne sont qu'une possibilité parmi d'autres, d'intégration avec un couplage lâche.

P ! : Quel est l'impact sur les éditeurs de serveurs d'applications, du retard de J2EE 1.4 ?

M. H. : Aujourd'hui, les éditeurs de serveurs J2EE ainsi que les développeurs, peuvent toujours implémenter les spécifications de services web. D'ailleurs, certains d'entres eux ont déjà mis en œuvre des éléments contenus dans la BP 1.0. Mais il ne s'agit là que de versions préliminaires et, à fortiori, propriétaires.

Il faut faire attention de ne pas pratiquer une fuite en avant dans des implémentations qui risquerait de fragmenter le marché. Les éditeurs de serveurs J2EE doivent attendre la publication de la spécification J2EE 1.4 pour bénéficier de la validation des tests de compatibilité avec ces standards. Le risque existe, que le retard subi par J2EE 1.4, bride pour un temps l'évolution des fonctions des serveurs d'applications.



P ! : Quelles sont les futures évolutions à prévoir, pour le support des services web dans J2EE ?

M. H. : En supportant déjà l'encodage de message SOAP, J2EE 1.4 est en avance sur la BP 1.0. Je rappelle aussi que JAX-RPC était déjà fournie en option dans J2EE 1.3. L'éparpillement des initiatives de standards de services web constitue un vrai problème. Bon nombre se résument à de simples papiers blancs et ont peu de chances d'aboutir. Dans la palette des fonctions J2EE, les services web ne sont qu'un des choix possible d'intégration.

J2EE Connecteur Architecture (JCA) prend en charge une intégration avec couplage fort, JMS (Java Messaging Service) gère l'intégration lâche, par messagerie asynchrone et RMI-IIOP fédère les modèles d'objets distribués. J2EE 1.4 jette les fondements d'une convergence entre ces trois approches.

■ Propos recueillis par **Samuel Cadogan**

J2EE prêt à déferler sur les Web Services

Avec Java Web Services Developer Pack (JWSDP), Sun livre une première feuille de route de migration vers les services web, mais confie à IBM le pilotage d'une convergence avec J2EE.

Nonobstant le manque d'interopérabilité, les développeurs J2EE ne sont pas privés de services web. On en veut pour simple preuve que la première version de JAX-RPC, qui est la clé de voûte pour des services web dans J2EE, a été finalisée depuis un an par la Java Community Process. Plusieurs serveurs J2EE en livrent déjà une première implémentation. Sun inclut dans la version 2.1 de JWSDP, une version 1.1 de JAX-PRC pour gérer, entre autres, l'asynchronisme. Destiné à être le socle de son futur RAD (Rave), JWSDP constitue un point d'ancrage aux éditeurs J2EE non seulement pour le développement des services web mais aussi pour les applications Web.

Les échanges SOAP circonscrits à HTTP.

Pour ce qui concerne les services web, tout l'enjeu consiste à aligner JAX-RPC sur toute une famille d'API complémentaires. Aujourd'hui, un premier niveau d'échange est déjà possible grâce SAAJ (SOAP with Attachments API for Java) Cette API qui permet au développeur J2EE de générer des messages conformes à SOAP, fonctionne uniquement sur HTTP. La prise en compte de SMTP est esquissée par la version 2.0 de JAX-RPC. Pour cette raison, JWSDP n'inclut pas l'API JAX-M. Quoiqu'il en soit, JAX-RPC demeure délibérément agnostique dans son support de broker de messageries autres que SOAP. En attendant, cette API converge avec JAX-P JAX-B, utilisés respectivement pour le parsing et l'importation de fichiers XML dans Java. Des opérations gourmandes en performances. Pour y pallier, Sun lance l'initiative FAST-WS.

De son côté, Bea Systems s'est fendue d'une implémentation de JAX-B baptisée XML Beans. Objectif : donner un accès à un document XML dans un environnement Java. Toute l'astuce des XML Beans repose sur la séparation de la présentation des données XML et de leur implémentation en Java. Evitant ainsi les opérations lourdes de sérialisation. De plus, les XML Beans visent une compatibilité intégrale avec les schémas XML, Les services web profitent du conteneur J2EE Aujourd'hui, le déploiement de services web s'appuie sur le conteneur J2EE. Ce dernier bénéficie d'une gestion éprouvée des transactions, asynchronisme inclus.

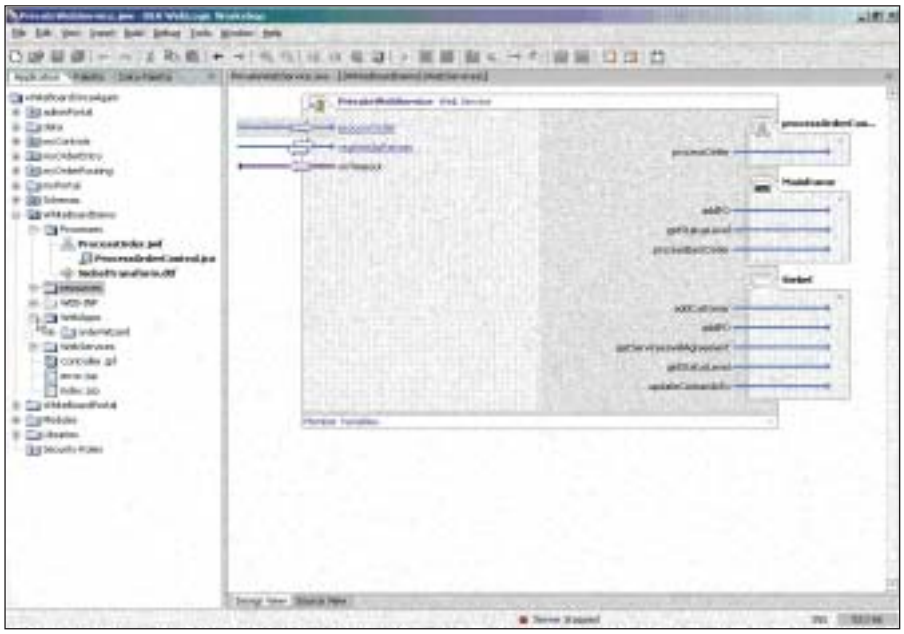
Pour mettre le pied à l'étrier, JWSDP livre le conteneur Tomcat. Mais, l'intégration de J2EE aux standards transactionnels dédiés aux services web n'est pas à l'ordre du jour. "Les standards WS-Transactions et WS-Coordination existent uniquement sous la forme de spécifications de fournisseurs et ne font l'objet d'aucun processus de standardisation" explique John Kiger, directeur services web chez Bea. Ce constat se vérifie pour la sécurité. Même si la JWSDP inclut une déclinaison XML de WS-Security, ainsi que l'API JAX-R pour l'accès aux annuaires de type UDDI. De la mise en place des briques de la JWSDP, dépend la simplification des développements Java.

Cette évolution s'incarne par deux initiatives d'intégration de Java aux meta données. Initiée par Bea, la première d'entre elle définit des attributs pour représenter les classes, les interfaces, les champs et les méthodes Java. La seconde s'appuie sur ces mêmes attributs pour réaliser un mapping entre Java et WSDL. "Nous travaillons activement à la mise en place d'un framework de MetaData au sein de NetBeans, mais celui-ci ne sera disponible que dans la JDK 1.5" tempère toutefois Alexis Moussine-Pouchkine, évangélisateur Java chez Sun Microsystems.

Au sein de la Java Community Process, c'est à IBM qu'échoit la tâche de mettre en musique ces évolutions. En effet, la spécification JSR 109 qu'il dirige, a comme mission de définir un modèle fédérateur de programmation et d'exécution de services web au sein de J2EE. Prévue pour être disponible d'ici la fin de cette année, elle devra s'articuler autour de JAX-RPC. Toutefois, pour réaliser cette synthèse, IBM sera confronté à des arbitrages délicats. Notamment concernant le choix de l'implémentation de référence JAX-B ou encore le modalités de la convergence entre Java et les méta données.

■ Samuel Cadogan

Le standards JWS (Java Web Service) proposé à la JCP par Bea permet aux développeurs de bénéficier de services web portables, tout en s'affranchissant du maniement complexe des API J2EE.



C++, XML et Web Services se jouent des scores !

Quel est le point commun entre Rogue Wave, des panneaux de résultats sportifs, les Web Services, XML et le C++ ? Le point commun est l'environnement Score Live de la société Integral Net.

Beaucoup considèrent le C++ comme un langage vieux et un ancêtre. Certes, c'est un dinosaure mais toujours redoutable. Le C++ a su évoluer vers les nouvelles technologies. Aujourd'hui, le C++ peut parfaitement servir aux applications Web et pour concevoir des Web Services ! L'éditeur Rogue Wave est l'un des spécialistes mondiaux dans ce domaine. Quand Integral Net a conçu son projet Score Live, il lui fallait une architecture suffisamment rapide pour répondre aux impératifs de temps. Le premier sport visé est le basket. Sur ce jeu, le score et le jeu évoluent très rapidement.

Architecture & fonctionnement

L'architecture de l'ensemble est plutôt simple. Elle se décompose en trois parties : la partie capture des données, la partie serveur et enfin la partie cliente. Sur le terrain sportif, un ordinateur est connecté au tableau d'affichage afin de récupérer les données en temps réel. Une application récupère les données du panneau de résultat. Ces données sont traitées sommairement, puis envoyées au serveur sous forme des paquets XML, dans un souci de performance et de rapidité. Les paquets XML sont réceptionnés par les serveurs d'Integral Net, et les Web Services XML utilisés au niveau des serveurs. Les services web sont là pour déterminer la localisation de l'internaute connecté et déterminer la salle sportive qu'il souhaite. Le service indique à l'application cliente le bon serveur. Le service web a aussi un rôle de répartition de charge entre les différents serveurs. Côté client, le choix s'est porté sur Flash. Flash garantit le même affichage quelle que soit la plate-forme. De plus, Flash peut utiliser XML. Les serveurs envoient seulement des flux XML légers que le client Flash traite. Grâce à cet ensemble technologique, Score Live fournit du push de données et non du polling http/SOAP. On gagne ainsi en performance, en réactivité, tout en réduisant au maximum l'utilisation de la bande passante.

ScoreLive : schéma de fonctionnement

Pourquoi C++ ?

Le choix du C++ est du au problème de performances. Des tests avaient été réalisés sous Java, mais les résultats n'étaient pas concluants, surtout côté performances. Score Live travaille dans un contexte de temps réel ou presque. L'implémentation de XML et des Web Services n'a guère été un souci. " Avec les Web Services et les outils utilisés, on a pu se focaliser sur l'aspect métier " nous a confié Gaël Giraud d'Integral Net. De plus, aucun problème de montée en charge n'a été décelé. Le C++ permet d'obtenir de bonnes performances. La sécurité n'a pas été non plus un gros problème. Le décalage entre l'affichage des scores sur les tableaux du terrain et l'affichage de ceux-ci sur le site web est d'environ 0,5 seconde. " Le C++ amène d'abord et surtout la performance, sur ce point il est imbattable. Les performances des environnements .NET et plus encore J2EE sont bien loin des performances des composants RogueWave. " précise Gaël Giraud. La bonne performance de C++ évite aussi de déployer un nombre supplémentaire de serveurs. Et si l'application sature les ressources serveurs, il y a toujours la possibilité de mettre en œuvre l'outil ATS, qui permet d'optimiser l'allocation mémoire.

■ **François Tonic**

Web Services en C++

Integral Net a utilisé l'outil LEIF de Rogue Wave. LEIF (Lightweight Enterprise Integration Framework) doit faciliter l'intégration des applications C++ dans le nouveau système d'information de l'entreprise, comme par exemple .NET ou J2EE. En quelque sorte, LEIF permet d'interopérer C++ avec du .NET, J2EE, service web ! Cela évite de tout réécrire. LEIF permet aussi d'intégrer XML dans des applications C++. On peut alors ainsi créer des Web Services en C++.

Panorama des outils Web Services

Il y a deux ans, les pionniers des services web devaient se résigner à développer leurs projets avec des outils mal adaptés et à assurer la sécurité de leurs applications SOAP avec les moyens du bord. Cette situation a largement changé depuis et on compte désormais plus de 300 outils dédiés aux services web ! Revue des outils qui vous aideront à transformer la promesse des services web en une réalité !

David Thévenon

1 - Outils de développement

Java

Editeur	Outil	Commentaires
BEA	**Weblogic Workshop 8.1** www.bea.com/products/weblogic/workshop	Développement, assemblage et déploiement graphique. Générationdynamique des proxy SOAP et des contrats WSDL. Support de WS-Security.
Borland	**JBuilder 9** www.borland.fr/jbuilder	Génération du proxy SOAP à partir du contrat WSDL et exposition de classe java ou d'EJB existants, via un proxy SOAP et un contrat WSDL.
Cape Clear	**Cape Studio 4.0** www.capeclear.com/products/studio	IDE spécialisé. Création automatique de proxy SOAP et de contrat WSDL. Annuaire UDDI, création automatique de requête de test, interface de mapping.
IBM	**Websphere Studio 5 Application Developper et WSDK 5.0.1** www-106.ibm.com/developerworks/webservices/wsdk	Création et de test. Compatible WS-I Basic Profile 1.0.
NetAxiom	**Hyperservice Studio** www.nextaxiom.com/NextAxiom_Products_HyperserviceStudio.html	Assemblage graphique. Mapping et routage de données, debugger intégré. Gestion de l'état du processus,
Oracle	**JDeveloper 9.0.3** www.oracle.com/ip/develop/ids/index.html?java.html	Développement de services web et exposition de procédures stockées et de files JMS. Browser UDDI, outil de test debug local et remote, Développement visuel basé sur UML.
Sun	**Sun ONE Web Services Platform Developer Edition et WSDP 1.2** java.sun.com/webservices/webservicespack.html	Outil spécialisé. supporte toutes les API Java liées à XML et SOAP ainsi que JSF 1.0. Intègre Ant Build Tool 1.5.2 et Apache Tomcat v5.
Systinet	**Systinet WASP Developer** www.systinet.com/products/wasp_developer/overview	Plugins pour les principaux IDE Java. Génération automatique de proxy Java à partir de contrat WSDL, annuaire UDDI, déploiement, debug. Gratuit pour un usage non commercial.

.NET

Editeur	Outil	Commentaires
Borland	**C# Builder 1.0** www.borland.com/csharpbuilder/professional	Génère un squelette de classe, le proxy SOAP, et vice versa à partir d'un contrat WSDL. Génération automatique (AOP).
Microsoft	**Visual Studio .NET 2003 et Web Services Enhancements for VS.NET** www.microsoft.com/France/vstudio	Pas d'assemblages dynamiques.

C / C++

Editeur	Outil	Commentaires
Borland	**C++Builder 6** www.borland.com/cbuilder/enterprise	BizSnap permet de créer rapidement des services web en C++.
RogueWave	**Lightweight Enterprise Integration Framework** www.roguewave.com/products/leif/features.cfm	Créer des proxy C++ dynamiquement à partir d'un contrat WSDL ou d'un schéma XML.
Open source Robert van Engelen	**gSOAP 2.2.3** www.cs.fsu.edu/~engelen/soap.html	Librairies C/C++ et compilateur. Création de services SOAP en C/C++. Mapping des types, générateur à la volée de contrat WSDL et de proxy SOAP.
Open source David Crowley	**EasySoap++ 0.6** easysoap.sourceforge.net	Librairie SOAP pour C++. Pas encore de support de WSDL.

PHP / Perl

Editeur	Outil	Commentaires
Active State	**Komodo 2.3** www.activestate.com/Products/Komodo	Génère les proxy SOAP (en Perl) à partir d'un contrat WSDL. Documentation automatique, auto complétion.
Open source	**SOAP::Lite for Perl** www.soaplite.com	Modules Perl pour le support de SOAP, WSDL, UDDI, MOM (Jaber ou MQSeries)
NuSphere	**PHPEd 3.2** www.nusphere.com/products	Génère automatiquement un proxy SOAP à partir d'un contrat WSDL.
PEAR	**PEAR::SOAP** pear.php.net/package-info.php?pacid=87	Package PEAR. Génère dynamiquement les proxy SOAP à partir d'un contrat WSDL et vice versa.

2) Outils de test

Conformité et interopérabilité

Editeur	Outil	Commentaires
StrikeIron	**StrikeIron Web Services Analyzer** www.strikeiron.com/htmls/products.html	Analyse de contrat WSDL.
soapclient.com	**Generic SOAP Client** www.soapclient.com/soaptest.html	Permet d'accéder à un service web depuis un simple navigateur.
XMethods	**WSDL Analyzer** xmethods.net/ve2/Tools.po	Analyse de contrat WSDL et test de validité.
soapclient.com	**SOAP Attachment Test** www.soapclient.com/SOAPAttachment.html	Teste la conformité d'un service web avec les protocoles (SOAP with Attachments - MIME) et DIME (Direct Internet Message Encapsulation).
Soapware	**SOAP 1.1 Validator** validator.soapware.org/	Test d'interopérabilité SOAP 1.1

Montée en charge

Editeur	Outil	Commentaires
MindReef	**SOAPscope Personal 1.0** www.mindreef.coml	Trace de messages SOAP.
Parasoft	**SOAPtest** www.parasoft.com/jsp/products/home.jsp?product=SOAP&itemId=108	Crée un jeu de clients SOAP pour tester l'interopérabilité et la montée en charge. L'url du contrat WSDL suffit
PushToTest	**TestMaker 4.0** www.pushtotest.com/ptt/downloads.html	Enregistre des scénarios de test et les rejoue.
Red-Gate	**Ants Load** www.red-gate.com	Supporte le test de SOAP.

3) Socle d'exécution

Java

Editeur	Outil	Commentaires
Apache	**Axis 1.1** ws.apache.org	Couche SOAP et XML-RPC
BEA	**WebLogic 8.1** www.bea.com/products/weblogic/server	Conforme aux dernières spécifications.
Borland	**Borland Enterprise Server Web Edition 5.1** www.borland.com/besweb	Clustering, load balancing et tolerance aux pannes. requêtes SOAP routées vers un service web, un service CORBA, un EJB ou un MDB. Contient un runtime Delphi.
Cape Clear	**Cape Clear Server 4.0** www.capeclear.com/products/server	Expose des composants COM, Java ou Cobol. Couche de sécurité et d'administration intégrée. Se connecte aux principaux MOM du marché.
IBM	**WebSphere Application Server 5.0** www-3.ibm.com/software/webservers/appserv	Conforme aux dernières spécifications.
Oracle	**Oracle 9iAS J2EE** www.oracle.com/ip/deploy/ias/gs/index.html?j2ee.html	Conforme aux dernières spécifications.
Sun	**SunOne App Server 7 Platform Edition** wwws.sun.com/software/products/appsrvr/home_appsrvr.html	Gratuit. conforme aux dernières spécifications.
Systinet	**WASP Server for Java 4.6** www.systinet.com/products/wasp_jserver/overview	Load balancing & failover. Support de messagerie asynchrone, gestion d'état indépendante du transport, se connecte à la plupart des MOM JMS, outil d'administration et de supervision. Intégration de toutes les normes liées à la sécurité.
The Mind Electric	**Glue 4.0** www.themindelectric.com/glue	Utilisable en embarqué. Outil d'administration et de supervision. Exposition automatique d'EJB et de classes Java lors du déploiement.

DNA et .NET

Editeur	Outil	Commentaires
Microsoft	**.NET** www.microsoft.com/net	Génération à l'exécution des proxies et contrats WSDL.
WhiteMesa	**SOAP server 3.1** www.whitemesa.com/wmsoapsvc_about.htm	Daemon http et processeurs SOAP. couche d'authentification. Supporte WS-Routing.
Lucin	**SAL Server Professionnal** www.lucin.com/lu004/soap.htm	Expose automatiquement (proxy SOAP et contrats WSDL) des composants COM au déploiement. Inscrit le service dans le SAL Network. Parallélise le traitement des requêtes SOAP entrantes. S'appuie sur MTS.

C++

Editeur	Outil	Sécurité
Systinet	**WASP Server for C++ 4.5.2** www.systinet.com/products/wasp_cserver/overview	Load balancing & failover. outil d'administration et de supervision. Intégration de toutes les normes liées à la sécurité.
SQLData	**SOAP Server 3.0** www.sqldata.com/Soap.htm	Couche d'authentification, de signature et de cryptage.

4) Outils d'assemblage et d'orchestration de services web – Design Time

Java

Editeur	Outil	Commentaires
Intalio	**n3 Designer** www.intalio.com/products/designer	Définition des scénarios via BPML et WSCI. appels synchrones et asynchrones dans un même scénario.
Collaxa	**BPEL4WS Wizard** www.collaxa.com/product.welcome.html	Repose sur BPEL4WS. Environnement de création des scénarios complètement graphique. Nombreux assistants.
Cypress Logic	**ObjectView Professional Edition 3.0 (for Axis)** www.cypresslogic.com/professional.html	Assemblage graphique. Transformation et mapping. failover, gestion des redondance, parallélisation, cache, déploiement, etc
Vergil Technology	**Orchestration Suite (Composer)** www.vergiltech.com/vorchestration.htm	Interface graphique de modélisation. Assemblage basé sur de règles métier. Outil de mapping intégré.

.NET

Cypress Logic	**ObjectView Professionnal Edition 3.0 (for .NET)** www.cypresslogic.com/professional.html	Assemblage graphique. Transformation et mapping. failover, gestion des redondance, parallélisation, cache, déploiement, etc.
Microsoft	**Biztalk Orchestration Designer 2003** www.microsoft.com/biztalk	Modélisation graphique. XLang (Microsoft). BPEL4WS prévue dans la version 2004.

5) Outils d'assemblage et d'orchestration de services web – Runtime

Java

Editeur	Outil	Commentaires
Choreology	**Cohesions 1.0** www.choreology.com	Compensation, gestion des exceptions, services synchrones et asynchrones dans un même scénario. Basé sur JRE 1.4. Supporte WS-transaction, WS-coordination et BTP
Collaxa	**BPEL Orchestration Server** www.collaxa.com/product.welcome.html	Compensation, gestion des exceptions, services synchrones et asynchrones dans un même scénario. WS-Transaction, WS-Coordination, BTP, JMS, MQSeries, Tibco
Intalio	**n3 Server** www.intalio.com/products/server	orchestration via BTP. Assure l'intégrité des transactions, la coordination et la sécurité des scénarios à l'exécution.
PolarLake	**XML Circuits** www.polarlake.com/products/polarlake/xmlcircuits.shtml	Moteur d'orchestration (XML Circuit), outil de mapping et l'offre Database Integrator du même éditeur.
NextAxiom	**NextAxiom Server** www.nextaxiom.com/NextAxiom_Products_NextAxiomServer.html	Routage et parallélisation de l'exécution des scénarios, couche de sécurité intégrée ; load balancing et cache.

.NET

Cypress Logic	**ObjectView Professional Edition** www.cypresslogic.com/professional.html	Assemblage graphique. Support de BPEL4WS, WSFL, etc. quand les standards seront stabilisés.
Kinzan	**Adaptative Web Services Suite** www.kinzan.com/products.html	Moteur d'orchestration et outil de mapping associés à des connecteurs SOAP natifs.
Microsoft	**Biztalk Orchestration Server 2003** www.microsoft.com/biztalk	Support des transactions longues, parallélisation des scénarios, gestion de l'état des processus, etc.
Open Storm	**ChoreoServer** www.openstorm.com	Support de BPEL4WS, WS-Transaction, WS-Coordination, WS-Basic Profile. Simple moteur d'orchestration. Pas de couche sécurité.

6) Outils de déploiement, de surveillance et d'administration

Editeur	Outil	Commentaires
Actional	**LookingGlass** www.actional.com/products/web_services/looking_glass	Gestion de la qualité des services en fonction de SLA. Versionning, BAM, SSO (via SAML), load balancing, stats.
Lucin	**Web Services Watch** www.salcentral.com/watchx/run.asp	Des agents vérifient, à intervalle régulier, que les services sont bien en activité. Solution ASP.
NetAxiom	**Hyperservice Manager** www.nextaxiom.com/NextAxiom_Products_HyperserviceManager.html	Gestion intégrée des utilisateurs et groupes d'utilisateurs. Contrôle d'accès. Gestion de priorité, administration distante.
Westglobal	**mScape 2.0** www.westglobal.com/products	Noyau enrichi par des modules optionnels : enregistrement des performances de chaque service, plate-forme de paiement, gestion des priorités et des versions en fonction de SLA, routage dynamique, BAM, statistiques, etc.

7) La sécurité des services web - Firewall SOAP

Editeur	Outil	Technologie supportée
Quadrasis et Xtradyne	**SOAP Content Inspector** www.quadrasis.com/solutions/products/easi_product_packages/easi_soap.htm	Authentification et validation des messages SOAP.
Reactivity	**Reactivity Firewall (appliance + console)** www.reactivity.com/product/features.html	Authentification, autorisation, confidentialité. console d'administration
RSA	**RSA BSAFE Secure-WS** www.rsasecurity.com/products/bsafe/securews.html	Sécurise les messages SOAP. Cryptage, décryptage, signature et vérification des entêtes SOAP.
Vordel	**VordelSecure 2.0** www.vordel.com/products	Authentification, autorisation, signature, et validation des flux XML.
Westbridge	**XML Message Server** www.westbridgetech.com/products.html	Authentification, autorisation, cryptage, et validation des flux XML. Intègre un moteur de règles pré paramétré pour bloquer les attaques de type flooding ou pillage de données.

Les Web Services : le long chemin vers la maturité…

On trouve plusieurs définitions des Web Services. La plus courante et la plus simple est la suivante : "un Web service est une fonctionnalité ou une collection de fonctionnalités issues d'une application et publiées sur Internet à l'aide du langage XML".

Une Entreprise, confrontée à une problématique d'intégration de systèmes à l'intérieur de son système d'information ou d'applications de partenaires, doit absolument considérer les Web Services dans sa réflexion, en plus des techniques d'intégration plus classiques (batch, fichier CSV, middlewares orientés message, EAI). Le choix d'une technologie d'intégration demande une réflexion approfondie sur les capacités des middlewares mis en jeu. Le middleware opérationnel dans les Web Services est le duo de standards SOAP/WSDL. Le choix d'une implémentation SOAP/WSDL doit ensuite être réalisé. Les principales alternatives sont les offres Microsoft .NET, Java (regroupant des offres commerciales ou Open Source).

Le couple SOAP/WSDL

SOAP (Simple Object Access Protocol) est une grammaire XML représentant un protocole d'invocation de services supportant deux modèles de communication :

- L'appel de procédures à distance sans Session (RPC)
- L'envoi de message asynchrone

WSDL (Web Service Description Langage) est une grammaire XML représentant la documentation technique d'un Web Service (SOAP ou autre). Un fichier WSDL relatif à un Web Service contient des informations concernant, entre autres, le nombre et la nature de ses paramètres, ses messages d'erreurs renvoyés, sa localisation, etc. De nombreuses implémentations SOAP/WSDL existent sur le marché. Elles permettent de faire communiquer des systèmes hétérogènes (Microsoft.NET, Perl, J2EE, SmallTalk, etc) de manière assez simple.

Les lacunes de SOAP/WSDL

Les Entreprises attendent beaucoup des Web Services pour commencer à concrétiser leur stratégie Ebusiness. L'enveloppe technique des Web services (SOAP/WSDL) est aujourd'hui rudimentaire pour adresser efficacement l'interopérabilité des systèmes d'information. Certaines problématiques essentielles comme la sécurité ou la livraison asynchrone de message ne sont pas adressées de manière satisfaisante. La mise en œuvre de la sécurité avec les Web Services se fait aujourd'hui, avec un coût raisonnable, à l'aide du protocole HTTPs. Cette solution ne couvre qu'une partie de la sécurité relative à un Web service. On ne peut pas blâmer HTTPs à ce sujet car il n'a pas été conçu à l'origine à cet effet. On l'utilise car il est répandu et stable. Il répondra néanmoins aux problématiques de sécurité simples. Le deuxième problème concerne la livraison asynchrone de message. La notion de découplage, si chère aux Web services, est contradictoire avec la nature RPC du protocole SOAP. SOAP propose bien l'envoi de message asynchrone mais ne garantit pas la livraison. Cet aspect est souvent un frein pour la conception d'architecture de Web Services complexes. Des travaux sont cependant en cours et sont très prometteurs, comme WS-Reliability.

« La récente publication de la recommandation W3C SOAP est encourageante »

Les standards SOAP/WSDL ne représentent ainsi que le haut de l'iceberg de la pile des protocoles des Web Services. La standardisation avance lentement mais sûrement, et la publication récente de la recommandation W3C SOAP est encourageante.

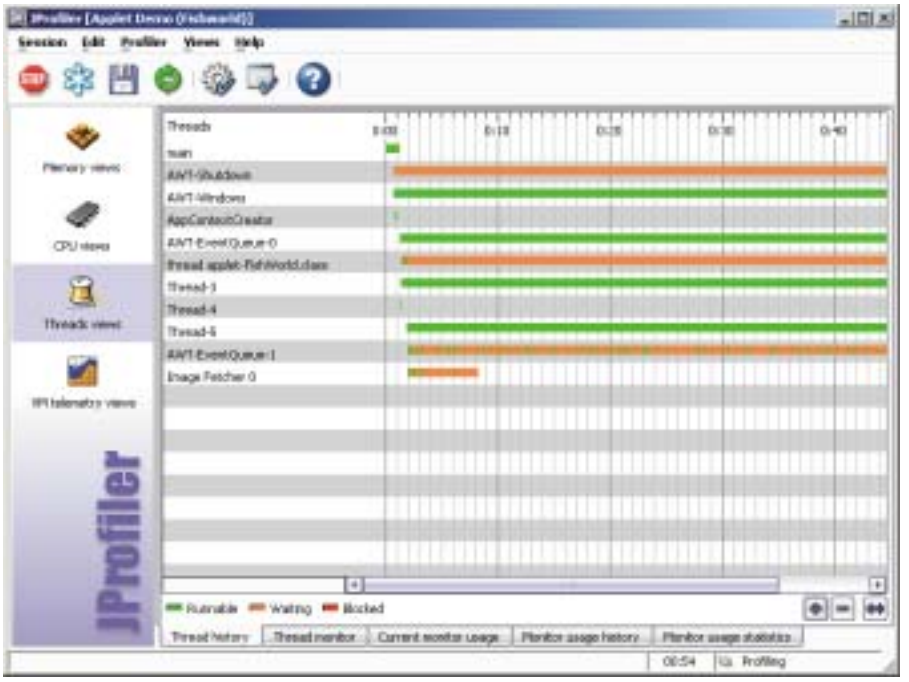
Le choix d'une implémentation Web Services

La mise en œuvre d'une implémentation SOAP/WSDL demande ensuite le choix d'une implémentation. Si une entreprise veut étendre un existant avec un minimum d'impact sur son système informatique, il lui faudra choisir l'implémentation SOAP/WSDL correspondant à ses technologies d'origine. Il existe ainsi des implémentations SOAP/WSDL pour Smalltak, PowerBuilder, Perl , Cobol pour CICS, etc. Pour une entreprise ayant déjà fait un choix d'architecture plus " moderne ", comme J2EE ou .NET ou PHP : le choix est quasiment déjà fait car ces plates-formes intègrent les Web Services nativement. Notons toutefois une subtilité pour Java : Une entreprise ayant déjà fait un choix de serveur J2EE commercial, par exemple BEA Weblogic, pourra utiliser l'implémentation SOAP du même éditeur, ou d'un éditeur externe, comme TheMindElectric Glue (implémentation SOAP/WSDL commerciale) ou Apache AXIS (implémentation SOAP/WSDL libre). La plate-forme PHP propose le support des Web services sur un mode libre. D'une manière générale les implémentations Web Services commerciales sont fonctionnellement plus riches que les implémentations libres. Pour une entreprise désirant faire un choix de serveur d'applications (notamment entre J2EE et .NET) : le critère de support des standards SOAP/WSDL n'est pas déterminant. En revanche, pour une vision d'intégration complexe à plus long terme, la plate-forme J2EE est plus intéressante, car supportée par les spécialistes de l'intégration, comme SeeBeyond ou WebMethods, qui ont acquis un excellent savoir faire dans ce domaine.

■ **Nicolas Farges** (Groupe SQLi)

Dessine-moi un thread…

On vous parle tous les mois dans Programmez de programmation et de technologies. Pourquoi ne pas revenir sur une notion fondamentale du développement moderne : le multithreading. Grâce à la possibilité de multiplier les threads, l'application peut acquérir une souplesse d'utilisation impressionnante. Encore faut-il maîtriser les concepts et la programmation des threads. Et ça, c'est parfois un cauchemar ! Petit cours.



De temps en temps, le développeur se demandera pourquoi se prendre la tête avec la programmation multithread dans son application. Le réflexe multithreading n'est pas encore systématique, excepté dans certains langages. D'un point de vue fonctionnel, une application mono thread marche très bien. Certes. Mais, le multithread possède des avantages qui font réfléchir. Qu'apporte le multithread dans une application ? Sur le plan architecture, cela permet une plus grande souplesse et le code est bien mieux structuré, tout en évitant du code redondant. Côté performance, le multithread offre des temps de réponse bien meilleurs et chaque fonction lourde devient indépendante, ce qui évite les blocages, tout en offrant une réactivité optimale (ex. : découplage des fonctions et l'interface utilisateur). Basiquement, le multithread permet de découper les fonctions et le code. Malgré tout, il nécessite une solide gestion des threads. Cela demande donc un découpage rigoureux des fonctions et de faire communiquer les threads. Quand on parle de multithreading, on parle bien entendu de priorité. La priorité permet de dire quel thread doit être exécuté et achevé en premier par rapport à d'autres. Mais, on peut avoir plusieurs threads ayant la même priorité et donc, ils s'exécuteront en même temps (enfin, suivant les autorisations du système). Mais là aussi dans certains cas, il peut y avoir des " ratés ", à savoir une inversion des priorités ! Un thread non prioritaire prend " le dessus " sur un thread prioritaire ! Il est plus facile de mettre en œuvre le multithread dans une application à créer, que dans une application existante. Dans un programme déjà existant, cela implique une refonte toujours (ou presque) aussi bien de l'architecture, du code que des fonctions ! Même si on ne conçoit pas l'ensemble de son logiciel en multithread, on peut tout de même isoler les fonctions les plus lourdes des autres. De toute manière dans les applications parallèles et distribuées, le multithread est indispensable. À l'origine, le thread permettait d'exploiter les capacités des systèmes multiprocesseurs. Habituellement, un processus normal possède un seul et unique thread. Dans le cas d'un processus multithreadé, on a plusieurs exécutions simultanées. Quand on parle de threads, il ne faut pas omettre les notions de ressources et donc de partage de ressources. N'oublions pas que les threads utilisent les ressources allouées à un processus. Plus il y a de threads, plus les ressources se fragmentent. Là, peut alors jouer la notion de priorité. Mais le nombre de threads alourdit d'autant la gestion et le logiciel.

La synchronisation

L'une des difficultés dans le multithreading est l'accès aux objets (ainsi qu'aux données). Si par exemple, deux threads accèdent au même objet et le modifient, il y a risque d'obtenir à la fin un objet corrompu. Là encore, le développeur doit parfaitement gérer cette " concurrence ". C'est à vous de synchroniser les accès dans les différents threads (et si possible dans la joie et la bonne humeur, cela va de soi en programmation !). Préférez donc la synchronisation à la non synchronisation. La synchronisation évite toute collision d'accès à un objet, ou à une donnée, par plusieurs threads en même temps. Avec la notion de synchronisation, on trouve la notion de verrou. En fait, c'est un concept très simple. Si le thread A veut accéder à une méthode synchronisée, il ouvre " la porte " et verrouille l'accès. Si un thread B veut accéder à la même méthode, impossible, il ne pourrait pas rentrer, donc, il se mettrait probablement en attente. Bien entendu, quand le thread A quitte la méthode, le verrou redevient ouvert.

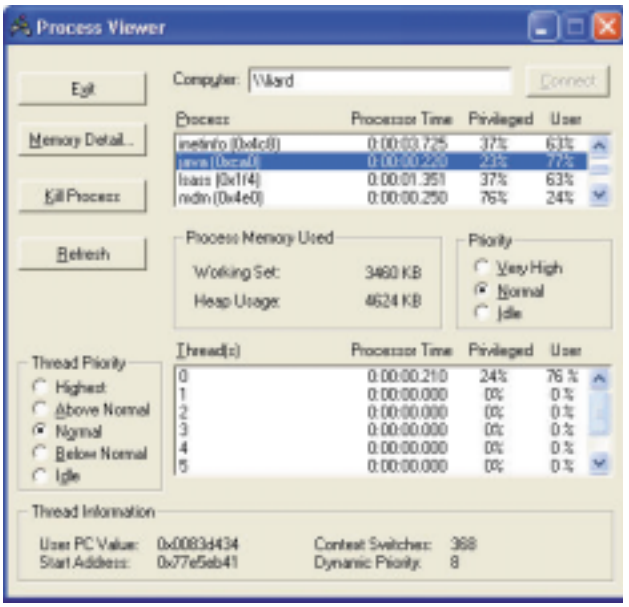
Systèmes multitâches coopératifs – multitâches préemptifs

Il existe deux grands types de systèmes multitâches : coopératif, préeemptif. Dans le cas d'un système coopératif, les applications autorisent ou non d'autres applications à exécuter des fonctions. Le système n'intervient pas dans la répartition des ressources et temps processeurs. Le multitâche préemptif est inverse. Le système gère tout. C'est l'OS qui va répartir le temps processeur et les ressources à chaque application, et process interne pour mener à terme chaque processus. Cela peut-être fait selon la priorité de chaque process.

Threads & Java

Le multithread dans Java est un aspect fondamental du langage. Lorsqu'une applet ou application Java démarre, il y a création d'un thread principal (fonction main). La création de nouveaux threads se réalise soit en instanciant la classe thread, soit en dérivant la dite classe. La méthode start permet de rendre actif le thread.

Le développeur n'a pas toute liberté lorsqu'il crée un nouveau thread Java ou .NET. Par exemple, en Java, le thread doit dire aux autres threads s'il devient inactif. Il ne faut pas oublier que dans un OS préemptif, c'est le système qui définit et répartit le temps processus et non l'application. Pour qu'un thread s'exécute, il faut que l'OS lui accorde du temps processeur. Mais pour faire fonctionner des threads, il ne faut pas qu'un thread fonctionne en permanence. Sinon, il y a risque de blocage des autres processus légers. C'est pour cela qu'il y a des périodes de fonctionnement et des périodes d'attente ou de sommeil. Si par défaut, un thread, lorsqu'il est créé, possède sa priorité (en Java, il hérite de la priorité du thread parent), c'est aux développeurs de s'occuper de la gestion des priorités. Dans le langage Java, il y a 10 niveaux de priorités. Mais là encore, l'application n'est pas seule à bord (et encore moins le thread). Le système a aussi son mot à dire. Ainsi, quand on exécute une applet Java possédant plusieurs threads, tout dépend de l'implémentation de la JVM sur l'OS hôte. Et il n'y a pas forcément toujours la même concordance dans les niveaux de priorités entre ceux de la JVM (et donc de l'applet) et de l'OS. Que se passe-t-il quand plusieurs threads possèdent une priorité haute ? Le gestionnaire des priorités choisit à votre place l'un des threads à exécuter en premier ! Java ne garantit absolument pas un traitement équitable de tous les threads. Bref, d'une JVM à une autre, d'un OS à un autre, le comportement des threads d'une applet risque de changer...



Threads dans C# / .NET

Basiquement sous .NET, on peut distinguer 3 " ensembles " :

- la classe Thread sert à créer et à gérer les threads
- la classe Monitor permet d'avoir un mécanisme de synchronisation dans les accès aux objets
- la classe Mutex offre une synchronisation primitive que l'on peut implémenter pour synchroniser plusieurs threads.

Finalement la mise en œuvre et la gestion des threads ne diffèrent pas des autres langages.

Sous .NET, le point de départ est la classe System.threading.thread. Il est possible d'instancier autant d'objets threads que l'on veut. On peut aussi exécuter n'importe quelle méthode d'une classe quelconque (la classe doit être référencée dans votre application). C'est l'objet thread qui contrôle le thread. On y définit sa priorité et on y surveille son état. La création d'un thread dans un projet se déroule en deux étapes : tout d'abord on crée un délégué, puis on passe celui-ci à la nouvelle instance thread.

En résumé...

Impossible d'aborder le multithread en seulement deux pages, le sujet est immense. Il y a quelques notions que l'on doit garder à l'esprit quand on développe en multithread :

- architecture et découpages des fonctions de l'application
- gestion des priorités
- synchronisation aux accès des objets et données

C'est quoi un thread ?

Le thread est une suite d'instructions. Il implémente un compteur indiquant au thread la prochaine instruction à exécuter. Le thread se situe dans un processus et partage la mémoire du dit processus. Cette " astuce " permet de gagner un temps précieux quand on change de thread. C'est pourquoi, on appelle aussi le thread, processus léger.

On a aussi vu que Java pose quelques contraintes supplémentaires et qu'une partie de la gestion des priorités est liée à l'implémentation de la JVM sur les systèmes. Nous n'avons pas abordé le problème des outils, mais il va de soi que pour faire du debug multithread, il faut avoir des débogueurs adéquats (ex. : sous Linux, il y a SmartGDB). Pour le comportement des threads, optez pour un multithreading viewer. Mais finalement, d'un langage à un autre (ou dans un système), les principes restent identiques. Ainsi si on comparait un thread Posix, un thread Java, un thread .NET, on y verrait bien des similitudes.

■ François Tonic

Qu'est-ce qu'un framework de mapping O/R ?

Suite de : **QU'EST-CE QU'UN FRAMEWORK ?** *de Jean Degos, Aston, paru dans le n°55*

De nombreuses applications utilisent aujourd'hui les langages objets, tel que C++, C#, Java, ainsi que des bases de données relationnelles pour le stockage des données.

Afin de rendre un objet persistant dans une base de données relationnelle, il est nécessaire d'écrire le code SQL adéquat et ceci, pour chaque objet concerné.
La principale difficulté rencontrée lors de cet exercice est ce que l'on appelle l' "impedance mismatch". Un exemple classique est la relation 1:n entre deux entités A et B (voir schéma) : dans le monde objet c'est A qui tient la liste des références vers B, tandis que dans le monde relationnel chacun des B possède une référence vers A.

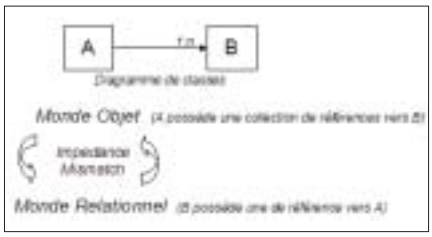


Fig 1 - "Impedance Mismatch" entre le monde objet et relationnel.

Le but d'un framework de mapping O/R est de rendre persistant, de préférence de manière transparente, les objets métiers.

Pourquoi utiliser un framework O/R ?

Ce type d'outil permet :
- un gain de temps lors du développement : le développeur peut se concentrer sur la valeur ajoutée d'une application (les fonctionnalités, l'ergonomie).
- une aide à la structuration du code (pas de "pollution" dans le code Java par du code SQL)
- une maintenance et relecture du code de l'application grandement facilitées.

Il faut garder à l'esprit que l'utilisation d'un framework O/R implique une perte logique de performances, si l'on compare ce type d'accès à des connexions directes à la base de données.

(1) ODMG : Object Data Management Group. ODMG 3.0 est une spécification portable écrite par l'ODMG et dont il existe des implémentations en C++, Java, Smalltalk.
(2) JDO : Java Data Object. JDO est une spécification de SUN (JSR-12).

Présentation d'un framework Open Source de mapping O/R : OJB

OJB est un framework de mapping O/R Java et Open Source, mis à disposition par "Apache Software Foundation" au sein du "Apache Jakarta Project", et qui permet la persistance transparente des objets Java dans une base de données relationnelle.
OJB propose une implémentation complète de l'api ODMG 3.0, une api propriétaire appelée "PersistenceBroker", ainsi que l'accès à l'api de JDO 1.0 (via un plugin s'appuyant sur l'implémentation de Référence JDO de Sun).
Quel est le principe de fonctionnement d'une application utilisant OJB ?

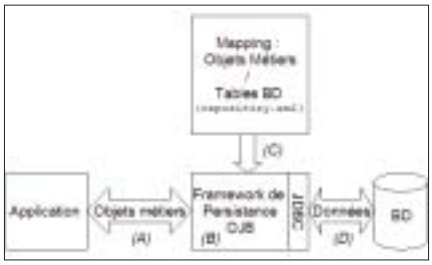


Fig 2 - Schéma de principe général.

Tout d'abord, notre application manipule des objets métiers persistants (A). Ensuite, cette persistance des objets métiers est assurée par le framework de persistance (B). Ce dernier s'appuie sur un fichier de mapping qui décrit comment les objets métiers doivent persister dans la base de données et inversement (C). Finalement l'accès à la base de données est pris en charge par un driver JDBC adéquat (E). Le schéma suivant présente l'organisation de l'api OJB :

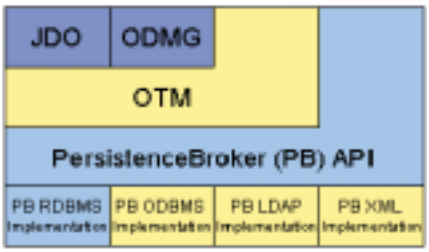


Fig 3 - Organisation de l' api OJB.

Exemple d'utilisation de l'api OJB

Cet article vous montre comment on peut utiliser l'api OJB nommée " PersistenceBroker ". Prenons comme exemple un programme de gestion des comptes clients dans une banque. Nous supposons que nous avons les éléments suivants à notre disposition :
- un MPD (Modèle Physique des Données),
- un diagramme de classe.

Nous vous proposons d'étudier successivement l'élaboration du fichier de mapping O/R qui servira de lien entre la base de données et nos objets, le code permettant de créer et enfin de récupérer des données dans la base. Toutes les informations permettant de faire fonctionner notre exemple ne sont pas présen-

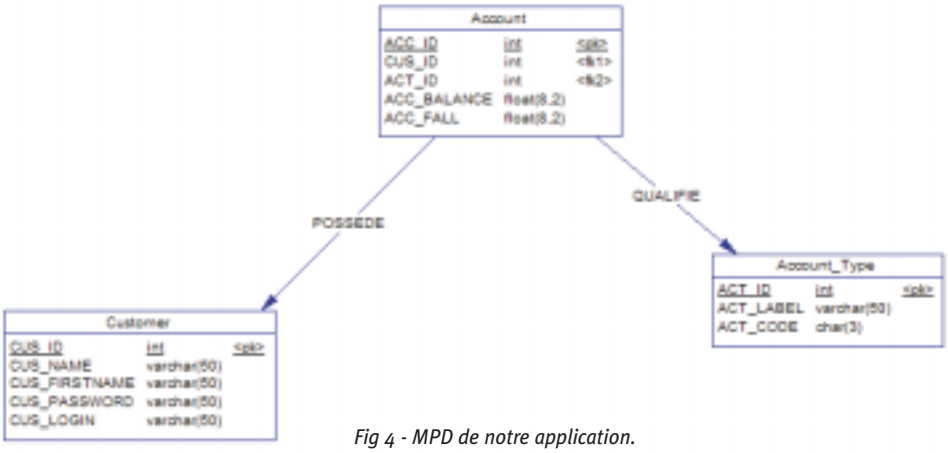


Fig 4 - MPD de notre application.

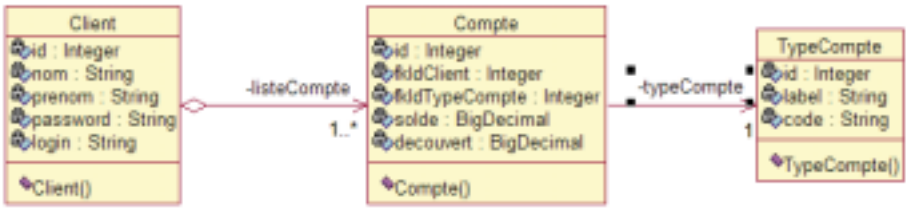


Fig 5 - Diagramme de classe de notre application.

tées dans cet article pour des raisons évidentes de pédagogie et de place dans ces colonnes. Nous vous invitons à télécharger l'ensemble des fichiers sources sur le site d'ASTON : www.aston.fr. (Fig 4 et 5)

Comme nous pouvons le voir sur le diagramme de classe, nous avons à gérer une relation 1:n entre un Client et Compte, et une relation 1:1 entre un Compte et TypeCompte.

La première étape consiste à écrire le fichier XML de mapping décrivant comment passer du monde objet vers le monde relationnel et inversement. Ce fichier est nommé *repository_user.xml*.

Nous y retrouvons chacune des classes qui doivent persister dans la base et nous associons chacune d'elles avec une table de la base de données. A chaque attribut (name) de classe on affecte un numéro unique (id), un champ dans la table associée (column) et le type base de données (jdbc-type).

Il est possible d'indiquer quel attribut correspond à la clef primaire de notre table (Primarykey) et si OJB doit prendre en charge l'incrémentation automatique de celui-ci (autoincrement).

Nous allons voir maintenant comment est décrit le mapping des relations 1:1 et 1:n.

RELATION 1:1

Au niveau du diagramme UML, la classe Compte est associée à la classe TypeCompte afin de qualifier un compte. Pour que cette association soit effective, nous devons la matérialiser dans le mapping de la classe Compte.

Nous utilisons la balise <reference-descriptor…></reference-descriptor> pour décrire l'association.

Cette balise possède les attributs suivants :

- name : (Obligatoire) Nom de l'attribut correspondant à l'association.
- class : (Obligatoire) Type de la classe associée.

Les attributs suivants sont facultatifs.

- auto-retrieve : (true/false) Indique que la classe associée doit être remontée automatiquement en mémoire lorsque la classe principale est instanciée en mémoire.
- auto-update : (true/false) Indique la mise à jour de la classe associée pour chaque mise à jour de la classe principale.
- auto-delete : (true/false) Supprime la classe associée lorsque la classe principale est supprimée (symbolise la composition).

Pour récupérer la classe associée, tout comme en SQL, OJB se sert de la clef étrangère portée par la classe principale. Ainsi vous devez inclure la balise <foreignkey ../> dans la balise <reference-descriptor..></reference-descriptor> pour indiquer à OJB la clef étrangère qui permet de récupérer la totalité de la classe associée.

La balise "<foreignkey ../>" possède un seul attribut :

- field-id-ref : La valeur de ce champ correspond à l'indice de la clef étrangère dans le mapping de la classe principale. Dans notre exemple, la clef étrangère fkIdTypeCompte est décrite en position 5.

Note : Les attributs auto-retrieve, auto-update, auto-delete doivent êtres positionnés à true avec la plus grande attention. Le risque étant de perdre le contrôle des accès (lecture, mise à jour, insertion) faits dans la base de données par OJB.

RELATION 1:N

Au niveau du diagramme UML, il existe une relation 1:n entre la classe Client et la classe Compte. Pour que cette association soit effective, nous devons la matérialiser dans le mapping de la classe Compte.

Nous utilisons la balise <collection-descriptor…></collection-descriptor> pour décrire l'association.

Cette balise possède les attributs suivants :

```
<class-descriptor
        class="domaine.Compte"
        table="ACCOUNT">

    <field-descriptor id="1"
        name="id"
        column="ACC_ID"
        jdbc-type="INTEGER"
        primarykey="true"
        autoincrement="true"
    />
    [...]
    <field-descriptor id="5"
        name="fkIdTypeCompte"
        column="ACT_ID"
        jdbc-type="INTEGER"
    />
    <reference-descriptor
        name="typeCompte"
        class-ref="domaine.TypeCompte"
        auto-retrieve="true"
        auto-update="false"
        auto-delete="false"
    >
        <foreignkey field-id-ref="5"/>
    </reference-descriptor>

</class-descriptor>
```

Fig 6 - Relation 1:1 entre Compte et TypeCompte décrite dans le fichier de mapping O/R

```
<class-descriptor
      class="domaine.Client"
      table="CUSTOMER">

   <field-descriptor id="1"
      name="id"
      column="CUS_ID"
      jdbc-type="INTEGER"
      primarykey="true"
      autoincrement="true"
   />
   [...]
   <collection-descriptor
      name="listeCompte"
      element-class-ref="domaine.Compte"
      auto-retrieve="false"
      auto-update="true"
   >
      <inverse-foreignkey field-id-ref="4"/>
   </collection-descriptor>

</class-descriptor>

<!-- Definitions for Compte -->

<class-descriptor
      class="domaine.Compte"
      table="ACCOUNT">

   <field-descriptor id="1"
      name="id"
      column="ACC_ID"
      jdbc-type="INTEGER"
      primarykey="true"
      autoincrement="true"
   />
   <field-descriptor id="4"
      name="fkIdClient"
      column="CUS_ID"
      jdbc-type="INTEGER"
   />
```

Fig 7 - Relation 1:n entre Client et Compte décrite dans le fichier de mapping O/R

- name : (Obligatoire) Nom de l'attribut correspondant à l'association.
- class : (Obligatoire) Type de la classe associée.

Les attributs suivants auto-retrieve, auto-update, auto-delete sont facultatifs (cf. relation 1:1) Il faut inclure la balise ‹inverse-foreignkey ../› dans la balise ‹collection-descriptor..›‹/collection-descriptor› pour indiquer à OJB la clef étrangère qui permet de récupérer la totalité de la classe associée. La balise <inverse-foreignkey ../> possède un seul attribut : field-id-ref : La valeur de ce champ correspond à l'indice de la clef primaire dans le mapping de la classe cible (Compte dans notre exemple). Maintenant que nous avons le fichier de mapping O/R nous pouvons écrire le code en java et vous montrer l'intérêt d'OJB.

Nous allons tout d'abord insérer de nouveaux enregistrements dans notre base de données :

```
PersistenceBroker broker = PersistenceBroker
Factory.defaultPersistenceBroker();

Client client = new Client();
client.setPrenom("Jean-Claude");
client.setNom("Dus");
client.setLogin("jcd");
client.setPassword("splendide");

Compte compte = new Compte();
compte.setTypeCompte(getTypeCompteCOMPTE
COURANT(broker));
compte.setSolde(new Double("1000.00"));
compte.setDecouvert(new Double("1000.00"));
client.ajouterCompte(compte);

// Commander à notre objet client de s'enregistrer
en base
broker.store(client);
```

La première étape consiste à récupérer une instance de la classe PersistanceBroker. C'est ce dernier qui va assurer la persistance des objets en base de données en exploitant le fichier XML de mapping. Le reste du code consiste à remplir tout simplement nos objets avec des données. Finalement nous commandons l'enregistrement en base avec la ligne broker.store(client).

Pour restituer un objet persistant, c'est tout aussi simple :

```
PersistenceBroker broker = PersistenceBroker
Factory.defaultPersistenceBroker();

// Spécifier notre critère de recherche
Criteria crit = new Criteria();
crit.addGreaterOrEqualThan("solde", "1000");

// Notre requête utilisera le critère de recherche 'crit'.
// Notre critère de recherche s'appuie sur la classe
'Compte'.
QueryByCriteria q = new QueryByCriteria
(Compte.class, crit);

// Lancer la récupération avec notre critère de recherche
Vector listeReponse = (Vector) broker.getCollection
ByQuery(q);
```

Ici, la démarche consiste en la récupération de tous les comptes dont le solde est supérieur à 1000.

Comme précédemment nous instancions un PersistanceBroker, puis nous créons une requête (QueryByCriteria) dont le critère de recherche (Criteria), s'appuie sur la classe Compte (Compte.class). Pour terminer nous exécutons la requête broker.getCollectionByQuery(q).

Jean-Baptiste Claramonte, ASTON

Créée en 1990, ASTON fonde sa stratégie sur la maîtrise des technologies alliée à une véritable capitalisation des problématiques métiers de ses clients.

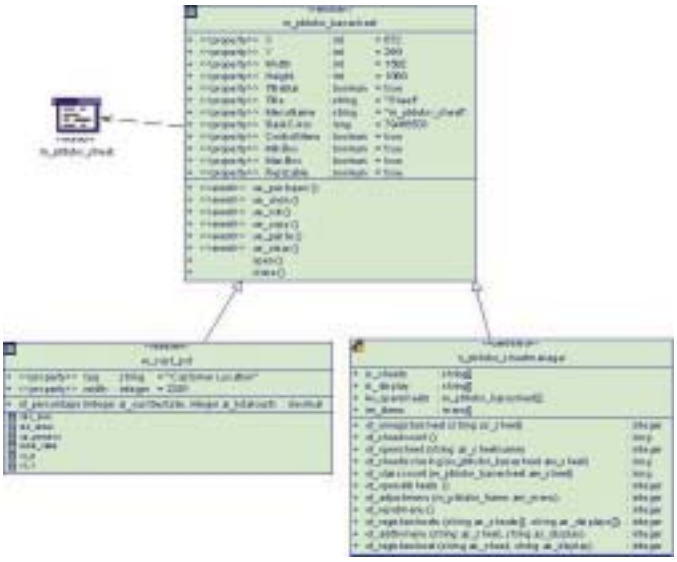
ASTON propose des prestations de conseil, assistance à maîtrise d'ouvrage, réalisation d'applications au forfait, en assistance technique, tierce maintenance applicative et formation.

Nos domaines de prédilection sont : le développement, l'intégration, le workflow, le décisionnel, les portails et Internet. Les résultats de notre cellule de veille technologique font l'objet de publications régulières dans la presse.

Par ailleurs, ASTON organise périodiquement des séminaires sur des sujets techniques d'actualité. Renseignements et inscriptions : www.aston.fr

PowerAMC & PowerBuilder : UML pour une meilleure productivité

PowerAMC 9.5.2 donne aux développeurs une productivité améliorée. Cette version prend en charge les versions 6, 7, 8 et 9 de PowerBuilder et permet la rétro-conception et génération des objets PowerBuilder, grâce au support de l'UML.



UML se compose de diagrammes favorisant l'analyse des exigences métiers en termes techniques et leur conversion en modèles implémentables. Le diagramme le plus important pour la mise en œuvre du modèle est le diagramme de classes UML. La rétro-conception du code PowerBuilder crée un diagramme de classes UML dans PowerAMC. C'est à partir de ce diagramme qu'est généré le code PowerBuilder.

PowerAMC met en correspondance les concepts UML avec les objets PowerBuilder comme suit : Application, User Object, Structure, Function, Window et Proxy sont modélisés sous la forme de classes. Les bibliothèques de support (.pbl) et les fichiers source (.srx) seront modélisés comme objets classés en fonction du type identifié dans PowerBuilder. Par exemple, un Menu sera un Menu dans PowerAMC. Signalons à ceux qui connaissent bien UML que nous utilisons les concepts «stereotype» et Profil UML pour mettre en œuvre une classe de type Menu dans PowerAMC. L'utilisation de l'UML avec PowerBuilder a de nombreux avantages. On peut, par exemple, utiliser la palette spéciale d'outils PowerBuilder de PowerAMC pour créer de nouveaux objets, permettant de rationaliser le développement de nouvelles applications pour des systèmes orientés objets ou n-tiers. Il est également possible d'utiliser les fonctions de Vérification de modèle de PowerAMC, pour contrôler la validité de vos modèles. Et comme l'UML de PowerAMC est directement lié à la conception de bases de données (les modèles physiques de données de PowerAMC), PowerAMC est en mesure de gérer la mise en correspondance objet/relationnel, qui détermine comment les composants d'accès aux données sont reliés aux tables de la base de données et comment générer et rétro-concevoir les attributs étendus de PowerBuilder.

Rétro-conception

Comprendre une application PowerBuilder existante

Si vous disposez d'une application PowerBuilder, mais pas d'une documentation utile concernant sa conception, vous pouvez aisément créer un nouveau diagramme de classes UML dans PowerAMC. Grâce à cette représentation visuelle des interdépendances du code PowerBuilder, vous pouvez relancer la maintenance de l'application, l'étendre avec de nouvelles capacités, l'intégrer à d'autres applications, ou la refondre dans une nouvelle architecture, par exemple une architecture n-tiers ou des services Web.

Pour rétro-concevoir avec PowerAMC 9.5.2, allez dans le menu Language et sélectionnez Reverse Engineering PowerBuilder. Vous pouvez effectuer cette opération en lisant soit les fichiers .PBL soit les fichiers source (objets PowerBuilder exportés).

Le diagramme de classes

Navigation dans le modèle PowerBuilder

Une fois la rétro-conception de votre application PowerBuilder menée à bien, vous pouvez commencer à étudier la composition du code existant. Le modèle UML se compose comme suit : un package distinct est créé lors de la rétro-conception de chaque bibliothèque. Les objets sont représentés sous la forme de classes dans ces packages, avec un stéréotype et une mini-icône qui vous indiquent de quel type d'objet il s'agit. Pour déterminer les dépendances des objets entre eux, vous pouvez utiliser une combinaison de fonctions UML. Vous utiliserez les associations UML issues de la rétro-conception, la liste des dépendances propre à chaque objet pour signaler quel menu ou DataWindow est utilisé par quelle fenêtre, et les classes internes pour représenter les commandes.

Fenêtre, menus et plus...

Gestion des objets BLOB

Certains objets PowerBuilder ne peuvent pas être mis en correspondance avec une classe en utilisant un stéréotype. Ces objets PowerBuilder sont la DataWindow, le menu, la requête, le pipeline et l'objet binaire. Tous ces objets sont représentés dans PowerAMC sous la forme d'une classe spéciale dans laquelle une icône remplace le symbole de classe. Pour conserver totalement l'objet d'origine, le code source est stocké dans la section Début de l'onglet Script. Lorsque vous vous apprêtez à regénérer le code source, remettez en place l'objet d'origine intact. Vous pouvez effectuer des modifications à partir de l'interface PowerBuilder et les synchroniser avec PowerAMC si nécessaire.

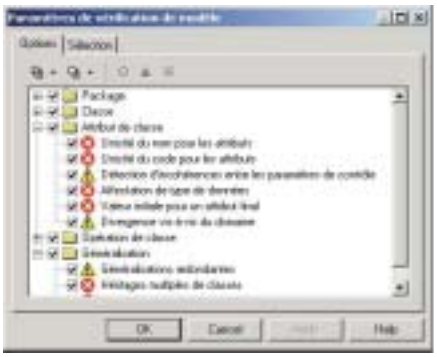
Création de nouveaux objets

Travailler avec le diagramme de classes

Partant du diagramme issu de la rétro-conception, vous pouvez modifier des associations, changer le fonctionnement d'une classe, créer de nouveaux objets à utiliser avec les objets existants ou bien commencer à réorganiser les éléments de code en une nouvelle conception de type n-tiers ou orientée service. Pour créer de nouveaux objets PowerBuilder, utilisez la nouvelle barre d'outils PowerBuilder de PowerAMC.

Lorsque vous modifiez le diagramme UML, vous pouvez aisément vérifier l'état d'avancement de vos travaux en utilisant la fonction "Aperçu du Code". Cliquez avec le bouton droit de la souris sur n'importe quelle classe ou objet et sélectionnez "Aperçu du Code" dans le menu contextuel. Si vous trouvez un élément que vous souhaitez modifier, vous pouvez modifier le code PowerBuilder dans la fenêtre de l'aperçu et cette modification sera répercutée dans le modèle et l'objet généré.

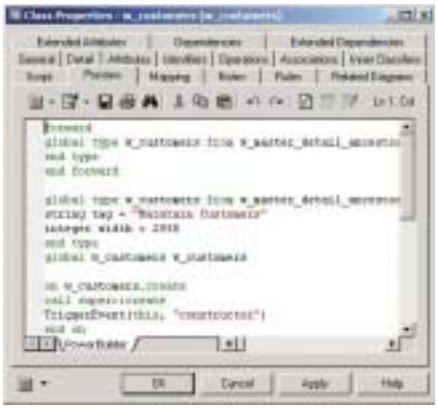
Comme il est important de documenter le système pour des besoins ultérieurs, vous pouvez ajouter des descriptions en format RTF au modèle. Il est possible de documenter chaque objet. Ce travail de documentation peut être partagé avec des non-utilisateurs de PowerAMC, grâce à l'éditeur de rapports. Cette fonction permet à l'ensemble de la documentation au format RTF, avec tous les graphismes



et les métadonnées concernant le modèle que vous choisissez, de partager, d'être générée, soit sous la forme d'un document Word (au format .RTF), soit d'un ensemble de documents avec liens HTML.

Travailler avec des contrôles

Comme indiqué antérieurement, les contrôles sont traités de manière particulière, car ils sont représentés sous la forme de classificateurs internes. Pour voir la liste des contrôles, ouvrez les propriétés d'une fenêtre, sélectionnez l'onglet Classificateurs Internes et ouvrez les propriétés du classificateur interne. En modifiant, créant et supprimant ces classes internes, vous pouvez utiliser PowerAMC pour concevoir comment utiliser vos contrôles et générer le résultat dans PowerBuilder pour un test final et pour le déploiement.



Préparation à la génération

Aperçu et modification du code

Avant de générer de nouveaux objets dans votre environnement PowerBuilder, vous souhaitez probablement prévisualiser d'abord le code. Pour ce faire, ouvrez les propriétés d'un objet et sélectionnez l'onglet Aperçu. Pour les fenêtres, les objets utilisateur, les objets application, les structures, les fonctions et les proxies, vous pouvez modifier le code que vous voyez dans la fenêtre d'aperçu (par exemple ajouter des attributs, modifier des opérations, etc.). Le modèle sera actualisé pour prendre en compte ces modifications.

Vérifier le modèle

PowerAMC vous aide à produire une application PowerBuilder fiable et de qualité, en appliquant des principes orientés objet. Pour préserver la cohérence nécessaire pour garantir la qualité, PowerAMC propose une fonction complète de vérification des modèles. Pour lancer la vérification de modèle à tout moment, allez dans le menu Outils et sélectionnez Vérifier Modèle. La fonction de validation de modèle, qui peut être personnalisée, est lancée et vérifie que le modèle est correctement défini.

PowerAMC comprend les vérifications générales orientées objets telles que définies dans le langage UML, mais il est capable de faire mieux en ce qui concerne PowerBuilder. Pour garantir une conception UML de qualité, et une conception PowerBuilder de qualité, PowerAMC a vu sa fonctionnalité de vérification de modèle enrichie par des vérifications spécifiques à PowerBuilder.

Génération de l'application

Terminer le travail

Une fois que vous avez procédé à la vérification du modèle, que vous avez fini d'afficher et de modifier le code, vous êtes prêt à regénérer l'application dans PowerBuilder. Notez toutefois qu'il n'est pas souhaitable à ce stade que cette génération soit irréversible, vous devez être en mesure de rétro-concevoir et de générer des objets PowerBuilder, de manière itérative, pendant toute la durée du cycle de vie de développement, afin d'assurer que la conception progresse avec le code à mesure que celui-ci est rédigé. Vous souhaitez également vous assurer que le code rédigé reste cohérent par rapport aux principes établis dans le modèle d'origine. Pour générer du code dans PowerBuilder, utilisez le menu Language et sélectionnez Générer pour PowerBuilder. À partir de cette fonction, vous pouvez choisir les objets que vous souhaitez générer, d'une seule fenêtre à la totalité du modèle vers un fichier .PBL ou vers les fichiers source (.SRX) qui peuvent être importés vers n'importe quelle bibliothèque PowerBuilder.

■ **David Dichmann**

PowerAMC et PowerBuilder

L'avantage UML

Les nouvelles capacités de rétro-conception et de génération de code PowerBuilder de PowerAMC sont maintenant étendues à tous les objets PowerBuilder. La possibilité de créer de nouvelles applications PowerBuilder rapidement et efficacement, en gérant aisément les complexités inhérentes aux architectures modernes n-tiers ou orientées service, permet aux développeurs d'accroître leur productivité. Les fonctions de rétro-conception et de regénération de code vous aident à réutiliser la logique applicative existante lors de la transition vers le Web. La documentation des systèmes peut être produite à mesure que vous découvrez des systèmes propriétaires et rédigez la documentation.

Programmez avec Eclipse

2ème partie

SWT est la boîte à outils graphiques sur laquelle est construit Eclipse lui même. Les développeurs d'Eclipse ont en effet estimé bon de développer une librairie de remplacement à Swing, vraiment trop lente.
A en juger par le résultat, ils n'ont pas eu tort, en dépit du fait que cela implique que le programmeur SWT doit être plus attentif à la libération de ressources que ne l'est le programmeur Swing. Après avoir appris à utiliser Eclipse comme un environnement de développement, allons plus loin dans SWT.

Sur le CD ROM

La structure d'une application SWT

Le listing 1 est un programme SWT minimum, de type 'Hello World' (figure 1). Ce simple listing met en relief les points importants de la programmation sous SWT. Dans la méthode Main, on commence par instancier un objet de type Display. Comme son nom ne l'indique pas, cette classe n'a rien à voir avec un quelconque affichage. Ou du moins pas directement. La classe Display a pour rôle de s'occuper de la collaboration entre SWT et le système d'exploitation de la plate-forme hôte. Principalement, Display met en place une boucle d'événement à la manière des boîtes à outils en C ou C++. Ceci implique quelques particularités :

Figure 1 - *Ben oui, encore le bon vieux 'Hello World' !*

Figure 2 - *La fenêtre de notre application s'adapte maintenant aux événements.*

- Au démarrage, la JVM lance la méthode main de la classe principale dans un thread. Avec SWT, cela implique que ce thread devient le thread de l'interface utilisateur. C'est donc ce thread qui doit impérativement se charger de travailler avec la boucle des événements.
- Le programme doit écrire une boucle qui invoque la méthode readAndDispatch de Display, afin de retirer les événements de la queue et de les transmettre aux widgets concernés. Voir le premier programme d'exemple.
- La plupart des méthodes publiques des widgets doivent être invoquées uniquement depuis ce thread. Si un thread séparé doit rafraîchir l'interface utilisateur, il le fera en utilisant les méthodes de communication inter-threads de la classe Display. Cette contrainte existe aussi avec Swing d'ailleurs.
- Le thread principal ne serait pas bien inspiré du tout de créer une seconde instance de Display, car cela reviendrait implicitement à créer une deuxième boucle d'événements pour la même application.
- Le thread principal est en contact direct avec le système d'exploitation, et a la charge d'en retirer les événements de la file d'attente. Pour cette raison, il faut éviter de lancer de grosses boucles de calcul dans ce thread. En effet, si le système d'exploitation est trop délaissé dans la gestion des événements, il peut s'en trouver déstabilisé. Il convient donc de lancer les longs calculs dans des threads séparés qui, éventuellement, rafraîchiront l'interface utilisateur, comme mentionné ci-dessus avec les méthodes de communication inter-threads de Display, les méthodes syncExec et asyncExec notamment. Les invocations de ces méthodes lançant elles mêmes un thread séparé. Voici un exemple pour nous clarifier les idées :

```
Display.getDefault().asyncExec(new Runnable() {
public void run() {
monWidget.redraw();
}
});
```

Finalement, ceci est dans le même esprit et fort semblable à ce que nous ferions de toutes façons sous Swing :

```
EventQueue.invokeLater(new Runnable() {
public void run() {
                                monWidget.repaint();
}
});
```

La libération des ressources

Bien que la JVM dispose d'un ramasse-miettes et que celui-ci soit bien actif sous SWT, le caractère bas niveau de cette librairie impose au programmeur de gérer la libération des ressources. On utilisera pour cela les méthodes 'dispose' des classes SWT. C'est d'ailleurs ce que fait le premier programme d'exemple qui invoque bien la méthode dispose de la classe Display, à la fin de la méthode main. La règle de travail est simple et tient en deux points :

- Tous les objets SWT instanciés par nous, doivent être libérés par nous à une exception près: les instances des classes Shell et leurs éventuels contrôles enfants. En effet ces classes encapsulent les fenêtres. Celles-ci peuvent être fermées par un clic de souris de la part de l'utilisateur. Dans ce cas, le système se charge de libérer

la ressource, ainsi que les ressources attachées à tous les widgets contenus dans la fenêtre. Bien entendu, on peut aussi explicitement appeler la méthode dispose pour une fenêtre, mais dans ce cas on veillera à s'assurer que la ressource n'est pas déjà libérée. Ceci se fait en invoquant la méthode isDisposed de l'objet. Le premier programme d'exemple illustre ceci.

- Les ressources qui nous parviennent du système ne doivent pas être libérées, puisqu'en vertu de la première règle, c'est le système qui doit s'acquitter de cette tâche.

Il tombe sous le sens que ce que nous venons de dire ne concerne que les classes SWT, qui ont quelque chose à voir avec le système d'exploitation. Ainsi, si ce que nous venons de dire est vrai pour une classe Shell, cela ne s'applique évidemment pas à la classe Point de SWT car cette classe n'est rien d'autre qu'une structure de données.
Il est certain que ces contraintes sont un peu en dehors de l'esprit Java, puisqu'il y a là des risques potentiels de fuite mémoire. En revanche les interfaces utilisateur écrites avec SWT sont vives comme l'éclair, puisque de bas niveau.

Les fenêtres

Nous l'avons déjà évoqué plus haut, c'est la classe Shell qui encapsule les fenêtres. Cette classe sert pour tous types de fenêtres. Si l'on souhaite créer une fenêtre réduite, ou une boîte de dialogue modale, par exemple, on instancie la classe Shell en passant des bits de style, particuliers à son constructeur, qui reçoit en outre une référence sur l'instance de Display. Les bits de style sont définis dans le paquet org.eclipse.swt.

Les événements

Comme il n'est pas d'application GUI qui ne soit capable de réagir à son environnement, nous abordons maintenant l'aspect "Evénements" de la programmation avec SWT. Nous voulons que notre application trace un cadre centré dans la fenêtre, qu'elle écrive le texte "Programmez!" également centré dans cette fenêtre, ceci au gré des redimensionnements éventuels. Pour cela, SWT offre deux façons de traiter les événements. La première façon est tout à fait semblable à celle de Swing. Des interfaces définissent des méthodes d'écouteurs (listeners) devant réagir à des événements. A charge d'une classe d'implémenter ces interfaces puis de se faire connaître du système par un appel à addxxxListener, ou xxx est le nom de l'événement. Comme avec Swing, la technique des classes anonymes peut être utilisée, et comme Swing, SWT propose des adaptateurs d'événements, afin d'éviter au programmeur d'implémenter une ribambelle de méthodes ne faisant rien pour les événements compliqués, c'est à dire des événements décrits par une interface déclarant plus d'une méthode. Le deuxième programme d'exemple ci-contre (listing 2) applique cette méthode. Une classe anonyme pour l'événement PaintEvent (une seule méthode) et utilisation d'un adaptateur pour l'événement ControlEvent (deux méthodes) qui survient lorsque la fenêtre est déplacée ou redimensionnée. L'autre possibilité proposée par SWT consiste à enregistrer un écouteur générique pour tous les événements qui, lorsqu'ils surviennent, ont tous le même type: Event. D'où le nom (pas très adapté) d'événements non typés. Le programmeur implémente donc une seule interface, contenant une seule méthode: handleEvent. Dans cette méthode, l'on doit tester le type réel de l'événement au sein d'une construction switch-case, à la manière des application Windows, ou X-Windows natives écrites en C comme autrefois. Le programmeur doit en outre enregistrer un écouteur pour chaque événement susceptible d'être traité par handleEvent, comme le montre le troisième programme d'exemple ci-contre (listing 3) Bonne programmation avec Eclipse.

■ **Frédéric Mazué** - *fmazue@programmez.com*

(listing 1)

```
// Un programme 'Hello World' minimal

import org.eclipse.swt.*;
import org.eclipse.swt.widgets.*;

public class Hello {

   public static void main(String[] args) {
        Display display = new Display();
        Shell shell = new Shell(display);
shell.setBounds(0, 0, 300, 200);
        Label label = new Label(shell, SWT.CENTER);
        label.setText ("Hello world avec Eclipse et Programmez!");
        label.setBounds(shell.getClientArea());
        shell.open();
        while(!shell.isDisposed()) {
           if(!display.readAndDispatch())
              display.sleep();
        }
        display.dispose();
   }
}
```

(listing 2)

```
// Démonstration d'événements typés

import org.eclipse.swt.*;
import org.eclipse.swt.widgets.*;
import org.eclipse.swt.events.*;
import org.eclipse.swt.graphics.*;


public class HelloEvent {
   public static void main(String[] args) {
              Display display = new Display();
              Shell shell = new Shell(display);


              shell.addPaintListener(new PaintListener() {
                   public void paintControl(PaintEvent event) {
                   String texte = "Programmez!";
                   Point size;
                   GC gc = event.gc;
                   Color rouge = new Color(event.widget.getDisplay(),
```

```
0xFF, 0, 0);
                gc.setForeground(rouge);
                Rectangle rect = ((Shell)event.widget).getClientArea();
                gc.drawRectangle(rect.x + 10,
rect.y + 10,
rect.width - 20,
rect.height - 20);
                Color bleu = new Color(event.widget.getDisplay(),
0x00, 0x00, 0xFF);
                Font font = new Font(event.widget.getDisplay(),
                    "Times New Roman", 24, SWT.ITALIC);
                gc.setForeground(bleu);
                gc.setFont(font);
                size = gc.stringExtent(texte);
                gc.drawString(texte,
(rect.width - size.x)/2,
(rect.height - size.y)/2);
                rouge.dispose();
                bleu.dispose();
                font.dispose();
            }
    });

    shell.addControlListener(new ControlAdapter() {
        public void controlResized(ControlEvent event) {
            String s = "";
            Rectangle rect = ((Shell)event.widget).getClientArea();
            s = s + "Largeur: " + rect.width;
            s = s + ", Hauteur: " + rect.height;
            System.out.println(s);
        }
    });

    shell.setBounds(0, 0, 300, 200);
    shell.open();
    while(!shell.isDisposed()) {
      if(!display.readAndDispatch())
        display.sleep();
    }
    display.dispose();
    }
}
```

(listing 3)

```
// Démonstration d'événements non typés

import org.eclipse.swt.*;
import org.eclipse.swt.widgets.*;
import org.eclipse.swt.graphics.*;

public class HelloEventSansType {
    public static void main(String[] args) {
        Display display = new Display();
        Shell shell = new Shell(display);

        shell.setBounds(0, 0, 300, 200);
        Listener listener = new Listener() {
          public void handleEvent(Event e) {
            switch (e.type) {
              case SWT.Resize:
                OnResize(e);
                break;
              case SWT.Paint:
                OnRepaint(e);
                break;
              default:
                System.out.println ("Reçu un événement inconnu");
            }
          }
        };
        shell.addListener (SWT.Resize, listener);
        shell.addListener (SWT.Paint, listener);

        shell.open();
    while(!shell.isDisposed()) {
      if(!display.readAndDispatch())
        display.sleep();
    }
    display.dispose();
    }

    // static, puisque invoquée depuis main qui est static
    private static void OnResize(Event event) {
            String s = "";
            Rectangle rect = ((Shell)event.widget).getClientArea();
            s = s + "Largeur: " + rect.width;
            s = s + ", Hauteur: " + rect.height;
            System.out.println(s);
    }

// static puisque invoquée depuis main qui est static
    private static void OnRepaint(Event event) {
        String texte = "Programmez!";
        Point size;
        GC gc = event.gc;
        Color bleu = new Color(event.widget.getDisplay(), 0x00, 0x00, 0xFF);
        Font font = new Font(event.widget.getDisplay(),
            "Times New Roman", 24, SWT.ITALIC);
        gc.setForeground(bleu);
        gc.setFont(font);
        size = gc.stringExtent(texte);
        Rectangle rect = ((Shell)event.widget).getClientArea();
        gc.drawString(texte, (rect.width - size.x)/2,
(rect.height - size.y)/2);
        bleu.dispose();
        font.dispose();
    }
}
```

Générer automatiquement des contrats WSDL avec PEAR::SOAP

La nouvelle version de ce package PEAR permet de générer des contrats WSDL et DISCO soit statiquement, soit à la volée.

Les couches SOAP utilisées avec PHP (PEAR::SOAP et NuSOAP) fonctionnent très bien, mais présentaient jusqu'à il y a encore peu un très gros inconvénient : elles n'étaient pas capables de générer automatiquement le contrat WSDL associé à un service web. D'une part on devait se coltiner à la main sa rédaction (une tâche lourde et fastidieuse), d'autre part il y avait un risque de désynchronisation entre l'API exposée dans le contrat WSDL et le service SOAP appelé. Ce problème est en passe d'être résolu avec la version 0.73 de PEAR::SOAP.
Avant de vous lancer dans la découverte de cette nouvelle fonctionnalité magique, commencez par vérifier que le package PEAR::SOAP est bien installé sur votre machine en tapant l'instruction " C:\pear ". Si vous obtenez un message du type " 'PEAR' n'est pas reconnu en tant que commande interne ... vous devez alors l'installer. Si PEAR est présent sur votre machine, passez directement à l'étape 2.

1) Installer PEAR

Les dernières versions de PHP (≥ 4.3.1) sont livrées avec le script go-pear.bat qu'il suffit d'exécuter pour installer l'installeur de PEAR. Ce script permet de prendre en compte les spécificités de windows. Si toutefois il n'était pas présent, téléchargez le script go-pear.php sur http://go-pear.org puis placez ce fichier dans un répertoire sous la racine de votre serveur web. Exécuter ce script dans votre navigateur et suivez les instructions. PEAR devrait s'installer correctement. N'oubliez pas de double cliquer sur " PEAR_ENV.reg " situé à la racine de votre répertoire PHP pour modifier votre path. Vérifiez également dans php.ini la directive " include_path = ".;C:\PHP\pear\" " où C:\PHP est le répertoire qui contient votre installation de PHP. Vérifiez également " register_argc_argv = On " pour que CLI prenne en compte les arguments de la ligne de commande de PEAR. Vous pouvez maintenant vous lancer dans l'installation du package SOAP.

2) Installer PEAR::SOAP

Commencez par lister les packages présents à l'aide de l'instruction " C:\php\pear list " pour vérifier si PEAR::SOAP est installé ou pas. Depuis une installation fraîche vous devriez obtenir :

```
C:\PHP>pear list
INSTALLED PACKAGES:
===================
PACKAGE        VERSION STATE
Archive_Tar    1.0     stable
Console_Getopt 1.0     stable
DB             1.4.0   stable
Mail           1.0.2   stable
Net_SMTP       1.2.3   stable
Net_Socket     1.0.1   stable
PEAR           1.1     stable
PHPUnit        0.6.2   stable
XML_Parser     1.0.1   stable
XML_RPC        1.0.4   stable
```

Le package PEAR::SOAP n'est pas présent. Installez le, grâce à la commande " pear install soap ". Il est possible que certains packages sous-jacents ne soient pas présents. Vous obtiendrez alors un message du type :

```
C:\PHP>pear install soap
downloading SOAP-0.7.3.tgz ...
...done: 73,630 bytes
requires package `Mail_Mime'
(...)
SOAP: dependencies failed
```

Commencez alors par installer chaque package dépendant, puis lancez l'installation de PEAR::SOAP.

```
C:\PHP>pear install Mail_Mime
downloading Mail_Mime-1.2.1.tgz ...
...done: 15,268 bytes
install ok: Mail_Mime 1.2.1
(...)
C:\PHP>pear install soap
downloading SOAP-0.7.3.tgz ...
...done: 73,630 bytes
install ok: SOAP 0.7.3
```

3) Créer le service

Le service SOAP que nous allons créer est tout à fait standard : une simple classe qui se contente de renvoyer la chaîne de caractères fournie par le client. Le recours à une classe est obligatoire avec PEAR::SOAP. Une instance de cette classe est en effet créée par le serveur SOAP, quand le service est demandé par un client.
La première étape, consiste à inclure la classe server.php de PEAR::SOAP.

```
<?php /* service-demo.php */
require_once 'SOAP/Server.php';
```

Créez ensuite la classe et déclarez les variables utilisées par la classe. A l'exécution, le tableau $__dispatch_map contiendra les données du mapping que nous allons coder un peu plus loin. C'est cette variable qui sera interrogée par le serveur SOAP pour trouver la fonction correspondant au service SOAP appelé :

```
class echoStringService {
    var $__dispatch_map = array();
```

Créez ensuite la fonction qui va réaliser les traitements demandés :

```
    function echoStringFunction($inputString) {
        return '<b>*** '.$inputString.' *** </b>';
    }
```

Puis précisez au serveur SOAP quelle méthode (un service web peut exposer plusieurs méthodes) sont exposées par quel service. Ici nous définissons un service qui contiendra une seule méthode publique (présentée dans le contrat WSDL). echoStringService() est un proxy qui expose publiquement une méthode privée. Ce proxy précise au serveur SOAP quelle fonction il doit appeler pour réaliser le traitement (echoStringFunction) ainsi que les paramètres attendus en entrée (inputString) et fournis en sortie (outputString) par cette fonction. Optionnelle, cette étape permet d'affiner le contrat WSDL :

```
    function echoStringService() {
        $this->__dispatch_map['echoStringFunction'] =
            array('in' => array('inputString' => 'string'),
                  'out' => array('outputString' => 'string'),
                  );
    }
}
```

Nous allons maintenant créer le bout de code générique qui permettra de fournir soit le contrat WSDL, soit la réponse SOAP attendue par le client en fonction de la querystring. Si le flux entrant est un appel de service on retourne le message SOAP attendu :

```
<?php /* service-demo-controleur.php */
if (isset($_SERVER['REQUEST_METHOD']) && $_SERVER['REQUEST_
METHOD']=='POST') {
    $server->service($HTTP_RAW_POST_DATA);
```

Sinon, on crée une instance du serveur DISCO puis on renvoie, soit le contrat WSDL – si la querystring finit par "wsdl", sinon on renvoie le contrat DISCO qui présente le service :

```
} else {
    require_once 'SOAP/Disco.php';
    $disco = new SOAP_DISCO_Server($server,$service_function);
    header("Content-type: text/xml");
    if (isset($_SERVER['QUERY_STRING']) &&
      strcasecmp($_SERVER['QUERY_STRING'],'wsdl')==0) {
       echo $disco->getWSDL();
    } else {
       echo $disco->getDISCO();
    }
    exit;
}
?>
```

Notez que ce fichier est générique : on passe $server et $service_function en variable pour simplifier les développements et favoriser la réutilisabilité de ce contrôleur. Vous pouvez maintenant assembler l'en-

semble de ces éléments pour créer votre service. Commencez par instancier le serveur SOAP puis votre service :

```
/* suite du fichier service-demo.php */

$server = new SOAP_Server;
$service = new echoStringService();
```

Vous devez ensuite associer un namespace à votre fonction :

```
$server->addObjectMap($service,'urn:echoStringFunction');
```

Stockez ensuite dans une variable le nom de votre fonction, de façon à rendre générique le contrôleur qui vous includez juste après :

```
$service_function = 'echoStringFunction';
include 'service-demo-controleur.php';
?>
```

4) Créer le client

Vous n'avez plus qu'à tester le serveur en créant un client SOAP. Rien de particulier ici. On commence par récupérer le contrat WSDL généré à la volée par le serveur, puis on crée un proxy SOAP à partir du contrat et on appelle le service distant, via ce proxy ,en lui passant les paramètres attendus en entrée. Le serveur nous renvoie sa réponse dans la foulée (on est en RPC) que l'on affiche aussitôt.

```
<?php
/* service-client.php */
require_once('SOAP/Client.php');
$wsdl = new SOAP_WSDL('http://localhost/service-demo.php?wsdl');
$client = $wsdl->getProxy();
$response  = $client->echoStringFunction('Hello World');
echo $response;
?>
```

5) Tester la génération WSDL

Connectez-vous à l'adresse http://localhost/service-demo.php?wsdl pour tester la génération dynamique du contrat WSDL. Vous devez normalement obtenir un contrat du type :

```
<?xml version="1.0" ?>
- <definitions name="echoStringFunction" targetNamespace="urn:echo
StringFunction" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:
soap="http://schemas.xmlsoap.org/wsdl/soap/"  xmlns:tns="urn:echo
StringFunction"  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types xmlns="http://schemas.xmlsoap.org/wsdl/" />
- <portType name="echoStringFunctionPort">
- <operation name="echoStringFunction">
  <input message="tns:echoStringFunctionRequest" />
  <output message="tns:echoStringFunctionResponse" />
  </operation>
  </portType>
- <binding name="echoStringFunctionBinding" type="tns:echoString
FunctionPort">
   <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/
soap
/http/" />
- <operation name="echoStringFunction">
   <soap:operation soapAction="urn:echoStringFunction#echostring
service#echoStringFunction" />
- <input>
  <soap:body use="encoded" namespace="urn:echoStringFunction"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
- <output>
  <soap:body use="encoded" namespace="urn:echoStringFunction"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
  </operation>
  </binding>
- <service name="echoStringFunctionService">
  <documentation />
- <port name="echoStringFunctionPort" binding="tns:echoStringFunction
Binding">
  <soap:address location="http://localhost/service-demo.php" />
  </port>
  </service>
- <message name="echoStringFunctionRequest">
  <part name="inputString" type="xsd:string" />
  </message>
- <message name="echoStringFunctionResponse">
  <part name="outputString" type="xsd:string" />
  </message>
  </definitions>
```

Si vous ne saisissez rien derrière " ? " vous obtiendrez un contrat DISCO du type :

```
  <?xml version="1.0" ?>
-  <disco:discovery  xmlns:disco="http://schemas.xmlsoap.org/disco/"
xmlns:scl="http://schemas.xmlsoap.org/disco/scl/">
  <scl:contractRef ref="http://localhost/service-demo.php?wsdl" />
  </disco:discovery>
```

6) Générez statiquement un contrat WSDL

Si vous ne souhaitez pas alourdir les performances de votre service en créant systématiquement un contrat WSDL à la volée à chaque appel, vous pouvez soit utiliser un système de cache, soit créer statiquement le contrat WSDL. Utilisez pour cela genproxy.php situé sous " PEAR/SOAP/Tools ". Il suffit de taper la ligne de commande :

```
genproxy.php http://localhost/service-demo.php?wsdl >mon_contrat.wsdl
```

■ **David Thévenon - Arnaud Limbourg**
(pearfr.org et co-développeur de PEAR::SOAP)
et **Pierre-Alain Joye** *(pearfr.org)*

Pour aller plus loin

Generating Server WSDL with PEAR::SOAP
(et un système de cache en prime !)
http://www.phppatterns.com/index.php/article/articleview/54/1/2/

Références :

La doc PEAR en partie en français : http://pear.php.net/manual/fr/

Le site français de référence sur PEAR : http://www.pearfr.org

PEAR ::SOAP expliqué Shane Caraveo :
http://talks.php.net/show/soap-forumduphp-paris2002/

Un jeu de puissance 4 en Lisp

1ère partie

Un temps tombé dans l'oubli, LISP connaît aujourd'hui un regain de popularité. Programmez! vous propose de (re)découvrir ce formidable langage avec une série de trois articles aux termes desquels, nous aurons écrit un jeu de puissance 4 en LISP.

Lisp est un vénérable doyen du monde de l'informatique. Tout ce que ne sait pas faire Lisp reste à inventer. Le créateur de Lisp, John McCarthy a posé les premières bases de ce langage il y a 50 ans. Nul besoin de dire que McCarthy était fort en avance sur son temps à l'époque. Par exemple, alors que les programmes en assembleurs fuyaient comme des paniers percés, McCarthy dotait son Lisp du premier ramasse-miettes. Oui oui, 50 ans avant Java... :-) Au cours des cinq dernières décennies Lisp a bien souvent été incompris, critiqué, raillé, pour ne pas dire plus. Toujours est-il que Lisp est toujours là, qu'il est étonnamment moderne aujourd'hui encore, et parfaitement adapté à l'industrie. En fait Lisp connaît un net regain de popularité depuis quelque temps, principalement aux Etats Unis. En France aussi, ainsi France Telecom a récemment choisi LISP pour développer une application d'envergure : Recife (http://www.franz.com/success/customer_apps/telecom/francetelecom.lhtml). Nous vous proposons, au cours d'une série de trois articles de (re)découvrir Lisp de façon ludique, puisque nous allons construire ensemble un petit jeu de puissance 4. Après tout, Lisp est aussi connu comme le langage de l'intelligence artificielle. Notre propos n'est pas d'étudier Lisp de manière absconse, ni même très rigoureuse, mais de nous faire plaisir en prenant contact avec un des langages les plus géniaux qui soit.

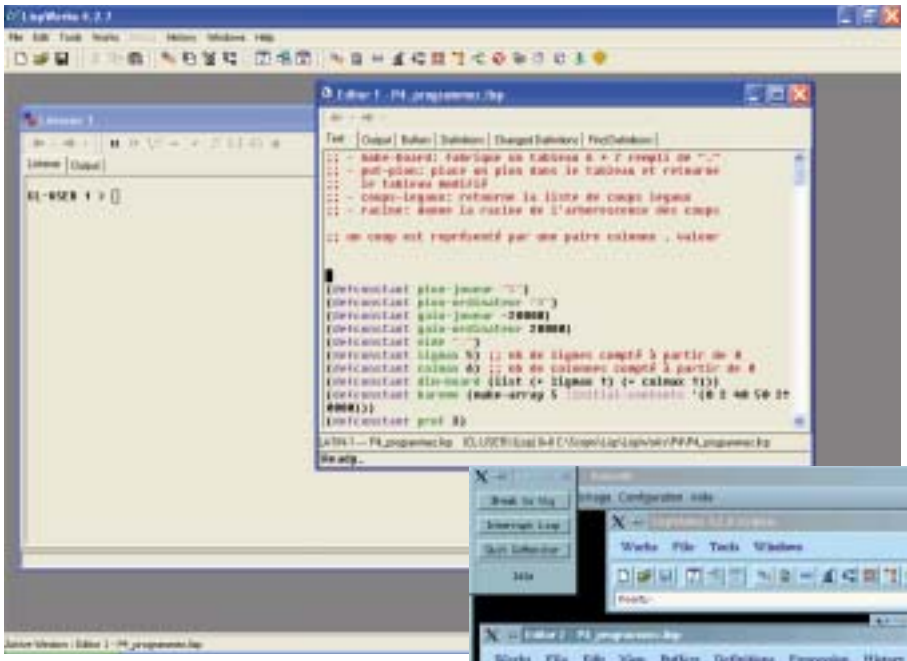


Figure 1 - *L'environnement Lispworks sous Windows.*

Lispworks

Savez vous qu'il existe des environnements de développement à la manière de C++ Builder, Delphi ou Visual Studio pour Lisp ? En effet Lisp permet de faire du Client/Serveur, du Web Service ou du CORBA tout aussi bien que d'autres langages plus populaires. Sans nous lancer dans pareille aventure, nous allons toutefois utiliser un de ces environnements dans sa version "Personnelle", largement suffisante pour notre propos. Je vous propose d'utiliser Lispworks qui présente l'avantage d'offrir la même interface sous Windows et sous Linux (figure 1 et 2). Vous pouvez télécharger le produit sur http://www.lispworks.com. Pour utiliser Lispworks sous Linux, vous devez veiller à ce que votre système dispose d'une librairie OpenMotif, de version supérieure ou égale à 2.1 ou encore la librairie Lesstif, si celle-ci est de version supérieure ou égale à 0.93. En outre, sous Linux, n'oubliez pas de lancer le script de licence avant même d'installer le paquetage rpm.

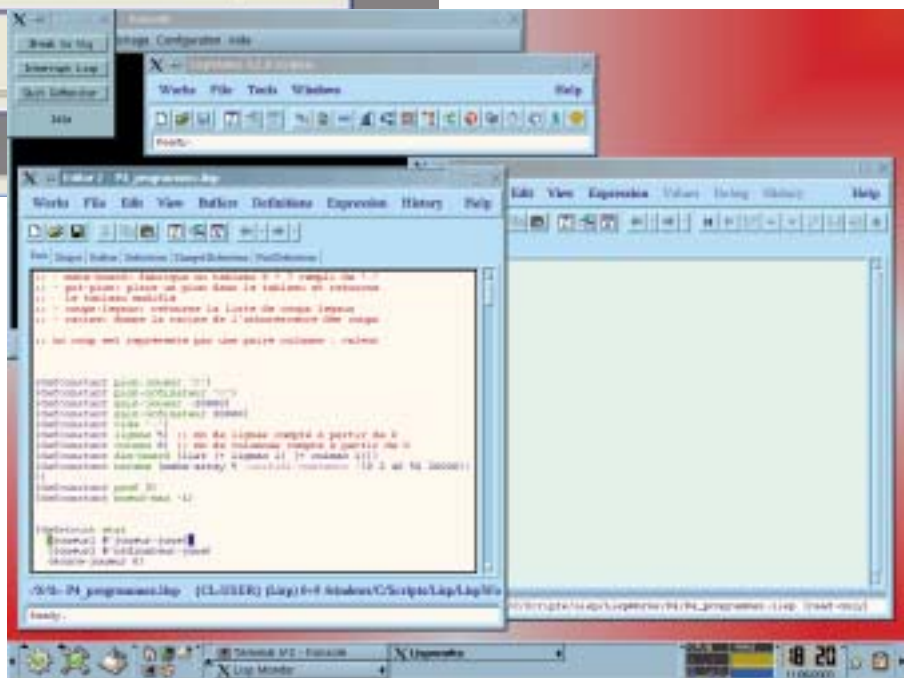


Figure 2 - *L'environnement LispWorks sous Linux.*

Prendre en main un environnement Lisp

Comme pour tous les environnements Lisp, l'interface de Lispworks n'est qu'accessoire par rapport au listener. Le listener (ou écouteur en français) est comparable à un terminal Unix ou une console Windows. Mais ici le système qui est derrière n'est ni Unix ni Windows, mais un système Lisp. On lance des programmes Lisp depuis le listener, on peut connaître l'état du système depuis le listener, on peut compiler du code depuis le listener, etc. Il est même

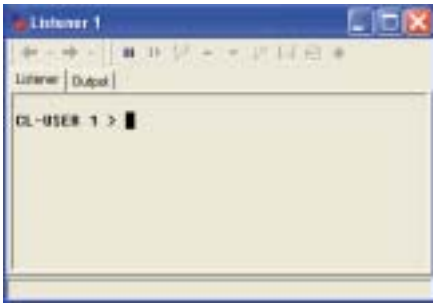


Figure 3 - *Le prompt du listener indique le package courant, qui est CL-USER par défaut.*

fréquent d'utiliser Lisp en production avec le listener, uniquement et rien d'autre.

Le Read-Eval-Loop

Le listener attend la saisie de code Lisp. Le code saisi est lu puis évalué, c'est à dire exécuté dans ce qui s'appelle la boucle de lecture évaluation (read-eval-loop). Lorsque le code évalué comporte par exemple des définitions de variables, celles-ci sont placées dans le package courant. Le package courant est toujours rappelé par le prompt du listener. Par défaut nous travaillons dans le package "CL-USER" (figure 3). En Lisp les packages définissent une portée de manière assez semblable au namespace de C++ par exemple. Nous travaillerons toujours dans le package par défaut pour ne pas compliquer les choses.

Lisp c'est la souplesse. Lorsque que du code est évalué dans le listener, il est converti en byte code à la manière de Java, afin que l'évaluation soit la plus rapide possible. Mais il est aussi possible de compiler le code. Dans ce cas, c'est une traduction en langage natif qui est effectuée. Alors les performances de Lisp sont irréprochables. Pas convaincu ? Lisp est réputé être lent dites vous ? Soit. Lorsque le code du Puissance 4 sera terminé, compilez le. Et portez le aussi vers un langage connu pour ses performances, C++ par exemple. Puis essayez les deux et comparez. Vous serez très surpris.

Ce qui est en outre amusant avec un système Lisp, c'est que byte-code et code machine peuvent cohabiter. Vous pouvez très bien évaluer un petit bout de programme modifié en byte code dans une application compilée. Mieux que cela, vous pouvez même faire vos modifications à chaud, c'est à dire pendant que votre application tourne.

L'éditeur

Bien évidemment, il ne viendrait à personne l'idée de saisir de longues portions de code dans le listener. Rien ne serait plus malaisé. Lispworks offre un éditeur à la Emacs, qui permet d'éditer plusieurs tampons ou 'buffers' à la fois. Les commandes sont grosso modo les mêmes qu'avec Emacs et parfaitement documentées. Pour entrer dans l'éditeur, il vous suffit d'actionner 'New' ou 'Open' depuis le menu file, ou encore en utilisant la barre d'outils. Vous pouvez alors taper votre code.

Du code!

Bon alors, à quoi ressemble-t-il ce fameux code Lisp ? Lisp est un acronyme construit sur LISte Processing, c'est à dire le traitement des listes. Une liste est un ensemble d'éléments. Les listes sont délimitées par des parenthèses. Ainsi :

```
(1 2 3)
```

est une liste de trois entiers, un deux et trois respectivement.

```
(1 "deux" 3)
```

est une liste constituée de deux entiers et d'une chaîne de caractères. Nous voyons ici que le type des éléments constitutifs de la liste n'est déclaré nulle part. En effet, Lisp est typé dynamiquement, comme Python par exemple.

Les listes que nous avons données plus haut n'ont pas d'autre intérêt que l'exemple, puisqu'on ne fait rien avec. Ce sont de simples listes de données. Un programme Lisp est en fait constitué de 'formes'. Une forme est une liste dont la tête est appliquée au reste de la liste. (Je simplifie, les puristes ne m'en voudront pas). Prenons un exemple. Nous voulons additionner les entiers 1 et 2. Il s'agit donc d'appliquer l'opération d'addition à la liste des entiers 1 et 2. La forme s'écrit ainsi :

```
(+ 1 2)
```

Saisissez cette forme dans le listener de Lispworks et validez en pressant la touche [return]. La forme est alors évaluée. Autrement

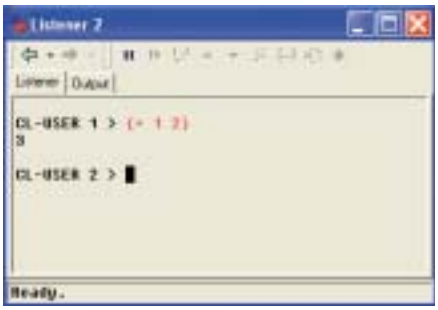


Figure 4 - *Ne croyez pas ce que racontent les mauvaises langues. Lisp sait compter.*

dit, Lisp fait le calcul et vous annonce fièrement que le résultat vaut 3 (figure 4).

Essayez encore :

```
(+ 1 2 3 4)
```

En cas d'erreur

Il se peut que vous vous trompiez en saisissant du code dans le listener. Celui-ci va immanquablement vous envoyer une volée de bois vert. Que faire alors ? Supposons, qu'un peu étourdi, vous ayez tapé :

```
(+ "un" 2)
```

Lisp ne sait pas faire cette opération, puisque les opérandes ne sont pas de mêmes types, et il vous le fait brutalement savoir comme le montre la figure 5. Le listener vous énumère des commandes possibles pour remédier au problème. Le plus simple pour nous est de revenir au niveau supérieur de la boucle d'évaluation en choisissant "Return to top loop level o". Pour ce faire, et dans ce cas précis tapez :

```
:c 4
```

dans le listener puis validez avec [return].

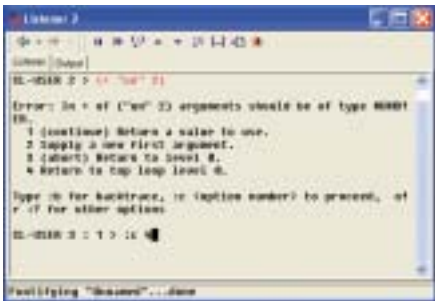


Figure 5 - *En cas de problème, savoir revenir au niveau supérieur de la boucle d'évaluation.*

Quelques autres exemples pour nous faire la main.

```
(> 3 2)
```

Lisp nous renvoie la valeur T pour true, car 3 est bien supérieur à 2 jusqu'à nouvel ordre.

```
(< 3 2)
```

Lisp nous renvoie NIL, ce qui dans son jargon veut dire false.

Définir une fonction

Supposons que nous voulions définir une fonction renvoyant la somme de deux nombres. Ceci se fait au moyen d'une forme spéciale. En vertu de la règle énoncée plus haut, on applique "l'opération" 'defun' pour définir une fonction à une liste contenant un nom, une liste d'arguments et une liste d'instructions qui sont elles mêmes des formes, bien entendu. Cela donne :

```
(defun somme (x y)
  (+ x y))
```

Evaluez ceci dans le listener, puis essayez:

```
(somme 10 20)
```

Lisp renvoie 30.

Oh là là toutes ces parenthèses!

Il est vrai que la syntaxe très particulière de Lisp est déstabilisante au premier abord. On ne s'y retrouve pas dans ces histoires de listes et de parenthèses. En fait le truc est tout simple. Un programmeur Lisp ne cherche surtout pas à s'y retrouver :-) Il se repose sur son éditeur Lisp qui indente automatiquement le code et met en surbrillance les couples de parenthèses. L'indentation du code est seulement conventionnelle mais tout le monde la respecte. Un codeur Lisp ne regarde jamais les parenthèses pour lire du code, mais seulement l'indentation.

Ecrivons notre fonction dans l'éditeur. Créez un nouveau document. Sauvegardez le sous le nom de essai.lisp. Tapez la première ligne de code

```
(defun somme (x y)
```

puis, avant d'entamer le seconde ligne, appuyez sur la touche TAB. Automatiquement, l'éditeur indente. Continuez :

```
(+ x y
```

Puis tapez autant de parenthèses fermantes

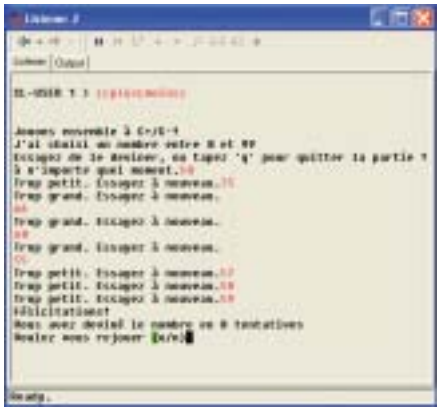


Figure 7 - *Une petite partie de C+/C- ?*

qu'il le faut pour que la surbrillance vienne sur la première parenthèse ouvrante avant le defun. C'est fini. Un codeur Lisp ne compte jamais les parenthèses.

Au départ, c'est étrange tout cela. Mais quand on a pris l'habitude, ce qui nécessite environ une heure ou deux, on ne voudrait plus coder avec autre chose que Lisp (si on n'a pas son Python sous la main :-).

Lisp est normalisé

C'est même sans doute le langage le "mieux normalisé" qui soit. Lispworks fournit une aide en ligne relative au ANSI Common Lisp Standard, qui n'est autre que le document cltl2 (Comme Lisp The Language, 2nd Edition) officiel. Dans cette documentation vous trouverez à chaque chapitre des exemples de codes correctement indentés qui pourront vous aider à vous familiariser avec le langage. Le mois prochain nous étudierons les fonctionnalités de Lisp que nous utiliserons pour coder notre jeu, puis le mois d'après, l'algorithme du jeu et son implémentation. En attendant je vous propose de vous familiariser avec Lisp au moyen du célèbre jeu C+/C- dont vous trouverez le code ci-contre et sur le Cd-Rom accompagnant la revue. Chargez le fichier CplusCMoins.lisp dans Lispworks, puis évaluez le buffer, soit depuis le menu :

➔ Work | Buffers | Evaluate

soit en cliquant dans le source avec le bouton droit puis en sélectionnant

➔ Buffers | Evalutate

dans le menu surgissant.

Puis lancez le programme en tapant dans le listener (figure 7) :

```
(cpluscmoins)
```

Je vous suggère d'étudier particulièrement les entrées/sorties, soit les fonctions read et format ainsi que la forme de control de flux d'exécution 'cond', omniprésente en Lisp. Bon Lisp et à bientôt.

■ **Frédéric Mazué** - *fmazue@programmez.com*

listing :

```
;;; Une implémentation du jeu C+/C- en Lisp

(defun GenererNombre()
   (random 100))


(defun CplusCmoins()
  (let ((nombre (GenererNombre)))
    (format t "~%~%Jouons ensemble à C+/C-!~%")
    (format t "J'ai choisi un nombre entre 0 et 99~%")
    (format t "Essayez de le deviner, ou tapez 'q' pour
quitter la partie à n'importe quel moment.")
    (Jouer nombre 1)))



(defun Jouer(nombre tentatives)
  (let ((essai (read)))
    (cond
      ((or (eq essai 'q) (eq essai 'Q))
       (format t "~%Vous avez quitté la partie.~%")
       (format t "Merci d'avoir joué avec moi.~%"))

      ((> essai nombre)
       (format t "Trop grand. Essayez à nouveau.~%")
       (Jouer nombre (+ 1 tentatives)))
      ((< essai nombre)
       (format t "Trop petit. Essayez à nouveau.")
       (Jouer nombre (+ 1 tentatives)))
      ((= essai nombre)
       (format t "Félicitations!~%")
       (format t "Vous avez deviné le nombre en ~d
tentatives~%" tentatives)
       (JouerEncore)))))

(defun JouerEncore()
  (format t "Voulez vous rejouer (o/n)")
  (let ((reponse (read)))
    (cond
      ((or (eq reponse 'O) (eq reponse 'o))
       (CplusCmoins))
      (t
           (format t "Merci d'avoir joué avec
moi.~%")))))
```

Programmation d'un jeu 3D

La technique de modélisation 3D d'un relief

2ème partie

La phase de modélisation 3D d'un jeu est souvent la phase la plus critique. Elle influera directement sur la fluidité du jeu. C'est également l'une des phases de conception les plus onéreuses car elle nécessite des compétences et des logiciels 3D adaptés. Cependant, il existe quelques techniques de modélisation 3D intermédiaires qui permettent de réduire les coûts et les délais de conception 3D.

La technique que nous utilisons pour notre jeu permet, par exemple, de créer le relief tridimensionnel d'un paysage virtuel. Nous allons étudier cette technique qui n'utilise pas de logiciel de modélisation 3D, comme 3D Studio Max ou Maya, mais deux simples images et quelques lignes de code.

Notre objectif, conformément au dossier de spécifications, est de générer une surface en relief, tout en intégrant (ou plutôt en incrustant) le circuit dans ce relief. Pour ce relief, nous allons utiliser un générateur de maille et une image bitmap *grayscale*, autrement dit une image avec des niveaux de gris. Mais avant de commencer, analysons le processus de mise en œuvre de cette technique.

Une recette de cuisine

Dans toute recette de cuisine, il existe une liste d'ingrédients. Et notre technique s'apparente également à une recette avec sa liste d'ingrédients. Fort heureusement, cette liste est courte, elle se limite à deux images au format bitmap.

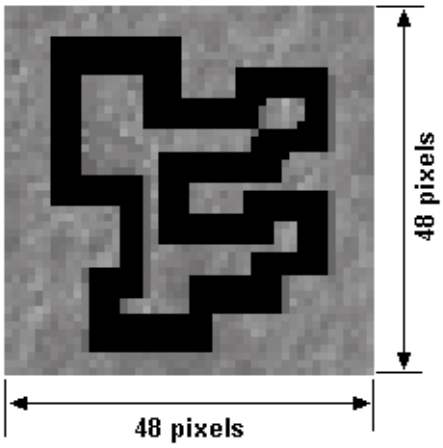


Figure n°1 - *Image grayscale de 48 pixels de large sur 48 pixels de hauteur. Cette faible résolution d'image nous permettra de générer un relief avec un nombre de polygones limités.*

Le premier ingrédient est une image bitmap 8 bits de type *grayscale*. Ce bitmap représente la surface du relief avec différentes teintes de gris plus ou moins foncés, d'où le nom anglais *grayscale*. L'image est volontairement de petite taille (figure n°1) afin de réduire le nombre de polygones du relief. En effet, nous verrons plus loin que plus cette image est grande et plus le nombre de polygones générés par notre technique est important. Le nombre de polygones est critique pour un moteur 3D car il influe directement sur la fluidité du jeu. La fluidité est un paramètre qui se mesure en FPS (frames per second ou images par seconde). Les meilleurs moteurs 3D tournent au delà de 60 FPS mais une valeur de 30 FPS est une limite acceptable. Pour un jeu de catégorie "web game" (jeux fonctionnant dans un navigateur Internet), il faut essayer de se limiter à 20 FPS minimum pour obtenir un bon niveau de jouabilité et de fluidité.

Vous pouvez distinguer sur la figure n°1 la forme du circuit en noir pur. Nous expliquerons l'intérêt de ces différentes teintes de gris un peu plus loin.

Le second et dernier ingrédient de notre recette est la texture globale du relief. Elle est, dans notre exemple, composée de la texture du circuit (une texture de type goudronné fera parfaitement l'affaire) et de la texture du sol environnant le circuit. Autrement dit, le sol du relief (figure n°2). L'idée derrière la réalisation de cette image, est de concevoir une vue "satellite" du circuit et de son environnement. Vous pouvez, à titre d'exemple, récupérer une photo satellite d'un circuit de Formule Un, produire son équivalent *grayscale* (la même image mais avec des teintes de gris et une faible résolution) et modéliser avec ces deux images un circuit F1 en 3D. Bien entendu, cette opération nécessite quelques ajustements mais le résultat est souvent intéressant.

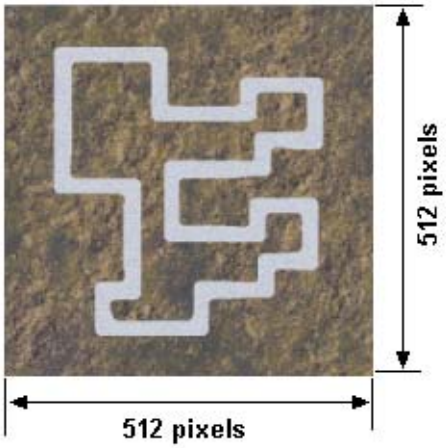


Figure n°2 - *La texture du relief est une image 32 bits 512x512*

L'image de la figure n°2 est donc la texture de la surface du relief de notre jeu. Cette image 32 bits a pour dimension 512 pixels par 512 pixels, mais elle peut être plus grande pour obtenir une meilleure précision de la texture. Il est important de conserver un même rapport de taille entre l'image *grayscale* et sa texture.

Reprenons à présent notre première image dite *grayscale*. La figure n°3 affiche quelques exemples de teintes de gris tirées de cette image, avec leurs différents codes couleur. Il existe en réalité beaucoup plus de teintes

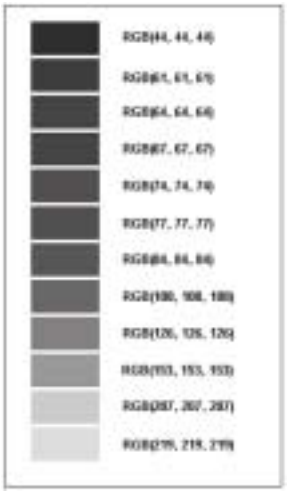


Figure n°3 - *Teintes grises de l' image grayscale et ses codes couleur*

dans cette image, mais nous n'en avons relevé que quelques unes à titre d'exemple. Notre image est en fait un bitmap 8 bits, avec une palette de couleurs grises. Notre objectif, en localisant les différentes teintes couleur de cette image, est de récupérer le code couleur, pixel par pixel, pour pouvoir déterminer une hauteur du relief par pixel (figure n°4).

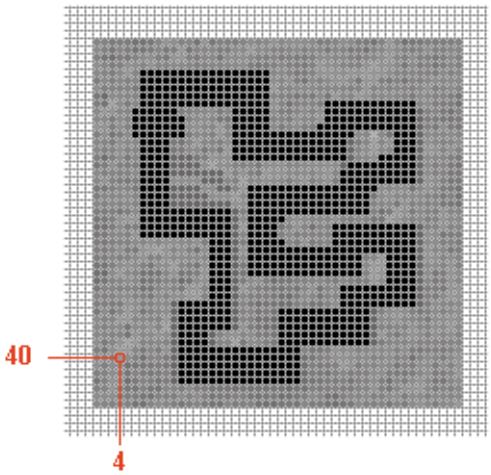


Figure n°4 - *Visualisation de l'image grayscale pixel par pixel*

Dans l'image de la figure n°4, nous définissons une grille virtuelle qui recouvre toute l'image. Chaque case de cette grille représente un pixel de l'image. Si nous localisons par exemple le pixel de coordonnées (4,40), nous pouvons déterminer son code couleur en valeur dite RGB (Red, Green, Blue) qui, dans le cadre d'une image *grayscale*, sera le code couleur d'une teinte de gris.

Au final, notre procédure de création de la surface se servira donc de la valeur du code couleur pour estimer la hauteur du relief. Par

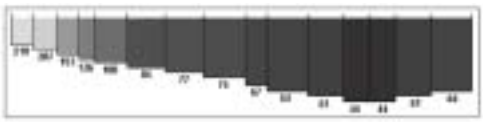


Figure n° 5 - *La hauteur du sommet est inversement proportionnelle à la valeur du code couleur*

exemple, si le code couleur du pixel de coordonnées (100,100) est égal à 108, la hauteur du relief en ce point sera inversement proportionnelle à 108. Si la valeur récupérée est 207, le relief sera inversement proportionnel à 207 soit une hauteur plus faible que la précédente. En conclusion, plus le pixel sera de teinte sombre, plus le code couleur de ce pixel sera élevé et plus la "hauteur" du sommet sera faible (figure n°5). C'est la raison pour laquelle le tracé du circuit est en noir pur, rgb(255,255,255), afin de retourner une hauteur de sommet la plus basse possible (coordonnées sur l'axe z de zéro) et donner l'impression ainsi que la piste est creusée dans le relief.

Analyse du code source

Le jeu que nous développons est un web game, un jeu dédié à Internet. Il sera aussi diffusé sur CD-Rom sous une forme optimisée, mais nous devons prendre en considération la démonstration qui fonctionnera directement dans un navigateur Web.

Le mois dernier, nous avons exploré les différentes technologies possibles pour la création et la diffusion de web games en 3D. Nous avons opté pour la technologie Shockwave 3D qui utilise l'environnement Director MX de Macromedia, un moteur 3D développé par Intel et le langage de script Lingo de Macromedia. Ce package apparaît aujourd'hui comme le mieux adapté pour ce type de réalisation avec l'environnement Virtools. Mais nous avons opté pour la solution Macromedia car les outils de développements sont accessibles en version de démonstration, contrairement à ceux de la société Virtools.

Le code en Lingo 3D

Le langage de référence utilisé pour notre exemple est donc le langage de script Lingo 3D. Ce langage est assez intuitif et complet. En termes de performance, il reste bien entendu en retrait par rapport à un langage comme le C++, mais il permet des délais de développement rapides, modulaires (orientés objet) et le moteur Shockwave 3D s'avère polyvalent. L'alternative Java 3D est intéressante, mais réaliser un jeu qui puisse fonctionner dans un navigateur Internet semble trop long avec ce langage.

Vous trouverez donc l'ensemble du code source de cet article dans le fichier relief_3D.dir (disponible sur le CD-Rom). Il vous faudra installer préalablement l'environnement de développement Director MX (version d'évaluation téléchargeable dans la section de téléchargement du site de Macromedia : http://www.macromedia.com/fr/). Cet environnement intègre le moteur 3D également.

Ouvrez donc le fichier relief_3D.dir et ouvrez l'acteur intitulé "Fonctions" (figure n°6).

Analysons à présent le code source de la procédure creationSurface(). Les premières ins-

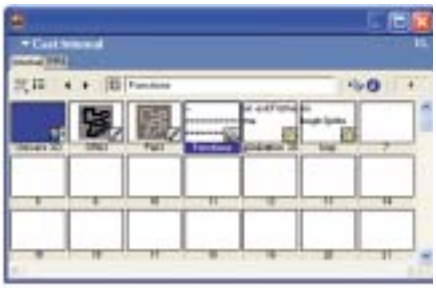


Figure n°6 - *L'acteur "Fonctions" intègre tout le code de génération du relief*

tructions permettent de sélectionner les images concernées, de définir la taille de la surface à explorer avec la variable dimSurface et de définir le degré du relief représenté avec la variable locale valeurHauteurs. Plus sa valeur sera élevée, moins l'altitude sera grande. Dans notre programme, l'image GR03 est l'image *grayscale*. L'image PL03 est la texture.

```
grayHauteursImage = member("GR03").image
surfaceTextureImage = member("PL03").image
dimSurface = 100
valeurHauteurs = 2
```

Le nombre de vertex (ou de sommets) est calculé en élevant au carré la valeur d'un côté de l'image GR03 (l'image GR03 étant un carré, sa surface se calcule en multipliant sa largeur par sa hauteur). Le résultat de cette opération sera enregistré dans la variable locale nbVertex. On calcule aussi le nombre de polygones et on affecte ce nombre dans la variable nbPoly (figure n°6).

```
nbVertex = grayHauteursImage.width * grayHauteursImage.width
nbPoly = 2*(grayHauteursImage.width -1)*(grayHauteursImage.width -1)
```

Nous connaissons maintenant un certain nombre d'informations permettant de définir le nombre de polygones et de vertex de notre relief. Pour rappel, le polygone est une unité de base en modélisation 3D. Il est constitué de 3 points, appelés aussi vertex (*vertices* au pluriel en anglais) ou bien "sommet" pour désigner le sommet d'un angle. Le polygone est donc un triangle qui constitue une surface d'un modèle 3D. La suite de notre fonction "tricote" un maillage de polygones, similaire dans l'absolu au maillage d'un tricot. Pour ce faire, nous utiliserons un générateur de maille. (figure n°7).

Utilisation d'un générateur de maille en Lingo 3D

Reste à présent à utiliser le générateur de maille pour créer les sommets de la surface. Le générateur de maille est la ressource de modèle 3D la plus complexe. Il permet de créer des géométries compliquées à l'exécution. La propriété type de primitive de générateur de maille est #mesh et est créée par la commande de newMesh. Les paramètres inclus avec cette commande définissent la taille de maille.

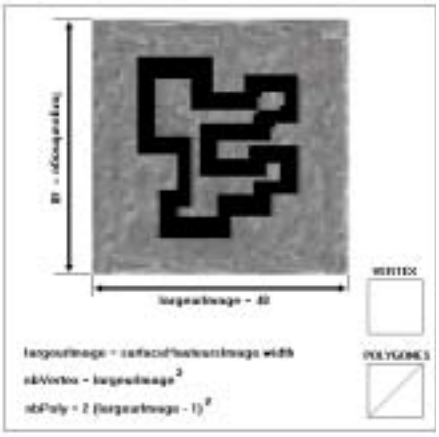


Figure n°7 - *Calcul du nombre de polygone du relief*

Vous pouvez utiliser le modificateur de déformation de maille pour manipuler la position des sommets à l'exécution pour #mesh ou tout autre type de ressource de modèle. Vous pouvez également utiliser la primitive #mesh pour modifier directement les propriétés de maille, bien qu'il ne s'agisse normalement pas d'une solution pratique dans la mesure où la maille doit être reconstruite après chaque modification. Nous utilisons donc la commande newMesh pour générer le terrain, avec comme paramètres, le nombre de polygones et le nombre de vertex. Le résultat est enregistré dans la variable terrainSurface.

```
terrainSurface = member("Univers 3D").newMesh("terrainSurface", nbPoly, nbVertex)
vertexList = []
tmp1 = grayHauteursImage.width / 2
tmp2 = grayHauteursImage.width - 1
repeat with n = 0 to tmp2
  repeat with m = 0 to tmp2
    p = (255 - grayHauteursImage.getPixel(m,n).paletteindex) / valeurHauteurs
      vertexlist.add(vector((m - tmp1) * dimSurface, (grayHauteursImage.width - n - tmp1) *
          dimSurface, p))
  end repeat
end repeat
```

Ensuite, nous utilisons la liste vertexList[] pour stocker les valeurs vectorielles pour chaque sommet de la maille. A l'aide de deux boucles repeat imbriquées (une boucle *repeat with* en Lingo est l'équivalent d'une boucle *for(;;)* en langage C). Nous parcourons ainsi tous les points de la surface *grayscale* et nous construisons une liste de valeurs permettant de définir le relief de cette surface. L'objectif est d'enregistrer le code couleur du pixel visé avec la fonction getPixel(X,Y). Cette valeur permet de définir la hauteur du sommet visé. Plus le code couleur du pixel traité est important, moins la hauteur sera élevée. Par exemple, un pixel blanc pur 8 bits retournera une valeur de 0, soit une altitude de (255-0)/10 = 25,5 unités. Au contraire, un pixel noir retournera une valeur de 255, soit une altitude nulle car (255-255)/10 = 0 unité.

```
terrainSurface.vertexList = vertexList
tmp = grayHauteursImage.width
surfaceCourante = 1
repeat with n = 1 to tmp2
  repeat with m = 1 to tmp2
    terrainSurface.face[surfaceCourante].vertices
= [m+(n)*tmp ,m+(n-1)*tmp+1, m+(n-1)*tmp]
    terrainSurface.face[surfaceCourante+1].vertices
= [m+(n)*tmp ,m+(n)*tmp+1, m+(n-1)*tmp+1]
    surfaceCourante = surfaceCourante + 2
  end repeat
end repeat
terrainSurface.generateNormals(#smooth)
terrainSurface.build()
```

Il reste maintenant à lier les vertex et les faces, avec l'aide de la propriété face[surface Courante].vertices qui est utilisée avec une ressource de modèle de type #mesh.

Nous utilisons ensuite la commande generateNormals(style) pour adoucir les faces d'une maille. Cette commande calcule les vecteurs de la normale pour chaque sommet de la maille. Si le paramètre style a pour valeur #smooth, chaque sommet ne reçoit qu'une seule normale, quel que soit le nombre de faces auquel il appartient, les trois sommets d'une face pouvant avoir différente normales. Chaque normale de sommet est la moyenne des normales de toutes les faces le partageant. L'utilisation du paramètre #smooth résulte en une apparence plus adoucie des faces d'une maille, à l'exception des bords extérieurs, qui restent nets (figures 8, 9 et 10). Pour finir, nous utilisons la commande build(). C'est une commande 3D qui construit une maille. Vous devrez utiliser la commande build() pour la construction initiale de la maille et après avoir utilisé la commande generateNormals().

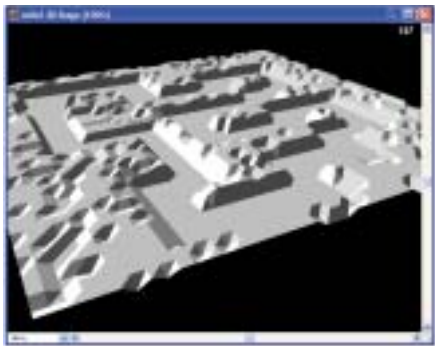


Figure n°8 - *Notre relief avec les vertex et les faces liés sans traitement. Les polygones anguleux sont particulièrement visibles, ils forment une surface peu esthétique et peu réaliste.*

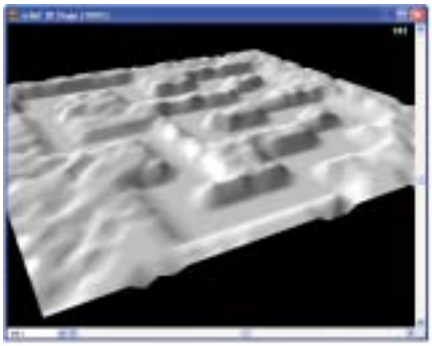


Figure n°9 - *Même surface avec traitement #smooth, la surface est à présent adoucie et plus réaliste*

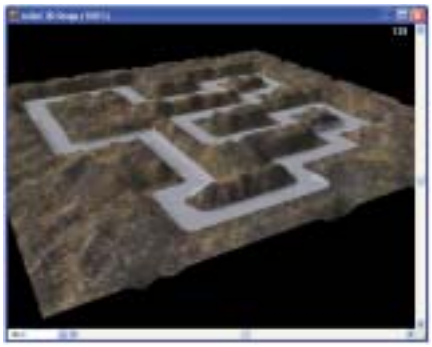


Figure n°10 - *Même surface avec application de la texture et de l' effet #smooth*

Application de la texture

La dernière étape de la construction de notre terrain est l'application de la texture et la défi-

Aparté 3D : la normale de sommet

Une normale de sommet est le vecteur de direction indiquant la direction "vers l'avant" d'un sommet. Si la normale de sommet pointe vers la caméra, les couleurs affichées dans la région de la maille contrôlée par cette normale sont déterminées par le matériau. Si la normale de sommet pointe en direction opposée à la caméra, la région de la maille contrôlée par cette normale ne sera pas visible.

nition du matériau. Nous utilisons ici la commande newTexture pour la définition de la texture. Ensuite, nous utilisons la propriété quality pour définir la qualité d'image d'une texture en contrôlant le niveau de *mipmapping* qui lui est appliqué.

La propriété de texture compressed qui indique si l'acteur source de la texture est compressé (TRUE) ou non (FALSE) est utilisée. La valeur de la propriété compressed passe automatiquement de TRUE à FALSE lorsque la texture est nécessaire pour le rendu. La valeur peut être FALSE pour décompresser la texture en avance. Elle peut recevoir TRUE pour annuler l'affectation de la représentation décompressée de la mémoire. La valeur par défaut de cette propriété est TRUE. La texture appliquée est bien sûr l'image bitmap 32 bits que nous appelons PL03 dans notre exemple.

```
textureSurface = member("Univers 3D").new
Texture("terrainSurfaceTex", #fromImageObject,

surfaceTextureImage)
materiauSurface = member("Univers 3D").new
Shader("materiauSurface", #standard)
materiauSurface.texture = textureSurface
materiauSurface.textureMode = #wrapPlanar
materiauSurface.shininess = 0
modeleRelief = member("Univers 3D").new
Model("relief_3D", terrainSurface)
modeleRelief.shader = materiauSurface
```

TextureMode est une autre propriété 3D de matériau #standard utilisée qui spécifie la façon dont la première couche de texture est plaquée sur la surface du modèle. Utilisez la propriété textureModeList pour spécifier les textures des couches autres que la première. L'option #wrapPlanar enrobe la surface du modèle avec la texture, comme s'il s'agissait d'une rétro projection.

Shininess est la dernière propriété de matériau standard que nous utilisons. Elle permet d'obtenir ou de définir le brillant d'une surface. Le brillant est défini en tant que pourcentage de la surface de matériau réservée aux tâches spéculaires. La valeur est un nombre entier compris entre 0 et 100.

Mise en œuvre

Vous savez à présent comment construire un relief en 3D avec deux images bitmap et quelques lignes de code. L'exemple qui vous est proposé ici est bien évidemment transposable sur n'importe quel autre langage de programmation ou moteur 3D. Les dernières lignes de code que nous décrirons en guise de conclusion vous permettront de positionner la caméra de l'application dans l'environnement 3D que vous venez de créer. La caméra est un concept virtuel bien connu des modeleurs 3D. Une caméra en 3D représente un point et un angle de vision. Elle symbolise en quelque sorte l'emplacement où se positionne l'œil de l'utilisateur, ainsi que son champ de vision. Vous trouverez le paramétrage initial dans l'acteur "loop" du fichier relief_3D.dir. Voici le code de ce paramétrage :

```
on beginSprite
  sprite(1).member.camera[1].worldPosition =
vector(0,-5000,2000)
  sprite(1).member.camera[1].rotate(60,0,0)
end
```

La commande qui permet de positionner la caméra sur trois axes se nomme worldPosition. Elle est définie sous la forme d'un vecteur à trois paramètres pour définir les coordonnées sur les trois axes. La commande rotate permet ensuite de définir un nouvel angle de vision en faisant simplement pivoter la caméra de quelques degrés. Dans notre exemple, nous avons donc opéré une simple translation de la caméra vers un point de coordonnée (0,-5000,2000) et une rotation de 60° sur l'axe X pour obtenir une bonne visualisation d'ensemble du relief généré et de sa perspective (figure n°10).

Pour finir, les lignes de codes suivantes permettent d'intégrer une boucle dans laquelle un processus de rotation du relief autour de l'axe Z s'opère à chaque frame :

```
on exitFrame me
  sprite(1).member.model[1].rotate(0,0,0.1)
  go to the frame
end
```

Conclusion de la seconde partie

Le générateur de maille est une fonction extrêmement utile pour la génération de surface. Mais notre méthode doit être utilisée avec parcimonie car elle génère un nombre important de polygones. N'oubliez pas qu'une application Director 3D est optimisée pour Internet et doit utiliser le minimum de surfaces possibles. De plus, la génération importante de polygones est susceptible de ralentir dramatiquement la vitesse d'exécution de votre application, surtout si un test de collision entre des modèles et la surface, est utilisé. C'est pourquoi, il est primordial de limiter les dimensions de l'acteur bitmap *grayscale* qui influera directement sur le nombre de polygones générés.

La prochaine fois, nous aborderons une autre technique de modélisation 3D qui nous permettra de créer un environnement tridimensionnel autour de notre relief. Cette technique reprend le concept de la technique d'environnement cubique qui porte aussi le nom de *skybox.* Nous présenterons ensuite le moteur de simulation physique qui nous permettra de concevoir toute la dynamique du jeu.

■ **Laurent Jayr** - *dev.net@skeddio.com*

Aparté 3D : le mipmapping

Le mipmapping est un processus qui crée des versions supplémentaires de l'image de texture, créées de tailles variées et plus petites que l'image d'origine. Le moteur 3D affiche ensuite à l'écran la version la plus appropriée de l'image en fonction de la taille courante du modèle et change la version de l'image utilisée selon les besoins. Le mipmapping trilinéaire produit une qualité supérieure au mipmapping bilinéaire mais demande aussi plus de mémoire. Le mipmapping est différent du filtrage, bien que les deux processus améliorent l'apparence de la texture. Le filtrage répartit les erreurs sur l'ensemble de la texture, pour qu'elles soient moins concentrées. Le mipmapping ré-échantillonne l'image pour lui donner la taille appropriée.