

IBM fait sa Java pour les entreprises

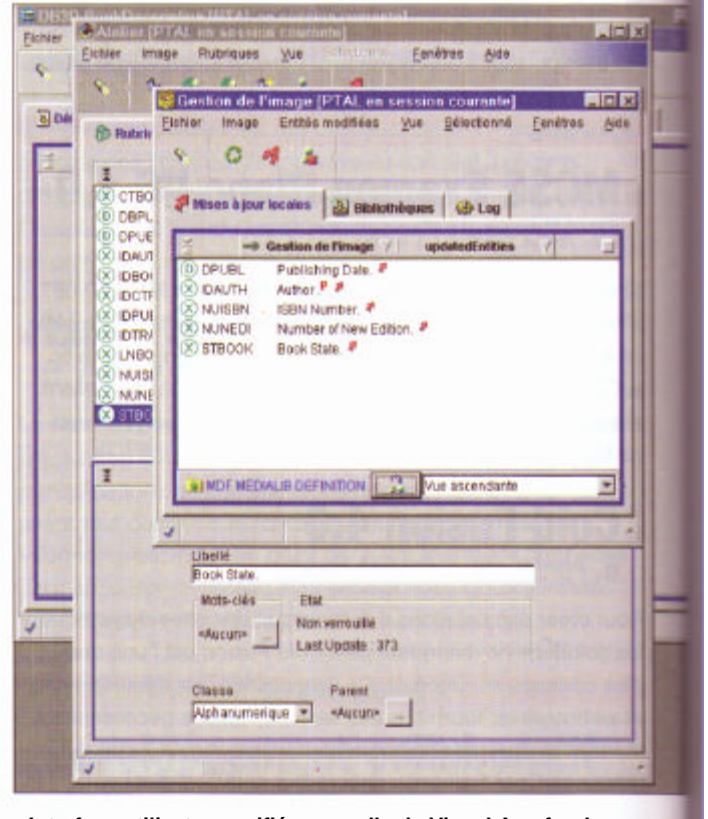
Les grandes et les très grandes entreprises utilisent bien souvent des systèmes *mainframes* et veulent continuer à les utiliser tout en migrant lentement vers les nouvelles technologies et les concepts de e-business. IBM étoffe sa gamme professionnelle avec une nouvelle mouture de *VisualAge Pacbase* version 3.0, disponible depuis la mi-juin.

Les développeurs et pauvres mortels que nous sommes, ne connaissent généralement que *VisualAge for Java*, le plus populaire des environnements de développement Java avec *JBuilder* d'Inprise. Or, *VisualAge* n'est que le cœur technique d'une gamme de logiciels, *VisualAge Pacbase*, *for Java* ou encore *Smalltalk*. *VisualAge Pacbase* est un environnement de modélisation et de conception qui évoquera bien des choses chez les responsables réseaux d'entreprises. Il s'agit d'un environnement de développement référentiel. *VisualAge Pacbase* sépare les phases de conception et de déploiement. Vous ne codez pas directement mais vous créez l'ossature (le référentiel du projet). C'est à partir de cette structure, que vous générerez vos applications suivant le système et le langage cible. Vous pourrez ainsi générer de l'ASP, JSP, Cobol ou encore Java, etc. Cet environnement est destiné aux grandes entreprises équipées de sites centraux. L'offre *VisualAge Pacbase* s'intègre, encore mieux, à la gamme IBM Software.

n'est finalement que l'environnement référentiel pour les composants serveurs et proxies. Il se mariera parfaitement avec le serveur *WebSphere Application Server*, le créateur *WebSphere Studio* et *VisualAge for Java*. La version 3 de *Pacbase* apporte deux grandes évolutions majeures : l'interface et Java. Si l'interface graphique existe depuis plusieurs années dans *Pacbase*, chaque produit *VisualAge* possédait la sienne propre. Désormais l'interface est unifiée et plus « moderne ». Sa caractéristique ? C'est une interface construite sur *Swing*, à l'instar d'un *JBuilder 3.5*. Les habitués des écrans textes vont être surpris. Java est la grande révolution de cette version.

EJB en standard

Outre l'interface, ce langage s'insinue en déploiement client, voire du côté serveur via les *servlets* dans les *proxies*, le e-business et avec les briques du support EJB (*Enterprise JavaBeans*), avec *EJB session*. Java est d'ailleurs mis en avant par rapport à Cobol, gardant



> Interface utilisateur unifiée sur celle de Visual Age for Java.

malgré tout, une place importante pour rester compatible avec les projets actuels. Nous avons évoqué la *proxy*. Sous *VisualAge Pacbase*, la *proxy* sert de lien entre le client et le serveur. La *proxy* est une simple *applet* ou une *servlet* Java. Cette ouverture vers Java a un double objectif : préparer les entreprises lorgnant sur les EJB et faciliter la prise en main pour les nouveaux développeurs, plus habitués à Java qu'à du Cobol. Sur les EJB, Ian Simpson (directeur du laboratoire IBM de Paris) et Philippe Schmitt (directeur de *marketing* IBM Software) en parlent comme d'une technologie indispensable à implementer, mais pour le moment, les grandes entreprises restent

plutôt observatrices et attentives que productives, peu de projets EJB se concrétisent encore. C'est en partie pour cela, que les EJB seront progressivement intégrées à *VisualAge Pacbase*. Une bonne manière pour laisser le temps aux EJB de s'imposer. L'aspect productivité n'a pas été oublié. Pour faciliter le travail de groupe et améliorer singulièrement le coût et le temps du développement, la station de travail (locale) a été revue et corrigée. Sous *Pacbase* le développement d'un projet est distribué entre le serveur *Pacbase* et les postes de développement. Pour développer, le programmeur passe par une station de travail rapatriée du serveur sur son

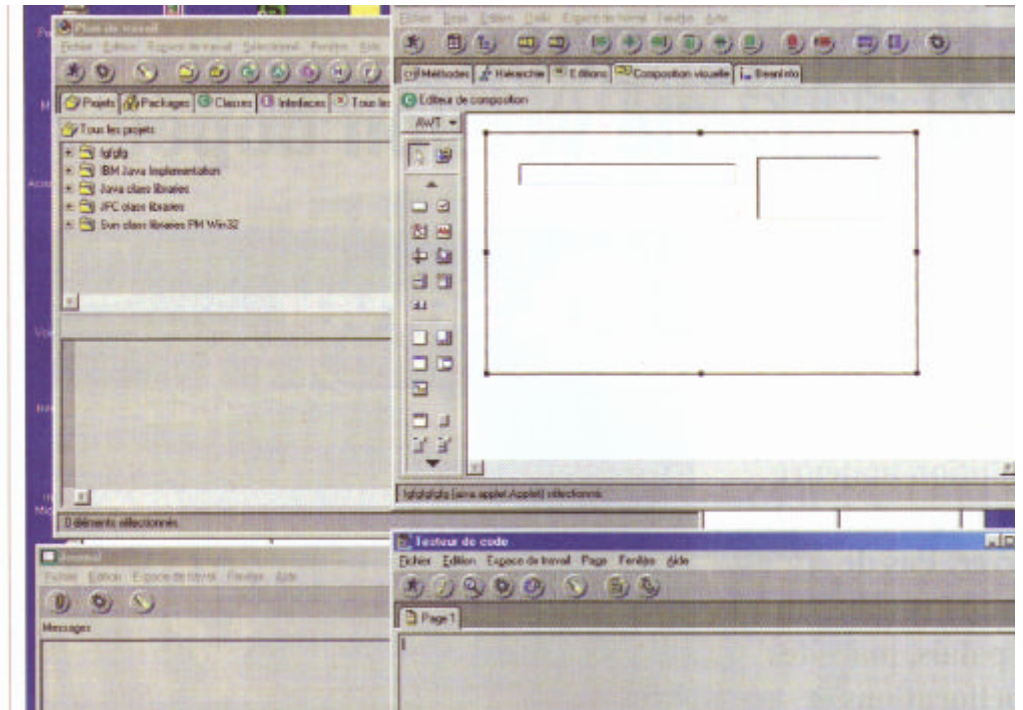
poste local. *Pacbase 3.0* améliore les échanges de données et leur synchronisation avec le serveur. Dans un souci de productivité toutes les données (du projet ou celles d'une base de données) sont immédiatement disponibles et mises sur le poste local. Grâce aux nouvelles fonctions dans le travail local et la nouvelle interface Java, IBM Software espère que le gain de temps (et aussi d'argent) sera d'environ 20 à 30 % par rapport à la version précédente de *Pacbase*. Argument de poids quand on sait que les coûts de développement ont eu tendance à augmenter avec le passage l'an 2000 et à l'Euro.

Et UML me direz-vous ? Pas d'inquiétudes ! Une passerelle est prévue vers *Rational Rose*, un des *leaders* dans la conceptualisation UML. Mais *Pacbase* s'enrichira sans doute d'un module interne UML dans les futures versions. Les anciens projets Cobol ne sont pas oubliés. On peut importer les projets déjà en production pour les adapter dans *Pacbase 3.0*. De nombreux tests ont été effectués durant la phase de développement pour assurer une transition en douceur. Comme *VisualAge Pacbase* est un outil hautement stratégique et critique pour les entreprises, le laboratoire IBM de Paris en charge de *Pacbase* travaille en très étroite collaboration avec les

clients, via le groupe des utilisateurs *Guépard*. Au sein de ce groupe, il existe un groupe *Évolution* où un responsable IBM est présent pour discuter des fonctions et futures évolutions. « Cela permet de coller aux attentes de nos clients », dicit Ian Simpson. La remontée des bugs se fait en grande partie par ce biais. Pour tous ceux qui craignent d'avoir un environnement fermé et non extensible, qu'ils soient rassurés ! Une API centrale est disponible. Cela permet de créer

des modules personnalisés pour ajouter des fonctions à *Pacbase*. Si cela est possible, IBM ne garantit pas la compatibilité des modules des anciennes versions. L'un des objectifs de *Pacbase* est de fournir la meilleure maintenance possible et de réduire au maximum les problèmes de bugs. Il intègre un module de tests des composants serveur pour traquer les erreurs et vérifier l'intégrité du projet à déployer. Un point que les entreprises apprécieront. Si

on ne devait retenir que les trois principaux points forts de *VisualAge Pacbase 3.0* ce serait : l'interface unifiée *Swing*, l'orientation *e-business* et l'administration centralisée et améliorée. Même si IBM se refuse à dévoiler tout chiffre de ventes sur *VisualAge Pacbase*, on sait malgré tout, que les ventes du projet permettent de financer développement et maintenance du projet dans le laboratoire IBM de Paris, ce qui n'est déjà pas si mal.



> Construction d'une applet sous Visual Age Pacbase.