

Une des difficultés du DSI et de son service informatique réside dans la gestion des applications et données, vieilles de 10, 15, 20 ou 25 ans. Ce patrimoine, souvent dénommé *legacy*, est parfois vécu comme un véritable boulet qu'il faut conserver, projet après projet. Quels sont les choix pour le DSI ?

Le patrimoine applicatif : statu quo ou évolution ?

Cinq milliards de nouvelles lignes de code Cobol chaque année

Dressons tout d'abord un constat : le patrimoine applicatif et de données représente un héritage très hétérogène, mainframe ou non, qui va du Cobol aux premiers Java en passant par VB, Smalltalk, Perl, Pac-Base, les outils et librairies maison non standard, etc. La documentation technique est de qualité variable, si elle existe. Il faut dire qu'il y a encore plus de 200 milliards de lignes de code Cobol. "Cinq milliards de nouvelles lignes de code Cobol se rajoutent chaque année !" commente **Henrik Jacobsen**, Directeur Technique de Micro Focus France.

Le poids du patrimoine applicatif se révèle par un chiffre simple : le coût de la maintenance de cet existant représente 70 à 80 % du budget IT (Forrester, 2005). Cela constitue une immobilisation des moyens humains et financiers considérable, qui grève l'investissement. On comprend mieux les enjeux colossaux du *legacy* et de sa modernisation. "Finalement, le patrimoine vieillit et même si on met en place une nouvelle application, celle-ci ne remplace pas forcément l'ancienne, qui perdure. La couverture (de la nouvelle application) n'est pas de 100 %. Cela ne simplifie pas le SI!", commente **Fabrice Bonan**, directeur R&D de Talend.

Legacy : source de TCO ?

Avec la crise, la pression se fait sentir pour diminuer le coût de production et de maintenance, et le *legacy* est directement montré du doigt. Comment, en effet, optimiser le coût de possession (ou TCO) ? Cela peut être un moyen de mieux aligner les besoins métiers et le IT, ou tout du moins de migrer une partie de son *legacy* dans de nouvelles applications, sans changer forcément le back office, mais en adaptant l'interface utilisateur. Libérer le mainframe de plusieurs applications peut aussi profiter en puissance et en disponibilité aux applications *legacy* restan-

tes, qui bénéficient des ressources machines rendues disponibles.

Les choix possibles de la modernisation

La modernisation peut se décliner, schématiquement, selon trois options :

- ❖ **statu quo** : on ne change rien. On continue la maintenance pour tenir la production
- ❖ **migration** : on migre tout ou partie d'une application *legacy*
- ❖ **approche mixte** : on modernise le frontal de l'application, ou on connecte le *legacy* à un nouveau frontal.

Il est essentiel de définir ce que l'on veut migrer ou moderniser, dans quels buts, et quel est l'objectif à long terme. Tout projet de modernisation (au sens large) doit impérativement avoir une valeur ajoutée, immédiate ou à long terme. Il y a dix ans le web-to-host fut un échec car la valeur ajoutée était quasi nulle.

Cobol en Open Source

Le marché reste dynamique, comme en témoignent ces projets annoncés en 2009. Ces initiatives Open Source pour le développement d'applications Cobol Grands Systèmes sous Eclipse, sont de véritable alternative aux solutions commerciales Mainframe existantes.

❖ **Metrixware** propose Cobos, un environnement de travail qui s'annonce ergonomique, complété d'un compilateur intégré et des outils d'analyse de la qualité et des services additionnels Metrixware. Dérivé du projet IDE Cobol, Cobos propose l'utilisation de l'environnement Eclipse pour le développement d'applications Cobol en local, avec une interface de communication basée sur SSH, limitant au strict nécessaire les échanges avec la plate-forme d'exécution. En natif, Cobos intègre un compilateur local, inspiré du projet OpenCobol, auquel

Compétence et fin de cycle : les pièges à déjouer !

L'un des plus graves problèmes du *legacy*, en particulier quand il est ancien, concerne la compétence. Si ce patrimoine a été codé avec un langage propriétaire et "mort" depuis des années, la compétence sur le code disparaît de l'entreprise. Pour éviter cela, il est vital de constituer une base de connaissances et d'entretenir la compétence en interne. Le Cobol est un exemple parmi d'autres. Aujourd'hui, les nouveaux développeurs formés au langage sont quasi inexistantes. Et avec les départs en retraite, ou départ du salarié pour diverses raisons, l'entreprise perd de la compétence Cobol. Le SI ne doit surtout pas laisser partir les compétences en outsourcing, sous peine d'abandonner la maîtrise de son *legacy*.

Nuançons tout de même cette affirmation. "Certaines écoles introduisent du Cobol (dans les cursus). La demande existe, sur Pac-base, par exemple, elle est assez forte. Mais

Metrixware ajoute notamment la génération sur Windows avec Visual Studio, le support de CICS, de DB2 et l'expansion des clauses copy Cobol.

❖ **COBOL-IT** est une société française d'édition de logiciel, créée en 2008, à l'initiative de **Stéphane Croce**. Elle est la première à proposer des services professionnels pour le monde COBOL en s'appuyant sur une gamme de logiciels Open Source. COBOL-IT Compiler Suite (compilateur et runtime COBOL) est une distribution Open Source d'un compilateur COBOL. L'éditeur vient d'ajouter à sa gamme un Atelier de développement basé sur ECLIPSE et un précompilateur dédié à la base de données MySQL. ■



Stéphane Croce
COBOL-IT

mettre ces technologies en avant dans une annonce, ce n'est pas 'vendeur'. Il y a danger pour les entreprises qui n'investissent pas dans la formation, cela fait le jeu de l'externalisation. Nous avons, chez Microfocus, mis en place des partenariats pour des cursus comme à l'IUT de Nantes. Il y a un réel besoin sur le marché, même si cela n'est pas toujours bien compris par les universitaires", explique Henrik Jacobsen. Une tendance croissante consiste à avoir la double compétence, Objet et legacy.

L'obsolescence des langages et des outils demeure un souci : de nombreux langages sont désormais des langues mortes et quantité d'outils arrivent, ou sont déjà arrivés, en fin de vie. Et le support de l'éditeur (s'il existe toujours !) n'est pas éternel. Le cas de Pacbase est éloquent, d'ici 6 ans, son support chez IBM s'arrêtera.

A chaque fin de cycle de vie d'un outil, d'un progiciel, le DSI doit prendre des décisions. Et à chaque nouvelle version, se poser des questions sur le support, sa durée et l'intérêt de garder une plate-forme en fin de vie.

La donnée legacy : un autre problème ?

Dans le legacy on parle plus volontiers des applications que des données. Or, la donnée patrimoine est cruciale pour l'activité d'une application critique et le business de l'entreprise. Pour assurer la reprise de la donnée legacy, la qualité est primordiale. Il faut auditer minutieusement les bases de données, la tables, les données, etc. avant de reprendre un patrimoine. La qualité de la donnée, établie il y a 10 ou 15 ans, n'est pas du même niveau que l'exigence d'aujourd'hui. "C'est une véritable enjeu. L'ésotérisme des formats, l'absence de documentation, la rigidité de la structure n'aident pas ! Il faut du pragmatisme" estime Fabrice Bonan.

L'erreur serait de partir tête baissée dans la migration des données, dans le nouveau SGBD. L'échec ne sera alors jamais bien loin ! La démarche progressive est conseillée, en évitant de tout migrer d'un coup, en procédant à une cartographie des données. La démarche demande du temps. L'usage d'un ETL peut "simplifier" la migration. Il est bien entendu possible de migrer directement d'un SGBD vers un autre mais la connaissance de la structure, ou au moins du format des données, est nécessaire. ■

François Tonic

Blu Age : le "modèle patrimoine"

L'éditeur français, connu pour son approche MDA et la génération automatique de code / des applications, s'attaque à la modernisation du patrimoine applicatif.

L'idée est de reprendre l'approche modélisation, MDA, pour générer sa nouvelle application à partir du modèle créé depuis son legacy. Comme le souligne **Christian Champagne**, Président Directeur Général, la question est de savoir comment extraire du patrimoine les informations nécessaires pour créer le modèle. "Avec Java, .Net, nous sommes dans de l'objet. Or, dans le patrimoine d'il y a 20-30 ans, il n'y a pas d'objet".

Moderniser les applications Pacbase

L'idée de Blu Age est de se placer au niveau du modèle métier. Comme nous sommes dans une approche MDA, l'objectif premier est de générer un PIM (Platform Independant Model).

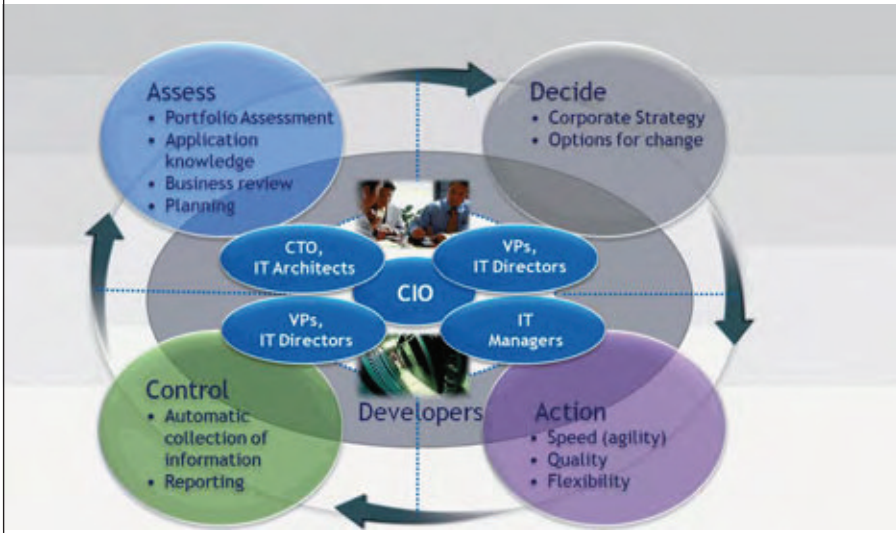
Ce PIM est créé grâce aux données, métadonnées, informations, que Blu Age récupère du legacy. A partir de là, l'environnement peut créer un PSM (Platform Specific Model) qui est la nouvelle implémentation applicative. L'éditeur met en avant la modernisation des applications Pacbase. Il s'agit de pouvoir récupérer le référentiel, le code nécessaire pour créer le modèle, puis générer une nouvelle application JEE.

Christian Champagne est très clair sur le niveau de reprise : "Sur Pacbase, c'est 100% sinon rien ! Il faut regarder le degré de pureté (du patrimoine). Pacbase est souvent bon mais le Cobol généré est illisible ! C'est une 'boîte noire'. Par contre, du legacy Java EE est plus simple." ■



C. Champagne

Modernisation et Gestion d'Applications d'Entreprise



Blu Age Legacy Modernization for zap

