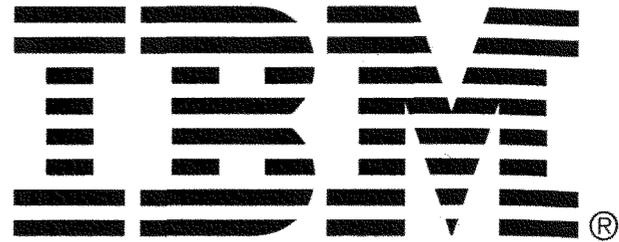




INTERVIEW



**Rémy Baranger**  
WebSphere Foundation  
& Tools and Linux,  
Market Manager



**Eric Charbonnier**  
WebSphere A/D Architect



**Jean-Francois LEVI**  
Responsable Architecture  
et Services du laboratoire  
WebSphere de Paris

■ **A quoi attribuez-vous l'arrivée récente sur le devant de la scène de la problématique d'intégration et de réutilisation de l'existant mainframe ?**

**IBM:** Nous assistons depuis quelque temps à un recentrage des grandes entreprises vers le mainframe, du fait du capital applicatif très important qu'ils hébergent. Le centre névralgique de ces grandes entreprises – banques, assurances, industrie – est constitué de plates-formes de type mainframe. Soumises à des contraintes d'ouverture et d'évolutivité, ces sociétés avaient deux possibilités, soit tout réécrire en J2EE, soit « transformer » l'existant. Cette deuxième voie est celle actuellement privilégiée.

La réécriture du système d'information, a souvent été écartée pour des raisons de coût élevé, associé aux risques importants qu'elle aurait fait courir à des données et processus vitaux pour l'entreprise. Par ailleurs, les sociétés concernées sont habituées à des systèmes d'in-

formation extrêmement centralisés et se sont heurtées au coût engendré par des applications distribuées sur différents serveurs.

Ceci étant, opposer J2EE et mainframe n'a pas de sens. Un mainframe sous ZOS ou Zlinux hébergeant un serveur Websphere offre un contexte d'exécution conforme à l'architecture J2EE et offre ainsi un système de production cohérent, autonome et efficace. Le mainframe, au-delà de la qualité de service intrinsèque qu'il offre, joue un rôle de consolidateur de serveurs. On se rapporte donc à un problème de TCO.

**« Le nouveau code coûte 5 fois plus qu'une réutilisation de code existant »**

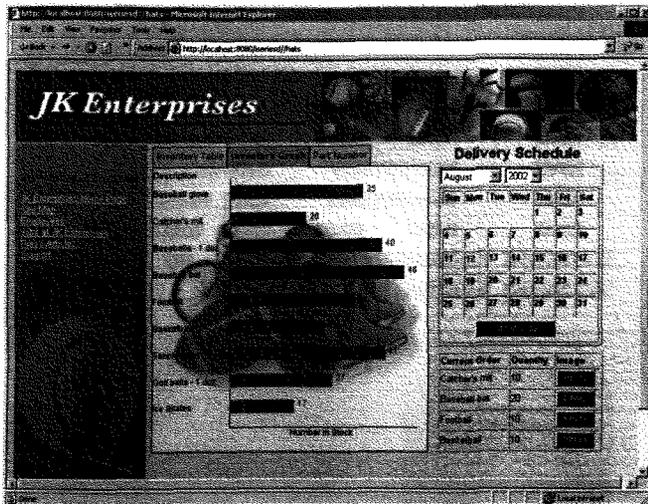
(Software Productivity Research)

**« La majorité des données sont toujours sur mainframes (zSeries, iSeries ou équiv), même si leur majorité est accessible par des applications Web et e-Commerce »**

(Don Greb, Mellon Financial Corp from Computerworld)

■ **Comment une entreprise peut-elle évoluer d'un existant COBOL monolithique à un système architecturé en composants ?**

**IBM:** Il convient tout d'abord d'identifier les applications candidates dont les fonctionnalités peuvent être exposées sous forme de composants. Le premier problème se situe au niveau du parc applicatif lui-même, dont les entreprises n'ont pas toujours une vision claire et exhaustive leur permettant de cerner précisément les aspects fonctionnels et



**Conversion d'écrans  
réalisée par IBM WebSphere Host Access Transformation Server.**

l'organisation en applications. Par ailleurs, les applications concernées ont été développées avec de multiples langages (COBOL, PL1, Assembleur...) et, dans la plupart des cas, à l'aide d'une vaste panoplie d'outils. Cette disparité place les entreprises face à un problème d'identification du périmètre technique des fonctionnalités.

La première étape consiste donc à découvrir ou redécouvrir le parc applicatif. Une bonne connaissance de l'existant est un préalable à tout passage vers des architectures à base de composants. Pour les applications, COBOL, PL1, Java, C, C++... nous proposons pour cela WebSphere Asset Analyzer. En ce qui concerne les applications VisualAge Packbase le problème ne se pose pas puisqu'un tel référentiel d'entreprise constitue déjà le cœur de l'outil.

La seconde étape consiste à identifier les composants. Certaines applications sont relativement bien écrites et donc faciles à migrer dans une architecture de composants. D'autres sont plus délicates, en particulier celles qui mélangent présentation, aspects fonctionnels et accès aux données.

■ **Une fois ces étapes franchies, à quels types d'intégration peut-on procéder ?**

**IBM:** Pour intégrer ces applications dans un environnement J2EE, il existe plusieurs approches.

La première est très proche du revamping. Elle consiste à intégrer les applications transactionnelles online dans des environnements J2EE en utilisant des javabeans J2EE aptes à consommer des écrans de type 3270, afin de récupérer les informations qu'ils exposent. La mise en œuvre permet de répondre rapidement à un besoin urgent d'ouverture du système d'information. L'architecture reste néanmoins centralisée et n'entraîne pas de séparation de la présentation et des éléments fonctionnels.

Une seconde technique plus ambitieuse implique de donner naissance à de véritables composants COBOL en

débarassant les éléments existants des traitements liés aux aspects de présentation. La plate-forme WebSphere Studio contient des outils permettant l'identification de ces éléments, la suppression des traitements de présentation et la régénération du code. Elle dispose également de javabeans pour l'exploitation à partir du monde Java des composants ainsi créés. Cette démarche est particulièrement intéressante dans le contexte de la problématique des services Web. Il ressort en effet qu'une architecture fondée sur des services d'entreprises correspond bien à cette identification et aux composants applicatifs réutilisables dans des architectures variées.

■ **Au-delà des problématiques techniques que vous évoquez, quels sont les autres aspects à considérer ?**

**IBM:** Il existe certains grands choix qui doivent être effectués par les entreprises et sur lesquels il n'existe pas d'unanimité quant à la meilleure voie à suivre. Ces orientations doivent être décidées relativement tôt dans la démarche.

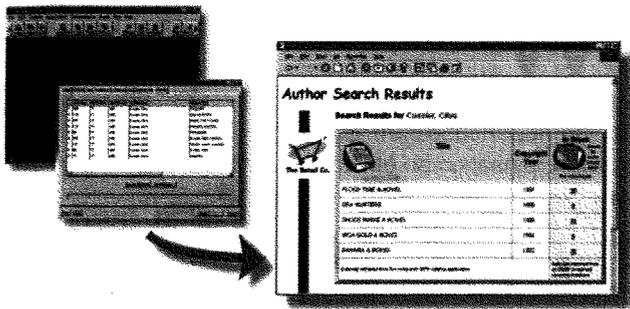
La gestion des transactions en est un exemple. Il convient par exemple de déterminer qui du serveur J2EE, de CICS ou d'IMS aura la responsabilité de piloter les transactions. Un autre aspect sur lequel il faut être vigilant est celui des performances. Le déploiement d'applications dans un environnement mettant en œuvre des serveurs J2EE et des plates-formes mainframe peut aboutir à des architectures qui ne supportent pas la charge escomptée. L'élaboration de prototypes est un moyen recommandé pour s'assurer de la montée en charge, de la sécurité et de l'ouverture vers l'extérieur des applications dans un tel environnement.

Les traitements batchs constituent une notion aujourd'hui totalement absente de l'architecture J2EE. Or, ce type de traitements représente pour certaines grandes entreprises jusqu'à 60 % du système d'information! Cet aspect est donc crucial dans une démarche d'évolution vers J2EE. C'est pourquoi nous avons commencé à la prendre en compte au travers de WebSphere et des évolutions que nous y apportons.

Il est donc essentiel que les composants issus de la phase de découverte et de réécriture puissent être utilisés dans le cadre de procédures batchs. Il en résulte que tant que les composants en question ne sauront pas gérer les volumes transactionnels ou les énormes mises à jour telles qu'elles sont pratiquées en batchs, les applications

**« Entre 850K et 1.3 millions de développeurs Cobol avec une érosion de la population de 12.000 par an »**

(IDC)



Conversion d'écran réalisée par IBM WebSphere Host Publisher.

dités traditionnelles auront de beaux jours devant elles. L'aspect humain est essentiel à la réussite de ce type de projet. Le transfert de compétences doit être assuré dans les deux sens. Les jeunes diplômés maîtrisant le monde Java et le monde distribué doivent s'approprier les grandes fonctions de l'application et comprendre la problématique d'intégration d'environnement plus centralisé de type mainframe. De leur côté, les développeurs traditionnels COBOL, PL1, ou autres doivent appréhender les implications des plates-formes J2EE, notamment en matière de sécurité, de performances, de distribution des traitements, de gestion cohérente des transactions... Certaines entreprises préfèrent conserver une spécialisation des individus. Cette politique nous semble assez dangereuse car elle crée des clans qui entravent la collaboration de l'ensemble des acteurs du développement et ne permet pas un véritable partage du système d'informations.

■ **Quel est le profil des individus aptes à procéder au redécoupage, à la réécriture en COBOL ou en Java et surtout au support des transactions distribuées ?**

**IBM:** Je ne pense pas que les aspects cruciaux que vous évoquez doivent être dévolus aux développeurs. La définition des normes d'utilisation de ce type d'environnement hybride doit impérativement être du ressort d'architectes. Ceux-ci doivent par ailleurs s'assurer de manière intransigeante que les applications produites par les développeurs respectent les standards fixés. Ce travail amont d'architecture est absolument essentiel. Il doit être effectué de manière méthodique afin de s'assurer de la fiabilité, de la flexibilité et de l'évolutivité de l'architecture retenue. Une fois cette architecture clairement définie, le plus difficile est fait. À partir de là, les aspects fonctionnels ainsi que la collaboration des équipes Java et mainframe deviennent l'aspect dominant.

■ **Développer des prototypes comme vous le préconisez laisse sous-entendre que vous manquez de repères...**

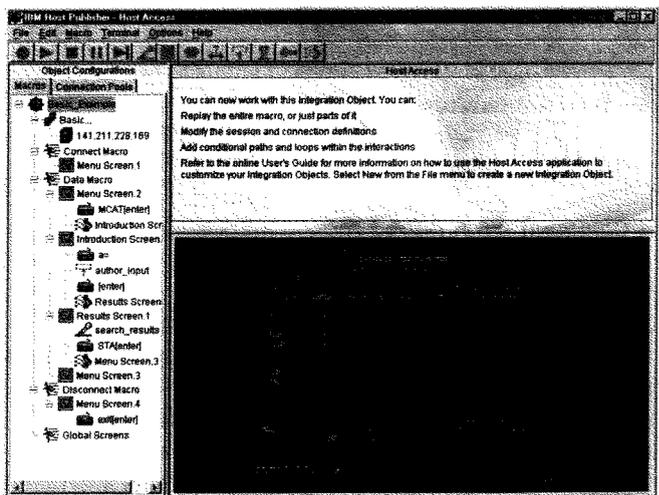
**IBM:** Le travail de prototypage est autant nécessaire pour les architectures full J2EE que pour les architectures de type hybride. Il est certain que les performances dépen-

dent en partie du serveur d'application. Mais c'est surtout dépendant de la qualité du développement et essentiellement des choix d'architecture. D'où le besoin de prototypage que ce soit en full J2EE ou en architecture hybride.

■ **En conclusion, peut-on aujourd'hui dégager une architecture hybride particulière ?**

**IBM:** A l'heure actuelle, la quasi-totalité des architectures hybrides est constituée d'un serveur J2EE placé en frontal de services métier exécutés sur le mainframe. Ces serveurs J2EE n'embarquent que très peu de logique transactionnelle afin de faciliter au maximum la réutilisation de l'existant.

L'évolution prévue est de montrer qu'il est possible de faire beaucoup plus avec Websphere par l'intermédiaire d'architectures orientées services. C'est pour cette raison qu'IBM parle aujourd'hui de serveurs d'intégration et non pas de serveurs d'applications.



L'outil de conversion d'IBM WebSphere Host Publisher.

Il n'existe encore que très peu d'architectures de ce type en production. Avec les récentes évolutions sur les services Web, nous proposons une première approche permettant de traiter les transactions sous forme de Services Web. En particulier, nous avons implémenté dans la version la plus récente de Websphere Enterprise, les propositions des couches basées sur WS Security et WS Transaction. Celles-ci vont permettre de réaliser concrètement des architectures au sein desquelles l'application est pour partie écrite en Java et pour partie en COBOL, le tout encapsulé dans un processus Websphere exécuté dans l'environnement mainframe. Le processus en question sera piloté par notre moteur de microflow/macroflow issu de la technologie Dragonfly. Il permettra l'enchaînement de services avec tout ce que cela comporte en matière de transactions distribuées notamment. ■