

BEA veut mener l'informatique au 'Truly Business Oriented'...

Que signifie le fait que BEA dispose d'une offre SOA « complète » ? Il y a plus de deux ans, nous avons beaucoup misé sur le concept de SOA (Service Oriented Architecture ou architecture orientée technique). C'est pourquoi aujourd'hui, tous nos produits suivent cette logique et intègrent ces mécanismes. Difficile pourtant de prétendre que nous disposons d'une offre SOA complète. En effet, qu'est-ce que cela pourrait signifier, alors que les définitions en la matière restent très ouvertes ? Aujourd'hui SOA est essentiellement perçue comme une technologie. Pourtant, cela va bien au-delà. Face aux problèmes applicatifs, de sécurité, de qualité de service ? SOA facilite le déploiement, le contrôle, et l'administration de tous les composants du système informatique. Néanmoins, l'intégration et le rôle de ces technologies varient selon qu'il s'agit d'un environnement mainframe, d'écriture de nouveau code, ou de solutions pré-packagées comme les progiciels de gestion intégrés (ERP) ou de gestion de la relation client (CRM). Certes, nous disposons d'un ensemble de solutions pour adresser les divers besoins en services Web. Cependant, les concepts SOA nous amènent à modifier notre façon de penser aux solutions pour résoudre des problèmes. Parmi ces spécificités, on notera en premier lieu la fédération des besoins, comme la prise en compte globale de la sécurité. Cela va d'ailleurs dans le sens d'une meilleure maîtrise des processus qui commence à apparaître avec les offres de management de l'activité, les solutions BxM (BPM, BAM?) Les applications mainframe restent nombreuses. Nécessitent-elles réellement une réécriture pour devenir des services Web ? Une très large partie des applications mainframes existantes ne seront pas réécrites, et actuellement les migrations concernent surtout les bases de données. Les services Web interviendront donc essentiellement sous forme d'encapsulation et de présentation, à l'image du revamping des écrans mainframe des années 90. Ainsi, les fonctions de messaging, du type MQSeries, intégreront SOA, bien que le code des applicatifs y recourant ne soit nullement modifié. D'ailleurs, cela se révèle très pertinent. En effet, les applications mainframes ont été pensées et conçues à l'origine pour ce type d'environnement et n'ont pas vocation à être migrées en services Web. Il suffit d'étudier quelles fonctions ont vocation à être ?exposées? comme en service Web et celles pour lesquelles ces adaptations sont inutiles. La question n'est pas ?Combien de services Web écrivons ou réécrivons-nous ??, mais plutôt ?Quelle est la pertinence de réécrire cette fonction selon son taux d'utilisation réelle ?.? Comment la migration vers les services Web répond-elle aux attentes de cohérence informatique des entreprises ? L'objectif du directeur informatique ne consiste pas faire migrer tout le système informatique vers les services Web, mais à envisager une migration pertinente des fonctions qui justifient cette évolution. On peut penser à des fonctions transversales concernant plusieurs services et nécessitant cohérence et administration avancée, comme le paiement ou la facturation. L'architecture SOA simplifie alors le déploiement, l'administration et surtout la conception de ces fonctions de plus en plus complexes avec l'ouverture des systèmes. Et surtout, cela modifie notre façon de penser en poussant la réflexion vers le ?Truly Business Oriented?. Cette interaction entre les lignes d'activité et les différents métiers de l'entreprise est primordiale. Avec SOA, le système d'information pourra assurer que les multiples utilisateurs utilisent la même version d'un service Web, y compris s'ils utilisent des versions différentes des applications y faisant appel. Une réalité souvent constatée sur le terrain. On retrouve l'image d'un jeu d'engrenage où le changement d'une pièce ne doit pas

interrompre le fonctionnement général du système, ni le résultat attendu. Java, .NET, Eclipse, comment pouvez-vous assurer l'homogénéité de la production sous autant de pseudo-standards et d'environnements ? Dans ma longue vie de technicien informatique, j'ai vu passer de nombreux standards magnifiquement conçus et qui ont rejoint le cimetière des belles idées. Ce sont le terrain et les utilisateurs qui finissent par établir les standards, et l'industrie s'aligne à cette demande. Sur les développements, nous nous devons de supporter aussi bien .NET que Java. Sur le lien entre Eclipse et .NET (avec ou non une interface commune Java et .NET, etc.), nous feront nos choix en fonction des demandes des développeurs. Selon les retours de terrain des directeurs informatiques, les types de développeurs sont apparemment différents et développent avec des outils et des pratiques distincts. Il n'y a pas forcément de nécessité d'homogénéiser les interfaces. Nous souhaitons que WebLogic reste la plate-forme de référence aussi bien pour .NET que pour Java.