

Introduction à l'étude de cas et à J2EE

En guise d'introduction, nous présenterons le contexte de notre étude de cas : l'entreprise virtuelle BlueWeb, son projet et son équipe de développement. Ce chapitre sera aussi prétexte à une justification des technologies J2EE au regard de technologies plus anciennes.

SOMMAIRE

- ▶ BlueWeb : une société virtuelle
- ▶ L'équipe de développement
- ▶ Technologies et méthodologie
- ▶ L'application de l'étude de cas

MOTS-CLÉS

- ▶ Projet pilote
- ▶ Gestionnaire de signets

Ce chapitre décrit une entreprise n'ayant d'existence que le long de cet ouvrage...

/// JSP (Java Server Pages)

Technologie permettant de définir des pages web créant du contenu dynamique. Cette API fait partie de celles regroupées au sein des spécifications J2EE.

/// Servlet

Composant web géré par un container et permettant la génération de contenu dynamique. Le chapitre 4 présente cette notion et signale des sites web ou des ouvrages de référence sur ce sujet. Cette API fait également partie des spécifications J2EE.

BlueWeb : l'entreprise

BlueWeb est une entreprise spécialisée dans la conception de sites Internet (intranet) clés en main pour des clients importants (banques, industries et administrations). Pionnière du Web en France, elle doit dorénavant opérer un tournant clé de son histoire et fournir des solutions complexes (techniquement) en vue de réaliser des portails en intranet/extranet fédérant diverses applications (solutions proches de l'EAI).

/// EAI (Enterprise Application Integration)

Nom désignant les techniques qui permettent d'intégrer différentes applications développées sans connaissance les unes des autres. Il s'agit donc d'utiliser des passerelles standards, telles que XML ou le protocole HTTP, pour transformer en un tout cohérent une série d'applications hétérogènes.

L'entreprise se prépare donc à faire un véritable saut technologique en abandonnant les compétences maintenant obsolètes qui avaient fait sa renommée :

- HTML ;
- scripts JavaScript ;
- scripts CGI, peu réutilisables et peu sûrs.

HTML et JavaScript ne sont pas obsolètes dans le sens où toutes les pages web du monde sont codées en HTML et souvent enrichies par des scripts JavaScript, mais ces seules technologies ne peuvent plus répondre aux besoins des applications d'aujourd'hui. C'est pourquoi, via des technologies comme les JSP/servlets ou les ASP dans le monde Microsoft, les pages HTML sont bien plus souvent dynamiques (produites dynamiquement par des programmes) que statiques (réalisées directement avec un éditeur de texte ou un outil dédié, comme Dreamweaver ou Quanta+).

L'exploitation de formulaires et l'envoi de courriers électroniques font partie de ces tâches autrefois réservées aux scripts CGI (codés en C ou en Perl par exemple) mais qui, pour des raisons de montée en charge, de réutilisation et aussi de sécurité, font maintenant les beaux jours des JSP/servlets. En fait, J2EE (voir encadré) apporte aux entreprises les avantages liés à toute technologie objet par rapport à des langages procéduraux comme le C, tels que la réutilisation et la facilité de maintenance. De plus, de par sa conception, Java est un langage sûr et portable. Ceci résout un des problèmes liés à l'utilisation des CGI : les failles de sécurité (par exemple les « buffer overflow », résultats d'une mauvaise conception, qui permettent à des programmeurs rusés d'introduire et de lancer du code indésirable sur le serveur).

/// EJB (Enterprise JavaBeans)

Nom d'une spécification de Sun (actuellement en version 2.1) qui vise à définir une bibliothèque permettant d'obtenir des composants métier côté serveur. Ces composants sont déployés au sein d'un conteneur pouvant leur fournir divers services de haut niveau comme la prise en charge de la persistance ou celle des transactions.

B.A.-BA Qu'est-ce que J2EE ?

J2EE (Java 2 Enterprise Edition) regroupe à la fois un ensemble de bibliothèques (servlets/EJB/JSP) et une plate-forme logicielle. C'est une version de la plate-forme Java spécialement étudiée pour les problèmes des applications d'entreprise. Elle fournit de nombreuses solutions au niveau de la gestion des problèmes récurrents dans ce domaine : sécurité (API JAAS), prise en charge des transactions (API JTA et gestion déclarative dans les EJB). JDBC fait aussi partie de ces technologies. Actuellement, la dernière version des spécifications disponible est la 1.4.

Cependant, avant de migrer toutes ses équipes de développement vers J2EE, BlueWeb veut utiliser ces technologies sur un projet test afin d'en retirer une expérience capitalisable pour son futur. Pour cela, quelques développeurs de l'équipe Recherche & Développement ont été choisis.

L'équipe de développement

L'équipe de développement choisie est composée de 5 personnes :

- Bob, le chef de projet, est un programmeur C émérite et, via son expérience du Perl, il a acquis une expertise dans le maniement des expressions régulières. Son bon sens et son pragmatisme sont les garde-fous de BlueWeb depuis des années. Le monde de J2EE, et en particulier la foule de projets issus de l'Open Source, le fascinent...
- Yann est un ancien stagiaire fraîchement sorti de l'école. Il est tombé dans l'objet quand il était tout petit et « parle » très bien UML et Java. Il va être en charge de la partie cliente et sera associé à Steve, qui devra assurer le déploiement/packaging de l'application.
- Pat est un spécialiste des bases de données. Programmeur C et DBA certifié Oracle et Sybase, il a le profil type du codeur côté serveur.
- Comme de coutume pour les projets au sein de BlueWeb, l'équipe de développement devra soumettre code et documentation à l'approbation du responsable qualité de la société : Michel, homme rigoureux entre tous.

À LA LOUPE Java et l'Open Source

Java a connu un succès très rapide, et ce au moment même où Linux introduisait la notion d'Open Source auprès du grand public. Cette synergie a alors donné le jour à de nombreux projets communautaires (au sens où un ensemble de développeurs issus des quatre coins de notre planète participent à un même projet dans le seul but d'améliorer tel ou tel aspect de leur travail/passion). Beaucoup de projets sont nés, beaucoup sont morts, mais la dynamique n'est pas retombée et aujourd'hui, certains d'entre eux constituent de réelles forces de proposition au sein de la communauté d'utilisateurs Java. Ainsi, Exolab accueille différents projets (Castor par exemple), de même qu'Apache accueille Ant, Lucene, Tomcat, Oro.... Il faut préciser que l'Open Source n'est en rien une spécificité de Java, puisque les premiers projets de ce type sont Linux, Apache, Samba, Sendmail...

/// Projet pilote

Ce type de projet est dénommé projet pilote dans de nombreuses entreprises. Il fait partie de l'arsenal des managers qui doivent faire face à un changement radical de technologie...

/// DBA (DataBase Administrator)

Administrateur de bases de données : nom donné aux personnes qui créent et maintiennent de grosses bases de données.

Technologies et méthodologie

Ce projet doit être l'occasion de mettre en œuvre pour la première fois les technologies J2EE (servlets et EJB) et d'adopter une approche méthodologique fortement teintée par les principes majeurs de l'eXtreme Programming.

MÉTHODE L'eXtreme Programming

L'eXtreme Programming est un mouvement récent (1998) très original, puisqu'il s'agit à la fois d'une méthodologie et d'outils concrets visant à prendre enfin en compte dans un projet un aspect oublié par les autres mentors de l'objet : l'homme. Avec certains préceptes et certaines pratiques courantes, ce mouvement tâche d'améliorer la communication et la qualité du code.

Il faut quand même préciser qu'à la base, les pratiques décrites par les gourous de l'eXtreme Programming sont très bien adaptées à de petites équipes, pour des projets où les développeurs sont en contact avec le client final (pas toujours vrai).

Vous trouverez par la suite un peu plus d'informations sur ce mouvement très intéressant et l'on ne saurait que trop chaudement recommander la lecture de l'ouvrage suivant :

📖 J.-L. Bénard, L. Bossavit, R. Médina, D. Williams.- *Gestion de projet eXtreme Programming*, Eyrolles 2002.

Les notions les plus largement appréciées au sein de l'équipe sont le travail en binôme (*pair programming*) et l'importance accordée aux tests unitaires

Certains diagrammes UML seront utilisés pour fixer le vocabulaire commun, délimiter avec précision le contour fonctionnel de l'application et faciliter le codage.

À ce titre, il va donc être question de réaliser une application facilement distribuable.

Pour cela, le **chapitre 6** de cet ouvrage présente l'optique de BlueWeb, qui a déjà eu ce type de préoccupations et a donc une expérience qu'elle veut utiliser afin de se prémunir contre les problèmes fréquemment rencontrés. Bien évidemment, il s'agira de minimiser les interventions humaines (trop onéreuses) et d'utiliser le Web comme média de communication pour les mises à jour.

L'interface graphique sera développée en utilisant une bibliothèque assez récente mais qui fait déjà beaucoup de bruit : SWT. Il s'agit là en fait de la base du projet Eclipse, projet initié par IBM et dont le développement est assuré par un consortium.

L'idée d'utiliser cette bibliothèque plutôt que la très documentée Swing (devenue la bibliothèque standard pour les interfaces graphiques en Java) provient du fait que BlueWeb a la preuve (avec Eclipse) que l'on peut faire un très bon produit avec cette bibliothèque.

Il s'agit donc de mesurer, sur un projet concret, l'impact sur une planification de projet d'une telle bibliothèque en matière de temps d'adaptation, d'obtention de réponses et d'informations. Il s'agira aussi de mesurer la stabilité de cette bibliothèque. Toutes ces questions ne peuvent trouver réponse que par le biais d'un

réel projet et, étant donné l'importance de la question, il vaut mieux utiliser un projet pilote qu'un réel projet client pour mettre en œuvre cette bibliothèque. Pour BlueWeb, il est évident que l'utilisation de Swing sur un projet aussi simple aurait été une solution de facilité, car l'abondance de la documentation (livres/articles) aurait permis de surmonter les rares difficultés posées par l'interface graphique à réaliser. Cependant, cela n'aurait nullement permis de se forger une opinion sur les qualités/défauts de SWT et donc n'aurait contribué qu'à repousser un problème qui de toute évidence se posera bientôt pour BlueWeb.

Avec une optique similaire, BlueWeb décide de mettre en œuvre les EJB. Il s'agit de voir comment les mettre en œuvre et de pouvoir capitaliser des expériences concrètes d'utilisation de ces composants. D'autres solutions permettraient de gérer la persistance des données :

- l'API JDBC (Java DataBase Connectivity) ou via iBatis ;
- certains produits comme Hibernate, Cayenne JDO, KodoJDO, Speedo ou encore TopLink ;
- la norme JDO (Java Data Objects) de Sun.

ATTENTION **JDO et Castor JDO**

Il n'y a pas d'erreur dans la liste ci-dessus (quelques omissions) : par un caprice du sort, il se trouve qu'un produit (castor JDO) porte un nom aujourd'hui proche de celui d'une norme de Sun. En fait, le produit et la norme visent tout deux à assurer la persistance d'objets Java, mais le produit Castor JDO n'implémente pas du tout la norme de Sun !

L'utilisation des EJB un jour ou l'autre semble évidente pour BlueWeb et, de nouveau, ce projet pilote semble être l'occasion parfaite. En choisissant cette solution, on pourra compter sur des conclusions étayant le savoir-faire de BlueWeb sur des questions comme :

- la performance des EJB (en mode CMP) ;
- la facilité et le temps de codage ;
- La portabilité réelle.

Enfin, le choix d'adopter une couche de présentation réalisée par des servlets Java est dû à l'importance de HTTP dans l'informatique d'aujourd'hui.

Plus anecdotiquement, le code client se devra de mettre en œuvre le produit JUnit (support des tests unitaires), et ce afin de mesurer concrètement les bénéfices d'une telle approche, ainsi que l'impact sur la gestion d'un projet. Pourquoi se contenter d'une portion seule du code ? Avec ce projet, BlueWeb ne prétend pas chercher à atteindre la réutilisation mais juste à tester grandeur nature différentes solutions ; il en découle qu'il n'est nullement nécessaire d'imposer l'utilisation de ce framework à toute l'équipe si d'aventure l'expérience s'avérait négative avec ce produit.

OUTIL **iBatis**

iBatis se trouve dans l'incubator Apache.

- ▶ <http://incubator.apache.org/projects/ibatis.html>

OUTILS **Hibernate, Cayenne JDO, KodoJDO, Speedo et TopLink**

- ▶ www.hibernate.org
- ▶ www.objectstyle.org/cayenne/
- ▶ www.solarmetric.com
- ▶ <http://speedo.objectweb.org/>
- ▶ www.oracle.com

B.A.-BA **Framework**

Ce mot désigne un ensemble de classes destiné à réaliser une fonction ou rendre un service, et ce de manière réutilisable. La distinction avec une API est un peu subtile, mais on peut caricaturer en constatant qu'un framework est plus qu'une API. En citant la définition donnée dans l'ouvrage *Design Patterns* : « Un framework est un ensemble de classes qui coopèrent et permettent des conceptions réutilisables dans des catégories spécifiques de logiciels. ».