

EMERGENCE D'UNE NOUVELLE APPROCHE



ALIAS APPLICATION DESIGNER
ANNÉE 2013 N°1

DATE DE PARUTION : JANVIER 2013

*La technologie Web est mature et sûre pour construire les applications métiers
d'aujourd'hui à condition que ...*

Au fil de la décennie 2000, les technologies Web ont fait émerger différents produits et pratiques informatiques. D'abord orientée vitrine de l'entreprise puis E-commerce, l'utilisation de ces technologies prometteuses a marqué le pas dans le développement des applications métiers car les outils disponibles n'étaient pas adaptés à la multiplicité des règles et la complexité des MCD. Une nouvelle approche se devait d'émerger.



Dans ce numéro :

Histoire et terminologie pour la compréhension	1
Les limites successives	2
Les langages	2
Les Frameworks	2
Le Browser - Navigateur	2
Le facteur humain : les compétences, la formation	3
Répondre aux problèmes des applications métiers	3
Quelle Synthèse peut-on faire?	4
Alias AD – Le Framework des applications métiers	4

Histoire et terminologie pour la compréhension

Avant 2000, les logiciels étaient « propriétaires » c'est-à-dire commercialisés par des entreprises. Afin de promouvoir les technologies Web et accélérer leur déploiement, un consortium informatique organisé autour du **W3C** a fédéré tout un ensemble de développements permettant de couvrir les nombreux besoins. Ce sont les « **open sources** ».

Les applications utilisant ces nouveaux concepts de serveurs Web étaient au départ de simples serveurs informatifs. Ils diffusaient les informations sous forme de « **page HTML** » surfant d'une page à l'autre via des liens hypertexte (**url**). Puis les serveurs se sont sophistiqués pour aboutir à des serveurs applicatifs gérant des « **sessions** » de telle sorte que l'utilisateur possède lors du traitement son propre **contexte** et qu'il est alors possible de développer un dialogue structuré entre l'utilisateur et le service.

L'architecture 3-tier

Par ailleurs a émergé une architecture applicative originale mais très puissante : l'**architecture 3-tier**. Cette architecture introduit une décomposition en 3 couches fonctionnelles :

- Une couche « Présentation »
- Une couche « Métier »
- Une couche « Données »

Cette organisation répond, et c'est assez rare pour être souligné, à l'approche scientifique qui découpe les problèmes complexes et globaux en problèmes plus petits et homogènes susceptibles d'une meilleure solution.

Les systèmes d'exploitation sont principalement Linux, Windows Server, Aix.

Les limites successives

La première génération de serveurs ne prenait en compte que les requêtes 'http' sans considération de contexte. Dans cette catégorie, on trouve le serveur logiciel **Apache**. Ces serveurs gardent leur intérêt pour les applications de type vitrine. Ils peuvent être complétés par des composants qui vont prendre en charge le contexte.

La deuxième génération de serveurs est représentée par des serveurs applicatifs gérant, sans ajout de composants, le contexte. C'est le cas de **Tomcat**. Il existe également des produits du commerce comme **Websphere** et **Weblogic**.

Il existe aussi des « packages » comme **Wamp** qui rassemblent l'ensemble des composants nécessaires.

Les langages

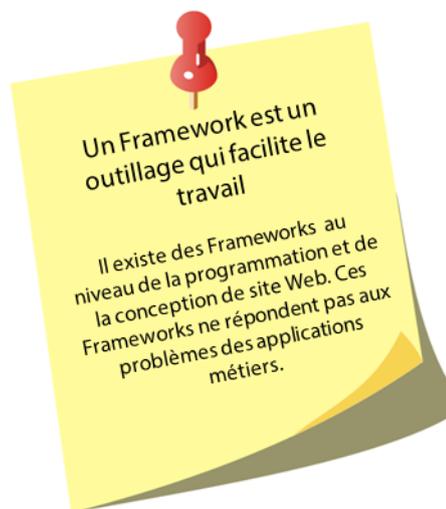
Un autre aspect est le langage de programmation.

Si Cobol a régné sur les applications métiers traditionnelles, il est exclu du monde du Web au profit d'une foison de langages tous supposés meilleurs les uns que les autres aux dires de leurs partisans.

La science s'est effacée pour faire place à des « chapelles ». On peut dénombrer le C, le C++, le C##, Python, PHP, ASP et beaucoup d'autres ainsi que le langage **Java** qui a une spécificité intéressante qui le différencie de tous les autres.

Pour plus d'informations sur Java, nous renvoyons aux bons auteurs. Des informations synthétiques sont disponibles sur Wikipedia. Les caractéristiques principales sont l'indépendance vis-à-vis du hardware ce qui permet une portabilité applicative et une très bonne performance sur les serveurs multiprocesseurs. Mais il existe bien d'autres aspects intéressants.

Les Frameworks



Le développement des applications Web était consommateur de temps. Des Frameworks de développement ont vu le jour pour améliorer la productivité. S'adressant aux développeurs, on devrait dire aux programmeurs, leur objectif est de faciliter et de rationaliser le codage d'une application en accélérant via des modèles (application blanche) ou des templates la mise en place d'une nouvelle application.

Ces Frameworks, s'ils permettent d'éviter les tâches répétitives de développement et une meilleure structuration des programmes, ignorent la problématique de traitement des données des applications de gestion. Le succès du Web n'était-il pas lié au côté ludique de la couche de présentation ?

Le Browser - Navigateur

Le « Browser » : c'est au travers de cette fenêtre ouverte sur le « Net » que s'exécutera l'ensemble des applications.

La liste est longue : **Internet Explorer, Firefox, Chrome, Safari...** Ce qu'il faut savoir, c'est que des différences existent en matière de restitution d'écrans et que ces différences existent non seulement entre différents Browsers mais aussi entre les versions d'un même Browser (par exemple entre IE8 et IE9).

Lors de la programmation de la couche de présentation, il faudra s'assurer de la bonne compatibilité au risque d'avoir des différences de rendu.



Le facteur humain: les compétences, la formation

Ce qui caractérise les nouvelles technologies : langages, design, systèmes, réseau, tout est différent.

Les cursus universitaires forment la nouvelle génération d'étudiants. Les langages et l'architecture sont maîtrisés. Si ces derniers sont très habiles sur la couche de présentation avec des effets spectaculaires, ils sont moins à l'aise sur la partie métier et base de données.

Les échecs de l'utilisation des technologies Web dans le développement des applications métiers, dont certains se font l'écho, proviennent essentiellement de l'absence de professionnalisme dans le système d'information, l'accès aux données et les contraintes de pistes d'audit. On constate alors un manque de rigueur dans la programmation en lien avec l'accès aux données et au respect des règles métiers.

L'ancienne génération de développeurs, c'est-à-dire la génération d'avant le Web, est exactement en contrepoint à ce qui vient d'être dit. Ils ne connaissent pas les nouveaux langages, la connaissance réseau est parfois faible et l'architecture applicative 3 tier peu familière.

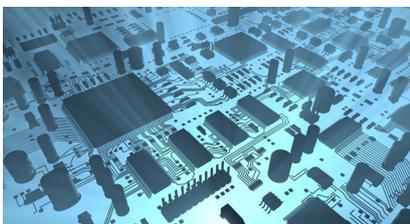
L'ancienne génération est-elle à « jeter » ? Bien sûr que non. Ces professionnels possèdent une grande compétence en méthodologie de développement d'applications métiers et de design de base de données. Cette compétence est ce qu'on appelle l'expérience. Elle s'est construite au fil du temps et elle n'est pas remplaçable.

Avec les outils dont on dispose, le « facteur humain » est un problème quasi insurmontable rapidement : personne ne regroupant l'ensemble des compétences nécessaires.



« LE FACTEUR HUMAN EST LA CLÉ DU SUCCÈS OU DE L'ÉCHEC »

Répondre aux problèmes des applications métiers



On peut répondre par quelques astuces et beaucoup de tuyaux à l'interconnexion des applications métiers.

Est-ce une approche pérenne?

La première réponse a été de juxtaposer aux applications traditionnelles un frontal Web et d'établir une connexion entre les deux environnements. Il y a eu la technique ancienne du RCP (Remote Call Program) puis SOAP qui a ensuite été structuré en Web Services.

La communication par Web Services est un concept très puissant qui va au delà de la communication avec des serveurs applicatifs. Par contre, il va trouver sa limite dans les cas de forts volumes de messages.

L'avantage de cette approche est de ne pas toucher aux applications métiers anciennes. Il laisse en revanche une cohabitation difficile entre deux environnements et deux générations de systèmes et de personnes et ne va pas dans le sens de la diminution des coûts.

Il répond juste à une situation d'urgence.



Développement et éditions de logiciels
59, rue Glesener
L-1631 Luxembourg
LUXEMBOURG



**Avec Alias,
Spécifier c'est développer !**

www.alias-ad.com

Quelle synthèse peut-on faire ?

Les outils évoqués ont parfaitement joué leur rôle et le jouent encore pour le développement de portails.

Par contre, ils laissent de côté trois points essentiels pour aborder les applications à forte valeur métier :

↳ Le traitement des données

- MCD
- Base de données
- Contrôle d'accès aux données
- Performance

↳ La gestion de l'évolution d'une application

- Mise en place de nouvelles fonctionnalités
- Correction des problèmes
- Gestion de la non-régression

↳ Un problème de génération d'informaticiens

ALIAS APPLICATION DESIGNER a pour ambition de relever le défi du développement d'applications à forte valeur métier.

Alias AD - Le Framework des applications métiers



La solution

Passer d'un site Web informatif à des applications métiers web introduit toute la complexité des systèmes d'information, des règles et contraintes ainsi que du contexte technique propre à la technologie WEB.

Est-ce un obstacle infranchissable (ou trop coûteux) ? La réponse est non, bien au contraire. A condition d'une bonne analyse et d'un peu d'aide apportée par un outil gommant la complexité technologique. Le concepteur se situe alors au niveau des spécifications fonctionnelles de l'application.

Compte tenu de l'absence d'un tel outil, nous avons décidé de le créer. Son nom :

ALIAS APPLICATION DESIGNER

Vous en saurez plus dans nos prochaines Newsletters

Disponible sur la boutique en ligne le 15 Octobre 2013