

CONSTRUIRE UNE APPLICATION METIER



ALIAS APPLICATION DESIGNER
ANNÉE 2013 N°3

DATE DE PARUTION : MARS 2013

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

Construire une application robuste et pérenne, c'est définir au moment de la conception non seulement les objectifs fonctionnels mais aussi envisager le cycle de vie et réfléchir sur la manière la plus simple d'assembler tous les éléments.



Dans ce numéro :

Normalisation des composants applicatifs	1
Les éléments d'architecture	2
La couche de présentation : l'IHM	2
La couche applicative : Spécifier plutôt que Développer	3
La couche Données	3
Les phases de constructions d'une application	4
Alias AD - Repense le métier du développement	4

La question alors posée d'un point de vue technique est : comment construire un système applicatif qui permet l'évolution et l'interchangeabilité des composants applicatifs dans le temps?

Normalisation des composants applicatifs

Rendre les composants applicatifs interchangeables, c'est normaliser les échanges entre les composants.

S'inspirer du Hardware !

L'architecture hardware des ordinateurs a été depuis longtemps confrontée à ce besoin. Ces derniers ont un composant principal qui s'appelle un « board » (ou carte mère) sur lequel tout est connecté. En particulier, on trouve des connecteurs (slot) qui permettent d'insérer des cartes de fonctions variées. On peut ajouter, supprimer, remplacer les cartes pour faire évoluer l'ordinateur. Les tous premiers ordinateurs n'avaient pas cette architecture.

Transposer !

En transposant, le composant Alias « base applicative » va jouer le rôle d'accueil des composants applicatifs. Il reprend le rôle de la « carte mère ».

En continuant la transposition, il faut également définir un « bus de communication » suffisamment ouvert pour permettre les évolutions.

Chaque composant applicatif peut alors être construit séparément à condition qu'il respecte les normes d'interface telles que le font les cartes d'extension sur PC .

On disposera ainsi d'une grande flexibilité dans l'évolution d'une application.

**Répondre à cette architecture suppose un outillage approprié.
ALIAS APPLICATION DESIGNER est conçu pour cet objectif**

Les éléments d'architecture

L'architecture applicative: le modèle 3-tier

Le modèle d'architecture retenu est le modèle 3-tier. On rappelle qu'il est constitué d'une couche de présentation, d'une couche métier et d'une couche données. (Cf newsletter n°2 de février)

La représentation des données : XML

Sous ce sigle ésotérique se cache un langage de description de données indépendant de toute technologie.

Sa capacité de description est de répondre aux problèmes complexes des MCD et des échanges.

XML s'est révélé être une formulation extrêmement puissante dans la description et les échanges de données. Décrire les données en XML est particulièrement utile dans des cas simples comme la représentation d'une facture mais plus encore dans les cas compliqués comme par exemple la représentation d'un contrat d'assurance.

On représentera en XML aussi bien les messages, les tables relationnelles, les règles etc.

On unifie la structure de présentation des données quelle que soit la forme physique du stockage.

NB: Nous renvoyons sur internet pour plus d'explications. Wikipedia indique « L'objectif initial (d'XML) est de faciliter l'échange automatisé de contenus complexes (arbre, texte riche...) entre systèmes d'information hétérogènes (interopérabilité) »

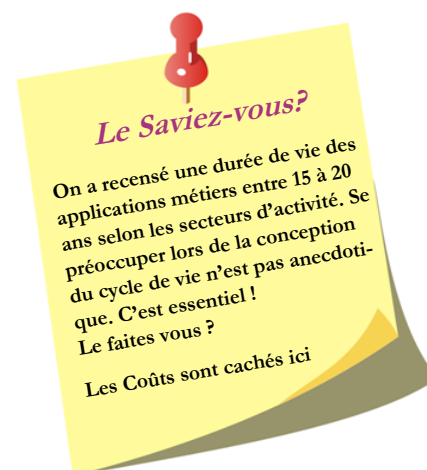
Une conception pour le « multi-canal »

Une application métier peut être sollicitée à partir de différentes provenances.

Il peut s'agir du transactionnel classique ou peut provenir d'un autre site en utilisant des Web services ou encore par transfert d'un flux de messages formés sur un autre système.

La « meilleure pratique » est de concevoir la couche métier pour qu'elle soit indépendante de l'origine des données.

La conséquence est d'appeler, dans tous les cas, les mêmes programmes. On ne développe qu'une seule fois. Cette approche est plusieurs fois gagnante.



La couche de présentation: l'IHM

Dans l'architecture 3-tier, la couche IHM est indépendante. Elle communique avec la couche applicative.

La couche IHM est donc composée du moteur de dialogue qui échange avec le moteur applicatif les données et la page formulaire.

Ces données d'échanges sont décrites en XML et le formulaire est de type JSP.

Par ailleurs, en ce qui concerne les aspects de couleurs, de taille, etc. la bonne pratique est d'utiliser des thèmes décrits dans des fichiers CSS. On associe au formulaire un thème. On peut ainsi changer à chaque initialisation du serveur le thème souhaité.

On peut créer les formulaires JSP par tout outil du commerce.

Lorsqu'on dialogue au moyen d'un formulaire, il est souhaitable de contrôler le plus près possible les données

saisies. C'est ce qu'on appelle les contrôles de surface. Les règles que doivent vérifier les champs doivent hériter des règles de la couche applicative.

Un bon outil de conception de formulaire doit pouvoir profiter de cet héritage pour être plus efficace et éviter les incohérences.

ALIAS APPLICATION DESIGNER propose un outil de conception de formulaire qui respecte les principes énoncés.

Note: IHM signifie Interface Homme Machine



La couche Applicative: Spécifier plutôt que Développer

Le moteur applicatif

Le cœur de la couche applicative est le moteur applicatif. Par lui transite toutes les données en provenance des canaux de liaisons et du moteur de données.

L'ensemble des données qui participent aux traitements sont décrites en XML. Pour standardiser les échanges, le moteur applicatif traite des messages.

Les informations peuvent se classer en messages d'entrée/sortie, en message de données persistantes parmi lesquelles on trouvera pour les applications professionnelles les données de configuration, les données de paramétrage applicatif et les données dites de production.

Dans le contexte de gestion de la vie de l'applicatif, on verra l'importance de cette distinction.

Si on analyse les divers types de fonctions que doit rassembler une application, les schémas de traitement à réaliser sont en petit nombre. A titre d'exemple, on va trouver des fonctions de création, de modification, de suppression, de copie, de lecture pour les basiques et d'autres plus élaborées comme une modification avec historique, etc. A chaque schéma de traitement correspondra un programmes type.

Le moteur applicatif est construit pour réaliser, dans le cadre d'un schéma de traitement, les opérations sur les données précisées par les règles et les contraintes.

Les données nécessaires au moteur applicatif pour une fonction sont décrites dans une structure XML appelé **Dossier Fonctionnel**. Ces données sont « associées » aux informations gérées par le moteur de données. D'autres informations appelées « zones de travail » peuvent aussi être définies.

C'est sur ce dossier fonctionnel qu'on précisera les règles de contraintes et de dépendances (par exemple : un calcul est une règle de dépendance dont l'action est de délivrer le résultat). On peut considérer en première approximation que la manière de décrire l'objectif souhaité s'apparente aux spécifications des tableurs. Les mécanismes de données sont considérablement plus sophistiqués, il faut arrêter ici la comparaison.

Chaque fonction de l'application est donc spécifiée. Il n'y a pas de programmation.



*Compétence, formation, expérience
Rien n'est simple ...*

*L'approche applicative d'ALIAS AD
remet les fonctionnels
au centre du jeu*

La couche Données

Généralisation du concept de données

Il est plus facile de décrire l'accès aux données en faisant référence à une « base de données ». Mais les données dont a besoin le traitement peuvent provenir de tous les points du réseau via des protocoles différents. Le moteur de données doit résoudre ce besoin.

Ainsi le moteur de données n'est pas une « méthode d'accès à telle ou telle base de données », c'est le composant qui va rassembler l'information et la structurer pour que le traitement applicatif n'ait à gérer ni la complexité du référentiel, ni la topologie.

La virtualisation des données

Le moteur de données communique d'une part avec les couches d'accès aux données et d'autre part avec les traitements via le moteur applicatif où il adresse le dossier fonctionnel

Le dossier fonctionnel peut être considéré comme une « vue » logique. Mais s'il a des éléments d'analogie, la réalité est différente : il n'y a pas spécifiquement de définition de « vue » au niveau de la base de données.

Le moteur de données règlera les échanges entre la couche applicative et la couche physique.

En déplaçant la vue et la récupération des données au niveau du moteur de données, le serveur applicatif n'a plus à formuler d'accès aux données. Le moteur applicatif est considérablement simplifié.

La vision des données par la couche applicative est virtuelle.



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

alias



Alias Application Designer

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

www.alias-ad.com

Les phases de constructions d'une application

- Les éléments d'architecture
- Le cycle de vie
- Les composants applicatifs
- Le packaging
- L'installation
- Le déploiement

Une prochaine Newsletter abordera les points du cycle de vie, du packaging et du déploiement qui est souvent appelé maintenance. Mais cette appellation est restrictive et ne rend pas compte de la complexité sous-jacente qu'il faut résoudre.

ALIAS APPLICATION DESIGNER a pour ambition de relever le défi du développement d'applications à fortes valeurs métiers

Alias AD - Repense le métier du développement



La solution

ALIAS APPLICATION DESIGNER va bien au-delà d'une simple conception d'applications métiers en nouvelles technologies. Il redéfinit tout le concept d'Applications dans un cadre global où données et serveurs sont dispersés et où la rigueur de conception est plus que jamais indispensable.

Alias AD gomme la complexité technologique pour situer le concepteur uniquement au niveau fonctionnel de l'application, tout en lui faisant disposer des solutions techniques les plus sophistiquées. Nous en reparlerons.

Compte tenu de l'absence d'un tel outil nous l'avons créé.

Son nom :

ALIAS APPLICATION DESIGNER

Vous en saurez plus dans nos prochaines Newsletters

Disponible sur la boutique en ligne le 15 Octobre 2013

Copyright Sun i-Tech 2013.

La reproduction des textes de cette newsletter est autorisée sous réserve du respect des trois conditions cumulatives suivantes : - que la diffusion de ces reproductions soit gratuite ; - que les documents reproduits respectent l'intégrité du contenu des documents d'origine, sans modification ni altération d'aucune sorte ; - que la source soit citée clairement et lisiblement. Les éventuelles marques citées dans la Newsletter sont déposées par les sociétés qui en sont propriétaires.