

Modèle multicouche

Architecture à 5 couches

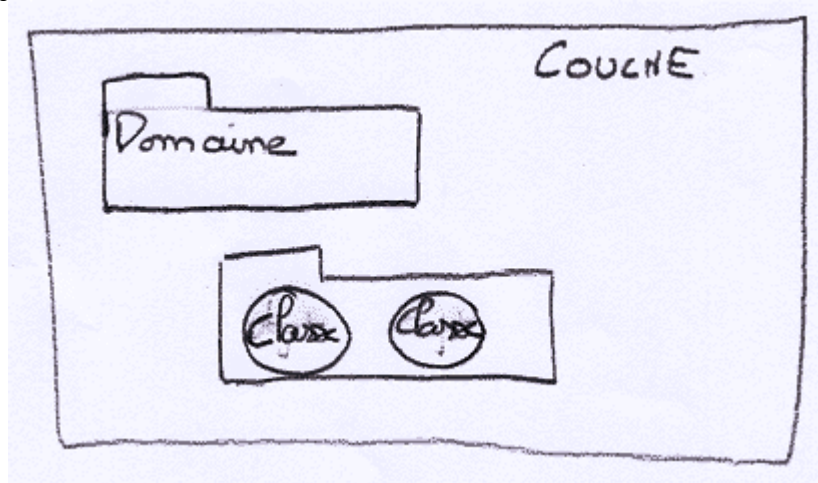
Introduction

Les cycles de développement ne cessent de s'accélérer. Malheureusement, les ressources se font de plus en plus rares. La production de logiciels nécessite donc, aujourd'hui et plus encore demain, des hauts niveaux de productivité et de réutilisation.

Nous décrivons ci-dessous comment le modèle multicouche IMPROVE répond à ces défis.

Les architectures à couches

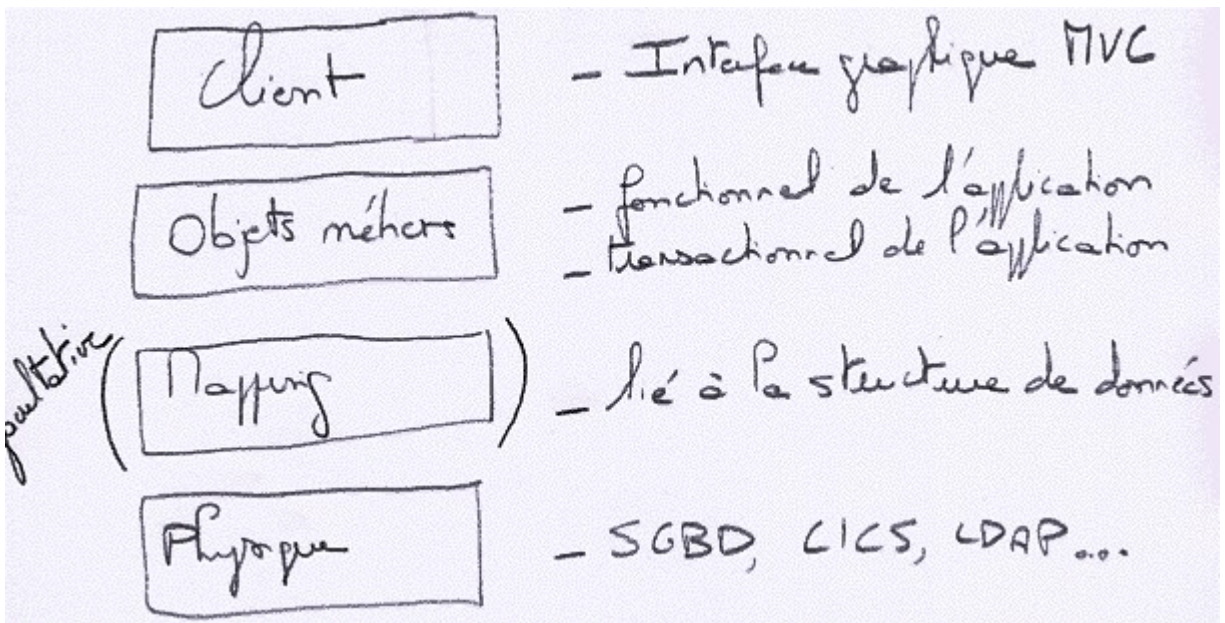
La technologie Objet requiert une architecture. C'est cette architecture qui organise les interactions entre objets. On a l'habitude de regrouper ces objets en classes, ces classes en domaines, et ces domaines en couches.



Les couches permettent de présenter l'architecture de l'application. Les équipes de réalisation s'attribuent alors des responsabilités sur le développement de chaque couche. Aussi, si modéliser est indispensable, construire une architecture à couche est un critère de qualité dans le cadre d'un développement Objet. Reste à choisir le nombre de couches et à définir leur contenu.

Les Objets Métier ne suffisent pas

Parmi les architectures à couches, l'architecture à base d' "Objets Métier" occupe la première place en matière de notoriété. On parle aussi de "Composants Métier" ou encore de "Logique Métier". Cette architecture correspond généralement à un modèle à 3 ou 4 couches.

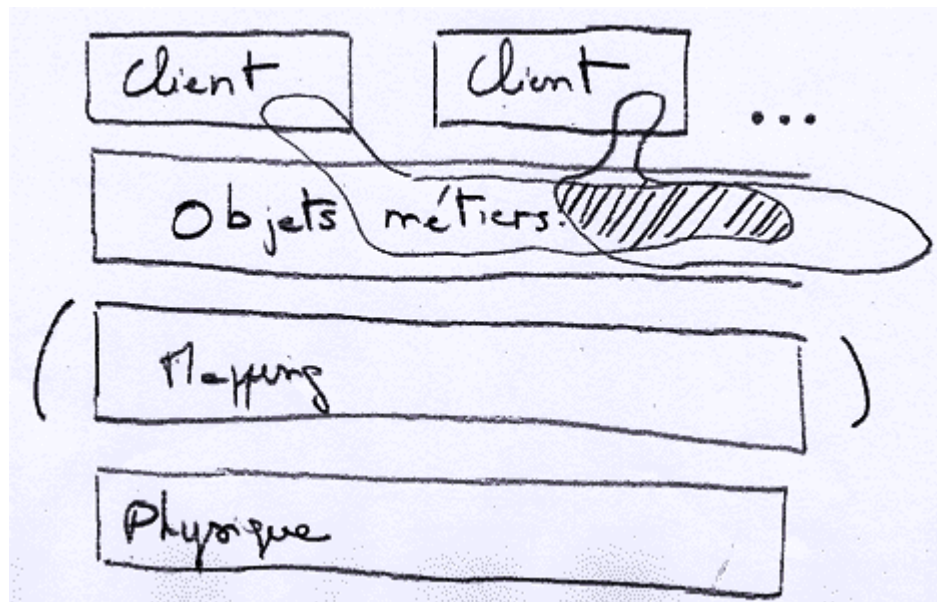


Dans cette architecture, les "Objets Métier" sont directement manipulés dans l'interface graphique. Ils contiennent la logique fonctionnelle, c'est-à-dire les traitements à effectuer, ainsi que la logique transactionnelle (i.e. mises à jour des bases de données).

L'architecture à base d' "Objets Métier" mêle donc la logique fonctionnelle et les aspects transactionnels. C'est pourquoi, une évolution fonctionnelle ou transactionnelle entraîne des modifications dans les deux niveaux de code. Pour garantir l'intégrité, cette architecture contraint à tester l'ensemble de l'application à chaque modification, alors que seul un point fonctionnel peut avoir été ajouté.

De ce point de vue, l'architecture à base d' "Objets Métier" nuit à l'évolutivité et à la maintenabilité des développements.

Plaçons-nous maintenant dans le contexte de développement de plusieurs applications. Ce mélange fonctionnel et transactionnel pose de nouveaux problèmes. En effet, chaque application possédant sa logique fonctionnelle, il devient nécessaire de mixer ces logiques fonctionnelles au sein des mêmes "Objets Métier". Et ce tout en conservant le même transactionnel !



Deux voies s'offrent à nous :

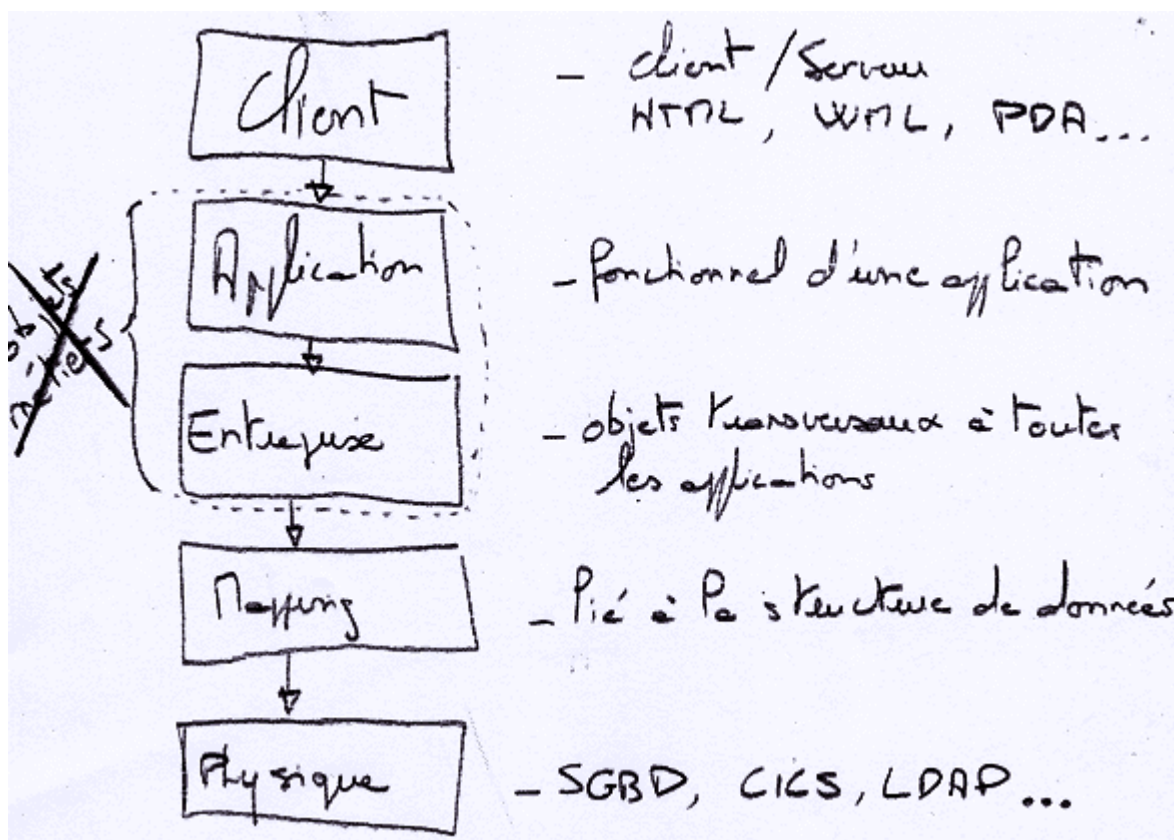
1. capitaliser sur les "Objets Métier", auquel cas la modification d'une application impacte tous les développements réalisés !
2. ne pas capitaliser, ce qui revient à redévelopper l'ensemble de la logique transactionnelle !

Dans le premier cas, on pénalise l'évolutivité et la maintenabilité, dans le second, on sacrifie la réutilisabilité. Enfin, dans les deux cas, on reste incapable d'intervenir sur la logique fonctionnelle indépendamment des aspects transactionnels.

Bref, si les "Objets Métier" posent un problème d'évolutivité des développements mono application, ils ne permettent absolument pas de gérer des développements multi applications.

Le modèle Multicouche IMPROVE

Nous avons vu précédemment qu'il est indispensable de dissocier la logique fonctionnelle d'une application des aspects transactionnels. Pour répondre à cette problématique, IMPROVE propose une architecture à 5 couches, qui isole le périmètre fonctionnel des applications.



Pour illustrer le contenu de chaque couche, nous prendrons l'exemple d'une banque qui offre des services de gestion de compte, en ligne et en agence, l'application de gestion des comptes en ligne disposant de client WAP et WEB.

Description des couches	Exemple d'une banque
La couche Physique correspond à la structure physique des données: SGBD, annuaire LDAP, Transaction CICS, ..	Base de données relationnelle
La couche Mapping réalise les accès vers la couche Physique. Dans le cas d'un SGBDR, il s'agit d'un outil de mapping relationnel / objet.	Mapping JDBC avec un outil de type : Data Access Builder, TopLink, Castor, ...
La couche Entreprise correspond aux objets structurants de l'entreprise. Ces objets n'intègrent aucune notion fonctionnelle. Cette couche regroupe des objets transversaux à toutes les applications. De plus, la couche Entreprise propose des services d'accès à ces objets au travers de méthodes de création, recherche, modification, suppression. Ces méthodes contiennent les règles de gestion associées aux différentes opérations.	Objets : Client, Compte, Agence, ... Services : Création d'un compte, suppression d'un compte, ...
La couche Application regroupe la logique fonctionnelle d'une application, tel qu'elle est définie dans les spécifications fonctionnelles détaillées. La couche Application utilise les services de la couche Entreprise pour réaliser le fonctionnel spécifié, et présente ce fonctionnel sous la forme de	Services de l'application Banque en ligne : authentification d'un client, ... Services de l'application Agence :

services. Ces services retournent des objets de niveau application, qui sont autant de vues sur les objets entreprise. Les objets de niveau application peuvent, par exemple, masquer l'information "découvert autorisé" d'un objet Compte de niveau Entreprise.	authentification du guichetier,... Objets de l'application Agence : Compte (vue sur l'objet entreprise Compte)
La couche Client représente l'interface utilisateur. Elle est amenée à changer fréquemment dans le cas d'une application WEB. Parce que la couche Application regroupe les règles fonctionnelles, il est possible de faire évoluer les IHM à moindre coût.	Banque en Ligne WEB : HTML, Servlets Banque en Ligne WAP : WML, Servlets Client / Serveur Agence : JFC

Les plus du modèle Multicouche IMPROVE

L'architecture à 5 couches présentée ci-dessus, permet de décomposer le système en sous-systèmes. Chacun de ces sous-systèmes peut être modélisé de façon indépendante, et il est possible d'affecter des responsabilités aux intervenants du projet sur telle ou telle couche (analyse / implémentation).

Chaque couche est développée et testée de façon individuelle. Le modèle multicouche propose un cadre de conception systématique, qui vise à l'industrialisation de la production de logiciels.

Dans ce modèle d'architecture, la couche Application est non seulement indépendante de la couche Client, mais elle repose aussi sur une couche Entreprise normalisée. Le transactionnel et l'interactif sont complètement détachés du fonctionnel, au bénéfice de la maintenabilité et de l'évolutivité..

Le modèle multicouche propose un cadre de conception systématique, garantissant des hauts niveaux :

- d'évolutivité, en dissociant le fonctionnel du transactionnel,
- de maintenabilité, en attribuant des responsabilités par couche,
- et de réutilisabilité, en assurant un faible couplage entre les couches.

Enfin, l'architecture à 5 couches permet de reporter les spécificités liées à telle ou telle technologie au niveau de l'implémentation des services des couches "Application" et "Entreprise".

Cette généricité permet au modèle multicouche d'adresser des développements Client / Serveur aussi bien que J2EE (EJB et Servlets / JSP).

Conclusion

Nous avons présenté le modèle d'architecture de référence IMPROVE. Ce modèle garantit un haut niveau de qualité logicielle à travers ces caractéristiques de maintenabilité, d'évolutivité et de réutilisabilité. Le modèle multicouche IMPROVE vise ainsi l'industrialisation de la production de logiciels.

De plus, ce modèle peut être appliqué pour le développement d'applications WEB / PDA (client léger) aussi bien que Client / Serveur (client lourd). En isolant le fonctionnel de l'IHM, il répond à la problématique d'hétérogénéité des terminaux client, à laquelle est soumise la nouvelle économie.

La société IMPROVE est spécialisée dans la mise en place de ce modèle sur les technologies Objet : J2EE et WebObjects.

Auteur

De formation Télécom, Stève SFARTZ est consultant senior chez IMPROVE SA. SUN Certified JAVA Programmer et IBM VisualAge for JAVA Certified Programmer depuis 1998, Stève met en place des architectures JAVA depuis 1996 (Serveur d'application J2EE, Transactionnel, Client / Serveur JFC, architectures INTRA / INTERNET, architectures Objet / UML). Stève anime par ailleurs des conférences et des formations sur les technologies VisualAge, WebSphere, Processus de développement Objet, Architecture Objet UML...

Références

Four Layer Architecture de Kyle Brown
iCommerce Design Issues and Solutions de Gemstone