



Le projet **DELIRE**
Développement par Equipe
de Livrables Informatiques
et Réalisation Encadrée

PSp6 – Architecture Fonctionnelle



ARCHITECTURE FONCTIONNELLE

Pour être tout-à-fait honnête, je vous ai présenté l'Architecture Fonctionnelle et l'Architecture Logique de façon un peu simplifiée

1. L'Architecture Fonctionnelle est l'architecture telle qu'elle va être perçue par l'utilisateur.
2. L'Architecture Logique est la décomposition du système en sous systèmes et composant.

Ainsi, pour une voiture, dans l'Architecture Fonctionnelle vous diriez simplement que lorsqu'on tourne le volant, cela a pour effet de tourner les roues avant. Par contre, il n'est nullement spécifié ni expliqué comment la transmission de l'ordre du conducteur jusqu'au x roues de direction s'effectue : ceci sera du domaine de l'Architecture Logique.

L'intérêt de cette approche est de pouvoir modifier l'Architecture Logique sans impacter l'Architecture Fonctionnelle. Ainsi :

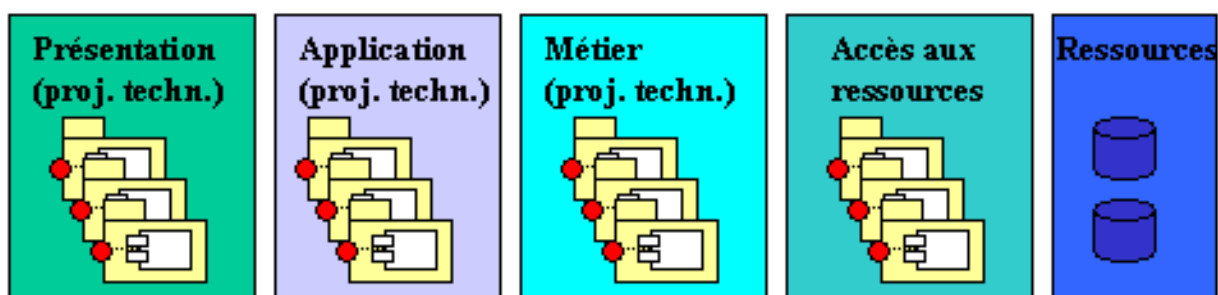
Si je passe d'une persistance fichier à une persistance en table de données relationnelle, j'ai un très gros travail à faire en termes de gestion des objets en mémoire et de la persistance, mais l'opération est « quasi » transparente pour l'utilisateur.

De même, si je décide de passer d'une application monobloc à une application multi-tiers (exemple : client/serveur), là aussi cela me conduit à modifier (et probablement améliorer) mon Architecture Logique mais non pas la présentation, la gestion des interactions et la mise en forme des informations.

Urbanisation

La réalité est un tantinet plus complexe. Ceux qui choisiront une spécialisation dans le SI (Système d'Information) seront appelés à travailler de façon plus approfondie l'architecture des systèmes d'information, ce que l'on appelle souvent l'Urbanisation. Vous verrez se dégager un certain nombre de couches d'architecture.

Exemple d'architecture 5 tiers : Présentation, Application, Métier, Accès aux ressources, Ressources



MVC

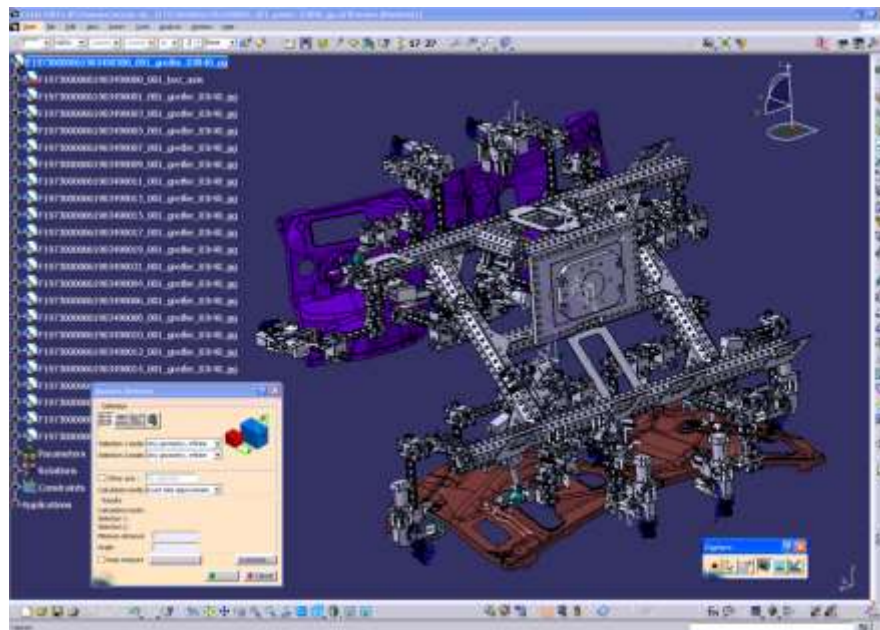
Mais dans de nombreux cas, l'architecture est une architecture basée sur le Design Pattern MVC. Un Design Pattern représente la capitalisation d'un savoir-faire en architecture informatique, reconnus par les experts du domaine.

Le Design Pattern MVC (**Modèle-Vue-Contrôleur** ou Model-View-Controller) définit un paradigme pour l'organisation d'une IHM (Interface Homme Machine) d'une application, qui repose fondamentalement sur la séparation entre les données, la présentation et les traitements.



Pour ce faire, il divise l'architecture en:

1. Le modèle, c'est à dire le comportement des objets de l'application : traitements des données, intégrité, persistance. Il offre en particulier les méthodes classiques de création, modification, suppression, sauvegarde et chargement en mémoire. Le modèle ignore toute notion de présentation
2. La vue, qui correspond à l'interface avec laquelle l'utilisateur interagit. Elle présente les résultats renvoyés par le modèle. Elle reçoit toutes les interactions utilisateur (sélection, entrée de champs...). Elle n'effectue aucun traitement sur les données. La vue peut être conçue en HTML. Plusieurs vues peuvent simultanément présenter le même modèle. Ainsi dans un système de CAO qui présente une pièce géométrique (le modèle) on a souvent une vue qui présente l'arbre des spécifications, et une vue graphique qui présente le résultat géométrique desdites spécifications.



3. Le contrôleur : il prend en charge la gestion des événements et gère la synchronisation des différentes vues et du modèle. Le contrôleur n'effectue aucun traitement sur les données, mais dispatche les événements utilisateur ou modèle et enclenche les actions à effectuer.
 - a. Soit une modification des données, et il demande ensuite aux vues de se mettre à jour
 - b. Soit une modification qui ne concerne que la vue (zoom, rotation...).

Les Design Patterns les plus connus sont au nombre de 23. Ils sont couramment appelés « patrons GoF ». Les patrons Gof (Gang of Four, parce que leurs auteurs étaient au nombre de 4) ont été formalisés dans le livre « Design Patterns -- Elements of Reusable Object-Oriented Software » en 1995.



Mais dans DELIRE

Dans DELIRE, nous nous contenterons de cette première approche simplifiée : Architecture Fonctionnelle et Architecture Logique sont deux activités disjointes. En d'autres termes, ce que j'appellerai Architecture Fonctionnelle correspond à la partie Présentation (ou Vue dans le MVC), et correspond à l'affichage, à la gestion des interactions et à la mise en forme des informations échangées avec l'application.

Sachez-le, faire une interface utilisateur qui plaise se révèle difficile et long. Je n'ai pas de processus miracle pour définir une bonne Interface Utilisateur. Et pour une raison très simple, qui est que c'est en effet assez culturel : c'est un peu comparable à la décoration d'un appartement. Ainsi, dans la vie vous verrez que certains sont fanas de Windows et d'autres d'APPLE.

Je peux néanmoins vous donner quelques principes à ne pas totalement négliger.

Soyez simple, soyez simple, soyez simple.

Des termes qui ne présupposent pas un niveau Bac + 7.

Pas de mots importants trop proches l'un de l'autre.

Enfin, encouragez tout ce qui pourra réduire la complexité de l'IHM, à commencer par la suppression des commandes à faible valeur ajoutée.

Adoptez des standards

La fenêtre de menus toujours à la même place, la fenêtre de présentation des résultats toujours à la même place, le même terme utilisé systématiquement pour le même objet ou le même type d'interaction tout au long du produit.

Maximisez la place dévolue à l'information.

En d'autres termes, réduisez la place prise par les bannières, la publicité ou les menus. Pour ce faire, utiliser par exemple des icônes standards.

Vous pouvez afficher le nom et le logo de votre entreprise, en tant que développeur du site, mais par exemple à la demande dans la commande « Qui sommes nous »

Ne présentez que l'information nécessaire.

Par contre débrouillez-vous pour que l'information importante soit visible sans avoir besoin de scroller la fenêtre de résultat : d'abord les informations importantes, les suivantes ensuite.

Evitez au maximum le multifenêtrage

C'est un cache misère qui masque une absence de véritable Architecture Fonctionnelle, et qui maximise l'effort à fournir pour l'utilisateur. . cr

Ne demandez pas à votre utilisateur de faire des efforts.

Sélectionner est toujours préférable à valuer un champ.

Je me souviens dans une étude d'ergonomie de CATIA que nous avons constaté le fait que le dialogue imposait à l'utilisateur de recopier sur un papier une valeur pour la ressaisir dans un champ quelques interactions plus loin : pas bon ça !



Soyez efficace

L'esthétique est importante, mais la convivialité est encore plus importante
Certaines fioritures sont certainement belles, mais sans intérêt informatif pour l'utilisateur.

Souvenez-vous du premier principe : soyez simple.

Prenez bien en compte les critères définis dans le PQ (Plan Qualité)

UML

L'utilisation d'UML (Unified Modeling Language) permet d'assurer une continuité entre Architecture Fonctionnelle et Architecture Logique. Ceux qui le maîtrisent peuvent utiliser UML, si tant est que cette utilisation ne les retarde pas trop dans le projet DELIRE.

Placez toujours l'utilisateur au centre de votre travail

Une fois que vous êtes satisfait de votre proposition, faites la valider par d'autres.

1. Commencez par l'équipe.
2. Puis des copains de l'université.
3. Et enfin, faites le test dit de la grand-mère.

Et partez de l'hypothèse que chaque remarque utilisateur est pertinente. Si la grand-mère dit que c'est compliqué, c'est que ça l'est.

A la fin de votre évaluation, c'est parce que vous aurez naturellement facilité la prise en main de votre logiciel, en simplifiant la recherche et en réduisant l'appréhension par l'utilisation systématique de standard que vos utilisateurs vous attribueront une bonne note.

Etre certain que votre proposition est simple et complète peut vous sembler totalement évident : quel besoin de demander leur avis à des personnes extérieures. Et pourtant, une petite anecdote.

De 1946 à 2003, nous avons eu en France le « Plan Quinquennal »

Le 3^{ème} plan (1958 - 1961) tente de définir une croissance dans un contexte difficile :

1. Entrée de la France dans le Marché Commun
2. Guerre d'Algérie
3. Arrivée sur le marché des enfants du baby-boom

Ce plan n'a qu'un problème : il a oublié le développement des télécommunications

Dans les années 60, le téléphone en France est une véritable misère : évoquez avec vos parents ou plutôt vos grands-parents, le sketch de Fernand Raynaud : le 22 à Asnières (1966).





Ancien élève de l'X et des ponts et Chaussées, docteur ès sciences, Pierre Massé est nommé responsable du plan, l'ardente obligation selon le Général, de 1959 à 1965. Il s'agit principalement d'orienter la production vers des secteurs prioritaires pour la nation : secteur hospitalier, autoroutes et télécommunications.



Son 4ème plan (1962 – 1965) où il a tenté d'analyser les vrais besoins de la nation, est aujourd'hui considéré comme un modèle du genre. Mais il faudra plus de 20 ans pour rattraper le retard engendré par le 3ème plan.

Et maintenant à vous : c'est en dessinant des IHM que vous finirez par acquérir des automatismes.

