

Architecture Modèle-Vue-Contrôleur (MVC)

Ludovic Liétard

1

Architecture Modèle-Vue-Contrôleur

- Architecture logicielle qui permet de séparer :
 - ◆ les objets applicatifs (Modèle)
 - ◆ leur représentation (Vue)
 - ◆ les interactions qu'ils subissent (Contrôleur)
- Utilisée pour la programmation
 - ◆ d'architecture client/serveur, d'interface graphique
- Généralisation du pattern Observateur

2

Architecture Modèle-Vue-Contrôleur

- Schéma de programmation qui permet de séparer une application en 3 parties :
 - ◆ Côté applicatif : Le modèle contient la logique de l'application
 - ◆ Côté visuel : Les vues affichent à l'utilisateur des informations sur le modèle
 - ◆ Côté événementiel : Le contrôleur agit sur demande de l'utilisateur et effectue les actions nécessaires sur le modèle (ou la vue).

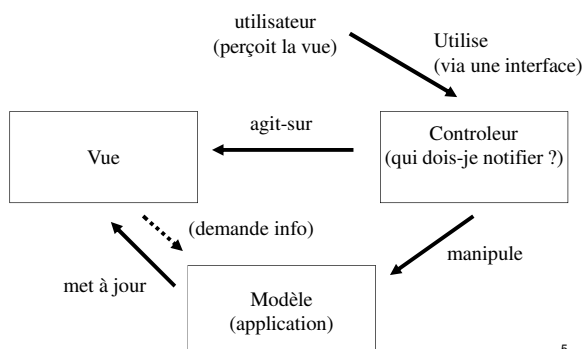
3

Architecture Modèle-Vue-Contrôleur

- Exemple : Application de calcul du budget d'une société
 - ◆ le modèle : les données et les procédures de calcul pour le budget prévisionnel de la société
 - ◆ les vues : une représentation sous forme de tableau, une sous forme graphique, une vue pour la direction, une pour la DRH,...
 - ◆ le contrôleur : des éléments de l'interface pour faire varier les paramètres du calcul

4

Architecture Modèle-Vue-Contrôleur



5

Architecture Modèle-Vue-Contrôleur

- Le contrôleur :
 - ◆ Reçoit les événements de l'interface utilisateur
 - ◆ Les traduit :
 - en changement dans la vue s'ils agissent sur le côté visuel
 - en changement dans le modèle s'ils agissent sur le contenu du modèle (exemple : modification d'un paramètre du calcul)

6

Architecture Modèle-Vue-Contrôleur

- Le modèle :
 - ◆ Il peut être modifié sur ordre du contrôleur
 - ◆ Il signale à ses vues tout changement de contenu en leur envoyant un événement qui leur spécifie de se mettre à jour

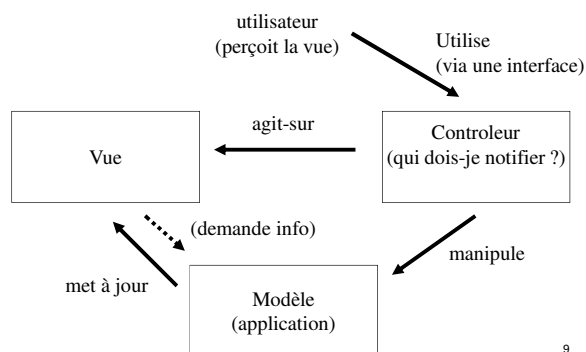
7

Architecture Modèle-Vue-Contrôleur

- La vue :
 - ◆ La vue se met à jour dès qu'elle reçoit un ordre de notification
 - ⇒ du contrôleur
 - ⇒ du modèle
 - ◆ Quand la notification vient du modèle, elle va consulter le modèle pour se réafficher de manière adéquate

8

Architecture Modèle-Vue-Contrôleur



9

Architecture Modèle-Vue-Contrôleur

- Le patron de conception *observateur* peut être utilisé :
 - ◆ le contrôleur et le modèle sont des *sujets* de la vue qui est un *observateur*
 - ⇒ Plusieurs vues pour une même application sont possibles
 - ◆ Plusieurs contrôleurs (utilisateurs) pour une même application sont possibles (intérêt ?).

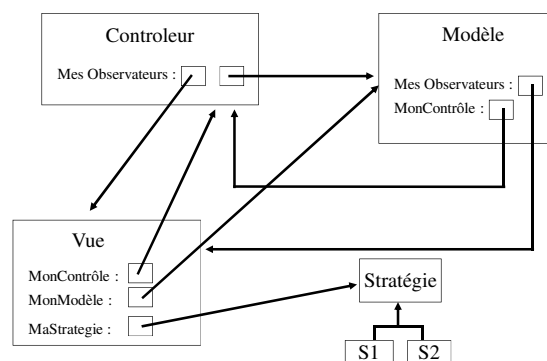
10

Architecture Modèle-Vue-Contrôleur

- Le patron de conception *stratégie* peut être utilisé :
 - ◆ la vue implémente le pattern stratégie (avantage de la délégation : la vue subsiste même si elle change de contenu)

11

Architecture Modèle-Vue-Contrôleur



12

Architecture Modèle-Vue-Contrôleur

- Intérêts de l'architecture MVC:
 - ◆ Indépendance entre :
 - la représentation logique d'une application (modèle)
 - la représentation visuelle qu'on en donne (vue)
 - les actions que l'utilisateur effectue (contrôleur)
 - ◆ Séparation claire entre les données du programme et l'interface graphique affichant ces données

13

Architecture Modèle-Vue-Contrôleur

- Intérêts de l'architecture MVC:
 - ◆ Possibilités de vues différentes d'un même modèle
 - L'application peut montrer l'état du modèle de différentes façons, avec différentes interfaces utilisateurs

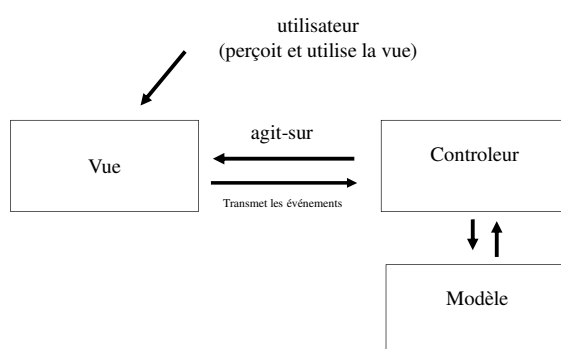
14

Architecture Modèle-Vue-Contrôleur

- ◆ Modularité dans la conception
 - vue et contrôleur peuvent être développés indépendamment du modèle (pourvu qu'une interface entre les deux soit définie)
- ◆ Meilleure répartition des tâches
 - développeurs du modèle/développeurs de l'interface ont des compétences différentes
 - développeurs du modèle : connaissance métier
 - développeurs de l'interface : connaissance des besoins utilisateurs, souci d'ergonomie...

15

Architecture Modèle-Vue-Contrôleur



16

Architecture Modèle-Vue-Contrôleur

- Les patrons de conception peuvent être utilisés :
 - ◆ le contrôleur est un sujet de la vue qui est un *observateur*
 - Plusieurs vues pour une même application sont possibles
 - ◆ la vue implémente le pattern stratégie

17