

Rappels Patron d'architecture MVC

Laëtitia Matignon

laetitia.matignon@univ-lyon1.fr

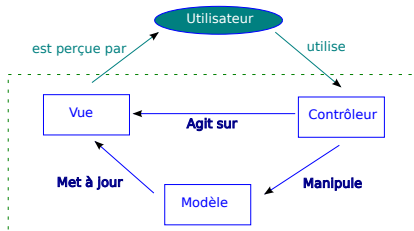
Département Informatique - Polytech Lyon
Université Claude Bernard Lyon 1
2014 - 2015

Architecture Modèle - Vue - Contrôleur

Séparer traitements, données et présentation.

Divise l'application en 3 composants :

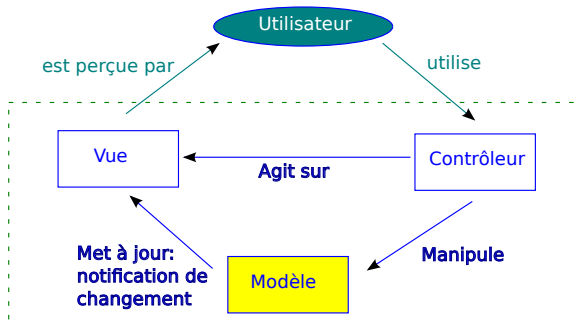
- côté applicatif : un modèle (ensemble des données du domaine et traitements)
- côté visuel : une (les) vue(s) (présenter/afficher les données du modèle dans l'interface utilisateur)
- côté événementiel : un contrôleur (reçoit les événements utilisateurs, déclenche les traitements)



Architecture Modèle - Vue - Contrôleur

Modèle

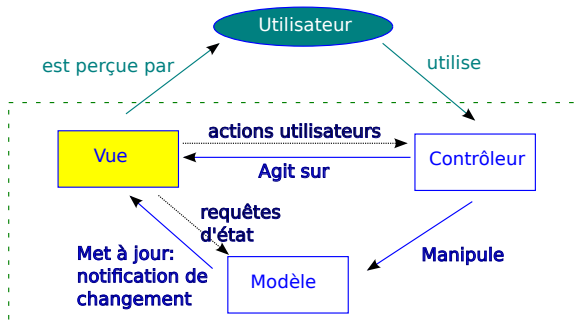
- indépendant de l'affichage
- le modèle peut être modifié sur ordre du contrôleur
- le modèle notifie ses vues associées au travers du patron *Observer*
→ modèle observé, vue observatrice



Architecture Modèle - Vue - Contrôleur

Vue

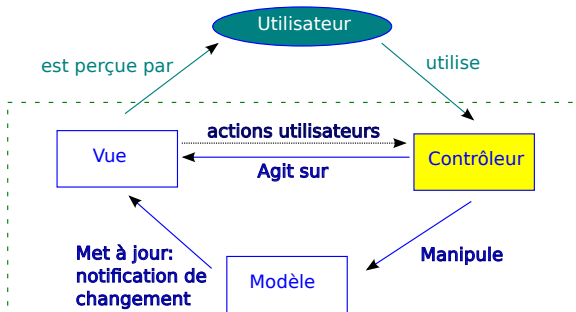
- affiche les données, capte les actions de l'utilisateur qui sont transmises au contrôleur associé à la vue (événement)
- mise à jour si notification du modèle ou du contrôleur → la vue peut faire des requêtes d'état sur le modèle
- plusieurs vues pour un même modèle possibles



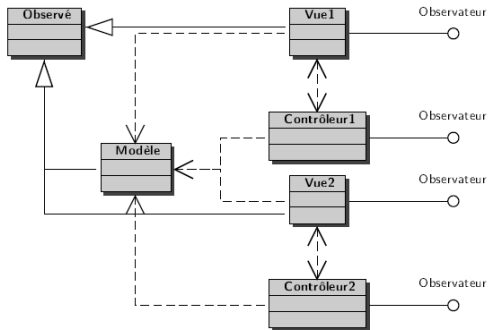
Contrôleur

Gestion de la synchronisation entre la vue et le modèle :

- rôle d'écouteur pour les événements utilisateurs captés par la vue
- définit la réponse aux événements utilisateur :
 - si changement des données, le contrôleur demande la modification des données au modèle → le modèle notifie en réponse les vues
 - si uniquement modification de la vue, le contrôleur notifie celle-ci directement



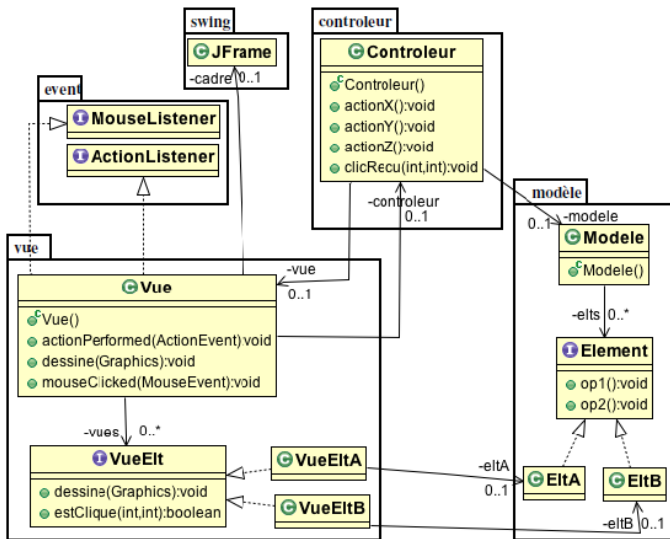
Modèle UML de l'architecture MVC



- Le modèle ne connaît ni les vues, ni les contrôleurs
- chaque vue est associée à un contrôleur
- la vue observe le modèle
- chaque contrôleur est associée à une vue (écouteur d'événements)
- chaque contrôleur connaît le modèle

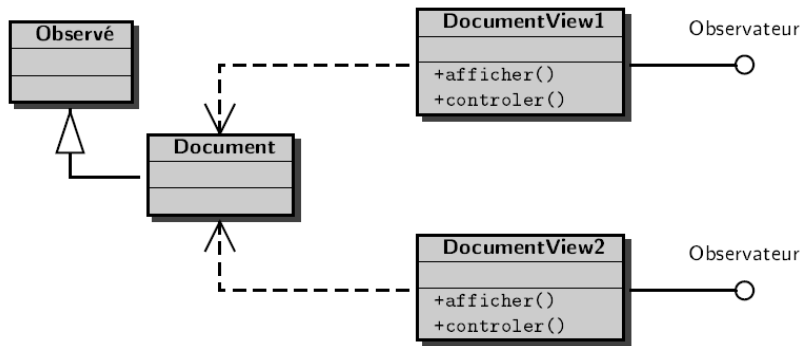
Modèle UML de l'architecture MVC

Exemple



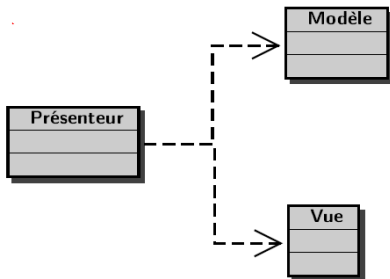
Architecture Modèle - (Vue / Contrôleur)

- architecture à deux éléments : la vue et son contrôleur sont fusionnés en un seul objet
- on appelle **Document** le modèle
- on appelle **Document View** le couple vue-contrôleur



Le motif observateur / observé n'est pas toujours utilisé

- la vue est découplée du modèle, le modèle est indépendant de la vue et du contrôleur
- la vue ne se met pas à jour automatiquement
- on appelle **présentateur** le contrôleur
- exemple : protocole http



- Indépendance entre la représentation logique, visuelle et les actions effectuées
- Séparation claire entre les données et leur affichage

Avantages

- Couplage faible avec séparation modèle-vue
- Cohésion : chaque composant se concentre sur une fonctionnalité
- Flexibilité : Il est facile de changer l'interface utilisateur
- Testabilité : On peut tester l'application indépendamment de l'interface

Inconvénients

- complexité inutile pour des interfaces simples
- dépendance forte de la vue et du contrôleur envers le modèle
- possibilités de mises à jour excessives

→ Variantes avec fusion vue - contrôleur