

**TELECOM**  
ParisTech



Institut  
Mines-Télécom

# MVC

## Un modèle de conception

Workflow et outils



# MVC, en bref

**Une façon d'organiser un programme pour qu'il soit évolutif (améliorations, maintenance, ...)**

## ■ **Modèle(s)**

- Le modèle des données sur lesquelles votre application travaille

## ■ **Vue(s)**

- La façon dont on voit et on interagit sur ces données

## ■ **Contrôleur(s)**

- La relation entre les deux, le reste

# MVC, le modèle

## ■ Modèle(s)

- Le cœur de l'application
- Le modèle des données sur lesquelles votre application travaille
- Des parties du code qui ne font que
  - Créer/supprimer une donnée,
  - Sauver un ensemble de données,
  - Relire un ensemble de données,
  - Mettre à disposition une donnée

Sur les catégories de données manipulées

# MVC, les vues

## ■ Vue(s)

- La façon dont on voit et on interagit sur les données de l'application
- Présente une 'vue' sur les données
- Reçoit les actions de l'utilisateur et induit les messages correspondants vers le contrôleur
- La vue ne fait pas (ou peu) de traitement
  
- On a souvent plusieurs vues différentes
  - En fonction de l'utilisateur
  - En fonction de l'étape de l'application

# MVC, le contrôleur

## ■ Contrôleur(s)

- La relation entre les deux, le reste
- En particulier, responsable de tous les messages permettant la synchronisation entre la vue et le modèle
  - Ex: s'il reçoit un message du modèle indiquant une modification de donnée, demande à la vue de se mettre à jour
  - Ex: s'il reçoit un message de la vue indiquant une saisie d'une donnée, demande au modèle la modification correspondante

# MVC, l'essentiel

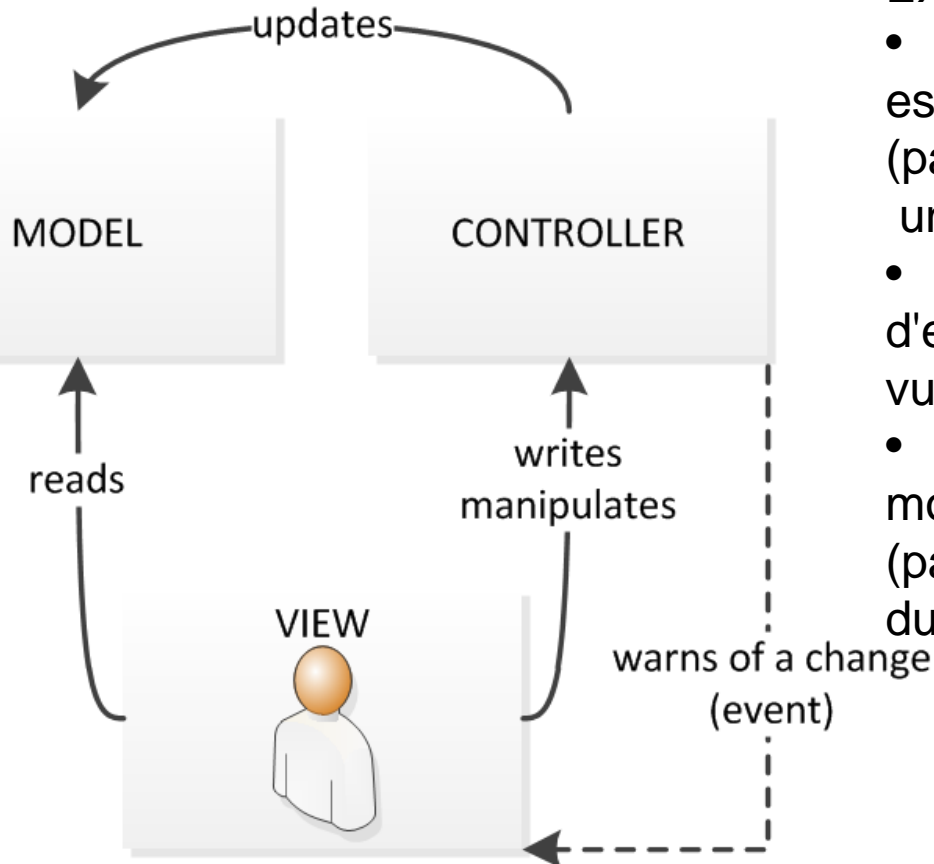
## ■ Bien séparer

- La ou les vues
- Le modèle

## ■ Pourquoi

- Aide à faire évoluer une vue sans toucher au modèle
- Aide à faire évoluer le modèle sans toucher aux vues

# Principe



## Exemple

- une requête envoyée depuis la vue est analysée par le contrôleur (par exemple un clic de souris pour lancer un traitement de données)
- le contrôleur demande au modèle d'effectuer les traitements et notifie à la vue que la requête est traitée
- la vue notifiée fait une requête au modèle pour se mettre à jour (par exemple affiche le résultat du traitement "via" le modèle).



# Avantages

- **Evolutivité**
- **Changer une vue ne change rien au modèle**
- **Changer le modèle ne change pas les vues**
  - Ex: passer d'un fichier XML pour stocker les données à une base de données



# MVC est un principe...

- ... qui connaît de nombreuses déclinaisons et variantes
- **Idée forte**
  - Bien séparer l'affichage des traitements, notamment de la gestion de base des données manipulées, pour pouvoir faire évoluer les uns indépendamment des autres