

Conduite et Gestion de Projet

Introduction à l'architecture MVC

Thierry Hamon

Bureau H202 - Institut Galilée

Tél. : 33 1.48.38.35.53

Bureau 150 – LIM&BIO – EA 3969

Université Paris 13 - UFR Léonard de Vinci

74, rue Marcel Cachin, F-93017 Bobigny cedex

Tél. : 33 1.48.38.73.07, Fax. : 33 1.48.38.73.55

thierry.hamon@univ-paris13.fr

<http://www-limbio.smbh.univ-paris13.fr/membres/hamon/CGP-20132014>

Que faire lorsqu'on veut une application interactive ? (exemple : page web)

- Architecture simpliste : regroupement du code, des données et de la visualisation dans une même page
 - Problème de duplication des données, des visualisations et des traitements
 - Manque de flexibilité (un seul développeur peut intervenir sur la page, la modification de la visualisation peut entraîner une modification du code et des données)

Deux grandes parties :

- la partie visible (*front office*) : l'IHM (Interface Homme-Machine

Mais en fait deux parties :

- les entrées : actions de l'utilisateur
 - les sorties : la visualisation des « résultats de l'action de l'utilisateur »
- la partie caché (*back office*) : le stockage, l'accès et le traitement des données
→ cœur de l'application

Décomposition de l'application en trois parties : le Modèle, la Vue, le Contrôleur



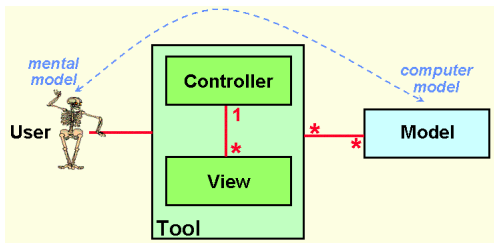
- Proposée en 1979 par Trygve M. H. Reenskaug
<http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>
- Fortement liée à la programmation objet et maintenant aux applications Web (notamment J2EE, avec des Frameworks adaptés à chaque cas – struts, Hibernate)
- Patron de conception (*design pattern*), c'est-à-dire une solution standard pour répondre à un problème
- Architecture logicielle pour structurer une application

Différencier

- ce qui est fait (le contrôleur)
- ce qu'on voit (la vue)
- ce qui se passe (le modèle)



Solution Modèle-Vue-Contrôleur



→ donner l'impression à l'utilisateur qu'il manipule les données du modèle

Réponse partielle aux problèmes liées à la complexité des applications interactives et au développement d'interfaces graphiques

Organisation et structuration d'une application en séparant :

- Les données et leurs traitements : Le **Modèle**
- La visualisation des données : La **Vue**
- Le comportement de l'application : Le **Contrôleur**



Fonctionnalités de l'application :

- Représentation des données
- Accès aux données
- Traitement des données

Sorties de l'application

- Visualisation(s) des données issues du Modèle
- Assure la consistance de la visualisation avec la représentation des données dans le modèle

Gestion des entrées de l'application

- Comportement de l'application face aux actions de l'utilisateur (programmation événementielle)
- Traduction des actions de l'utilisateur en action sur le modèle
- Traduction des résultats des traitements du modèle (réactions) et des actions de l'utilisation en action sur la vue (modification appropriée de la vue)

Flux de traitement

Modèle

Fonctionnalités de
l'application
Accès aux données
et traitement

Vue

Présentation des données à
l'utilisateur

Contrôleur

Gestion des entrées de
l'utilisateur
Définition du comportement
de l'application

Flux de traitement

Modèle

Fonctionnalités de
l'application
Accès aux données
et traitement

Vue

Présentation des données à
l'utilisateur

Contrôleur

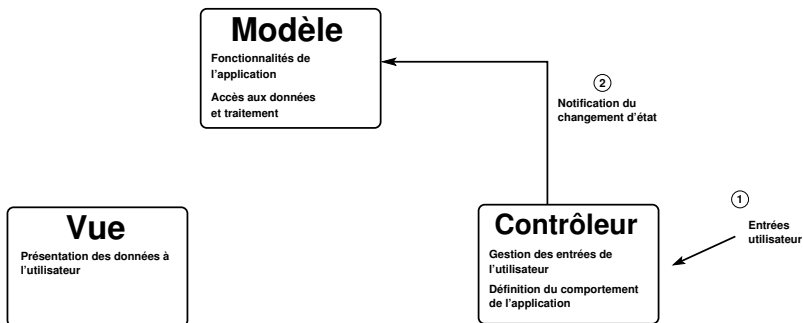
Gestion des entrées de
l'utilisateur
Définition du comportement
de l'application

①

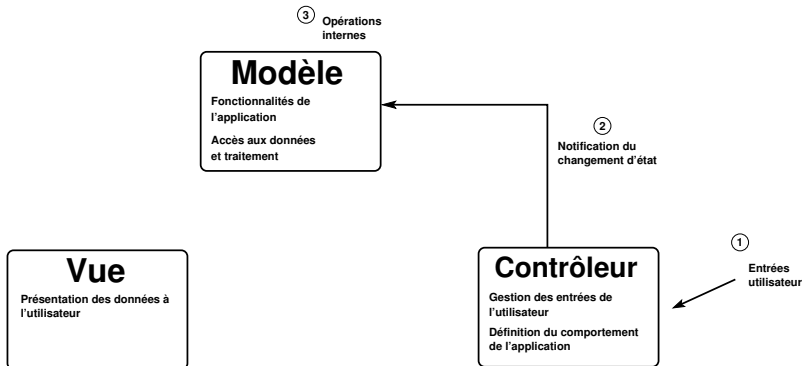
Entrées
utilisateur



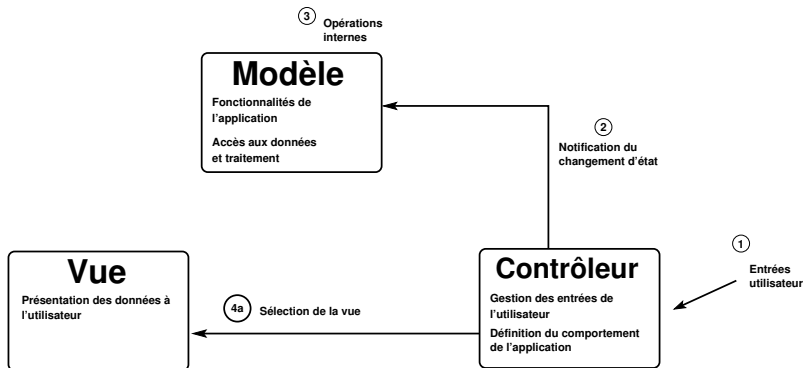
Flux de traitement



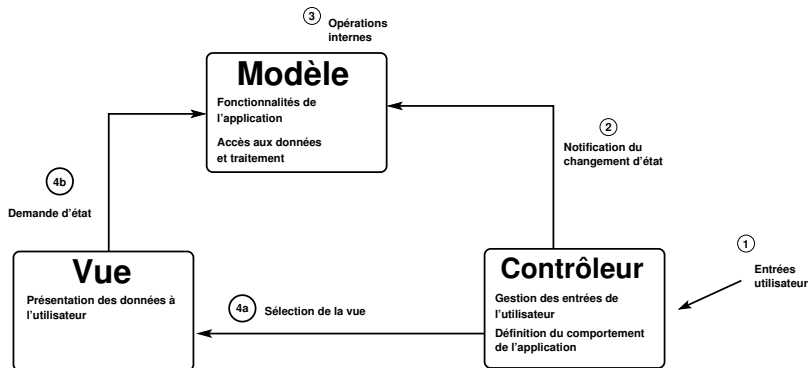
Flux de traitement



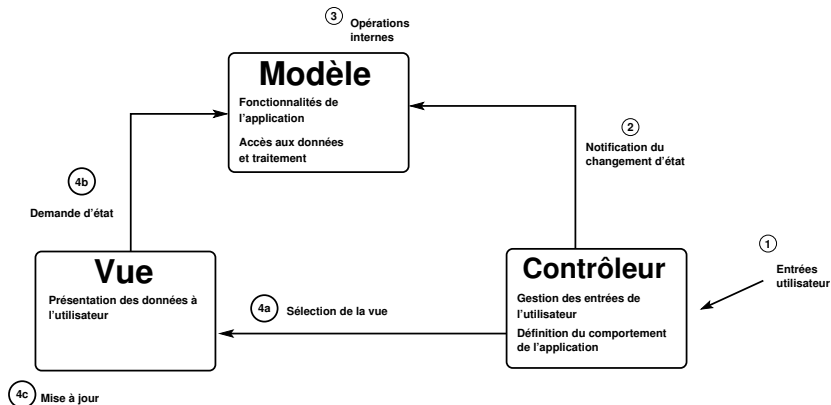
Flux de traitement

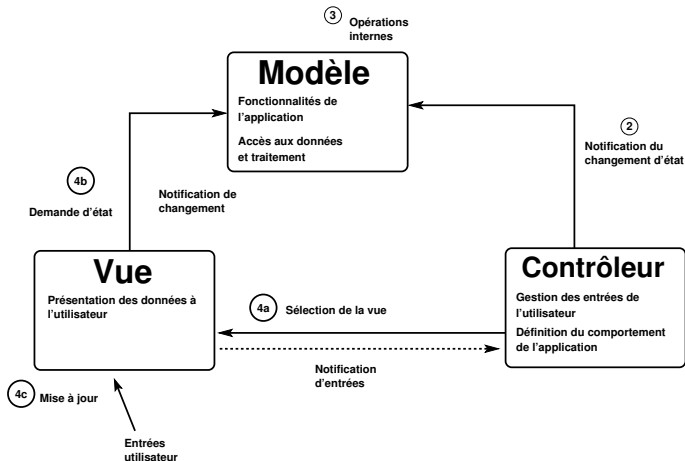


Flux de traitement

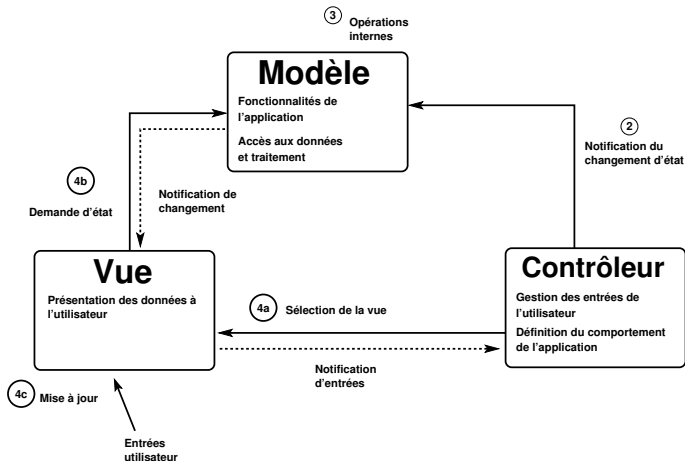


Flux de traitement





Possibilité d'interaction entre la vue et le modèle s'il n'y a pas de modification (sinon il faut obligatoirement passer par le contrôleur)



- Structuration de l'application
- Indépendance entre les données, la visualisation et le comportement de l'application
- Modularité et réutilisation des éléments :
 - Plusieurs vues possibles
 - Changement/échange de contrôleurs, contrôleurs multiples
 - Changement de représentation ou de moyens de/stockage des données au niveau du modèle
- Solution particulièrement adaptée à la programmation orientée objet
- Synchronisation vue/modèle à travers le contrôleur

Inconvénients

- Complexité accrue
- Mises à jour potentiellement trop nombreuses
- Augmentation du temps d'exécution
- Le contrôleur et la vue sont souvent liés au modèle



MVC : Outil

- d'analyse d'un problème
- de structuration d'application interactive
- pour l'implémentation d'un système modulaire, flexible et réutilisable

Mais aussi :

- Indépendance entre l'IHM et les données et leurs traitements
- Facilite la maintenance et la réutilisation de modules

Autre solution : architecture trois-tiers



Modèle en couche : chaque couche interagit avec les couches adjacentes



Sources des transparents

- Stéphane Huot, Département informatique, IUT d'Orsay
- Jean-Michel Richer, université d'Angers
- F. Michel

