

Interaction Homme-Machine

Anastasia.Bezerianos@lri.fr

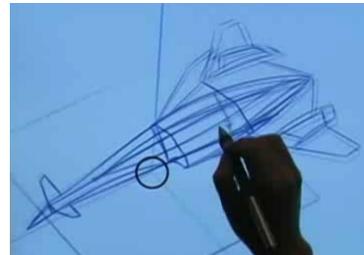
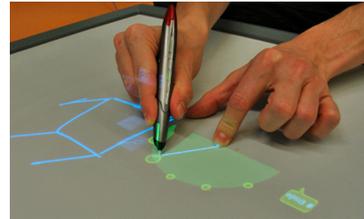
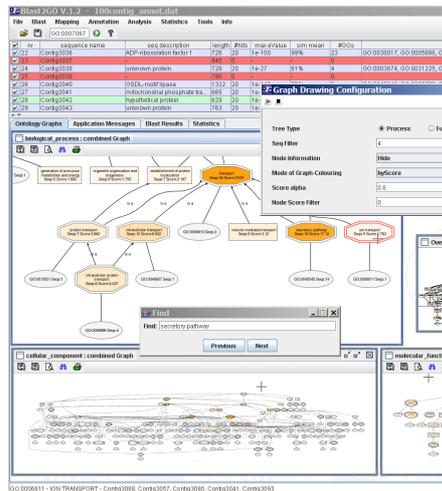
(partie de la présentation basée sur des transparents de M. Beaudouin-Lafon) et N.Rousel

Cours 3: Périphériques, Architectures Logicielles & MVC

Anastasia.Bezerianos@lri.fr

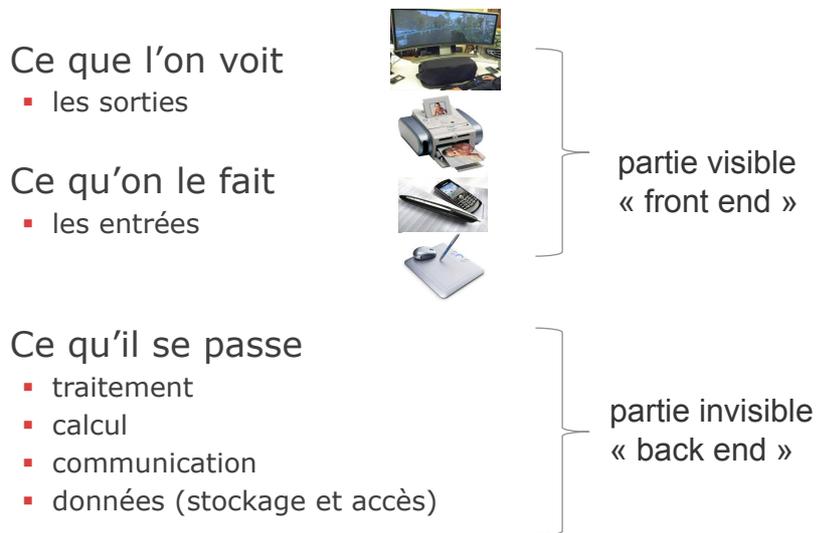
(basée sur des transparents de Olivier Chapuis et de Stéphane Huot)

ystème interactif



périphériques

structure d'un système interactive



structure d'un système interactive

Ce que l'on voit (les sorties)

Ecrans "bitmap" Cathodique, LCD, Plasma, OLED.

Taille exprimé en la dimension de la diagonal en pouce

(1 pouce = 2.54 cm, 30 pouces ~ 76cm) et le ratio largeur sur hauteur (e.g., 16/9)

Résolution exprimé en pixel (e.g., 2560x1600).

La taille plus cette résolution donne une densité exprimée en général en

"dpi": dot[pixel] per[par] inch[pouce]

(typiquement 100 dpi ~ 40 pixels par cm, c.a.d. 1 pixel ~ 0.25 mm)

Résolution en couleurs ("profondeur" RGB[A]):

8 bits (256 couleurs), 16 bits (65536 couleurs) ou 24[32] bits (16 millions de couleurs [+256 niveaux de "translucence"])

Résolution temporelle exprimé en Hz qui est le nombre de trame

[frame] par seconde que l'écran peut afficher (typiquement 60 Hz)

structure d'un système interactive

Ce que l'on voit (les sorties)



<http://youtu.be/u7Gm0OeKxwU>

structure d'un système interactive

Ce qu'on le fait (les entrées)

- claviers, boîtes à boutons
- potentiomètres (rotatifs, linéaires)
- souris, tablettes, joystick, trackball
- stylos, crayons optiques
- reconnaissance de la parole
- capteurs de position et de direction
- vision par ordinateur
- surfaces interactives
- dispositifs hybrides entrée/sortie
 - Écrans tactiles
 - Retour d'effort



<http://youtu.be/REA97hRX0WQ>

entrée de texte

Ce qu'on le fait (entrée de texte)

- Un problème: Optimisation de la position des touches
- Qwerty, mais disposition Dvorak: 10 a 15% de plus de vitesse, réduction de la fatigue
- Clavier Logiciel: optimisation → pointage
- Clavier dpy avec des touches qui possèdent des écrans (oled) ou projection sur le clavier <http://youtu.be/fhBH6KW2aT4>



entrée de texte

Ce qu'on le fait (alternative au clavier classique)

- Clavier à accords (chord keyboards):
 - peu de touches (4 ou 5)
 - utilisation de plusieurs touches simultanément
 - saisie rapide avec une seule main
- Téléphone portable:
 - tape multiple
 - tape plus ou moins long
 - Système T9: un seul "tape" par lettre le système découvre le mot



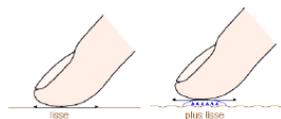
périphériques 3D, tactile

Ce qu'on le fait

- Périphériques 3D



- Retour tactile: vibrations et surface plus ou moins dur.
 - Vibrations lors que l'on passe sur certaines cibles, peut-on reproduire des vraies textures ?



type de contrôle et de périphérique

Absolus : transmettent une position (x, y)

Exemples : tablettes, écrans tactiles et stylos optiques

Relatifs transmettent un déplacement (dx, dy)

Exemples : souris, joysticks et trackballs

Ordre 0: déplacement du périphérique d'entrée correspond à un déplacement d'un objet

Exemple : couple souris - curseur

Ordre 1: le périphérique d'entrée contrôle la vitesse d'un objet

Exemple : couple joystick - curseur

Périphérique isotonique: Contrôle de position – Ordre 0, souvent débrayage pour les longues distances

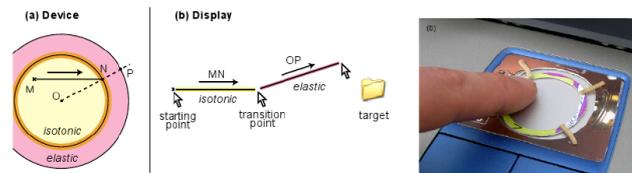
Exemples : souris, touchpad

Périphérique élastique: Possède un état stable avec retour « élastique » à cet état

Exemples : joystick (inclinaison → vitesse)

une périphérique isotonique & élastique

RubberEdge (Casiez et al. 2007): réduction du débrayage en combinant un contrôle positionnel isotonique et élastique



Centre du touchpad: position
Bords du touchpad: un système élastique pour contrôler la vitesse de déplacement

http://youtu.be/kucTPG_zTik

per. d'entrée : Control – Display Gain

Résolution : nombre d'impulsion que peut envoyer le périphérique d'entrée pour une distance donnée
Souvent exprimé en

$$\text{dpi} = \frac{\text{dot}[\text{impulsion}]}{\text{per}[\text{par}]} \text{ inch}[\text{pouce}]$$

[1 pouce = 2.54 cm]

Exemple pour une souris: entre 300dpi (lent), 600dpi (typique) et 2400dpi (max?), soit, respectivement une impulsion tous les 0.083, 0.042 et 0.01 mm

Contrôle-Display Gain [Gain contrôle-écran]:

1 pouce souris = 600 dpi =
600 pixels de mouvement =
6 * 100 dpi résolution écran =
6 pouces écran

$$CDGain = \frac{\text{Distance parcourue à l'écran par le pointeur}}{\text{Distance parcourue par le périphérique d'entrée}}$$

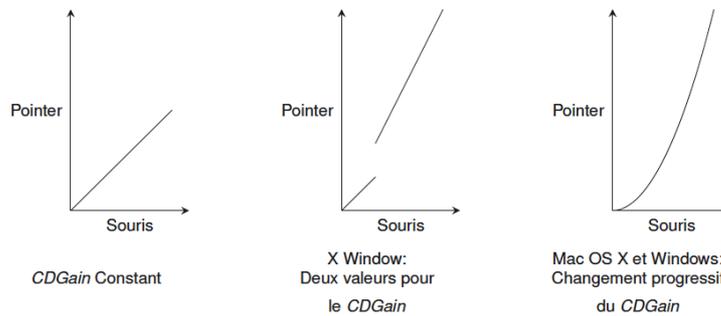
Exemples:

Tablette de même taille que l'écran avec contrôle direct tablette écran: $CDGain = 1$.
Ecran 100 dpi, souris 600 dpi, une impulsion souris traverse un pixel: $CDGain = 6$.

per. d'entrée : accélération

Problème: Si le CDGain est trop grand il est difficile d'être précis, ' mais si il est trop petit on est obligé de débrayer la souris pour parcourir de "grande" distance

Accélération: modification dynamique du CDGain en fonction de la vitesse de la souris. Plus on va vite, plus le CDGain devient grand



pointage: loi de Fitts



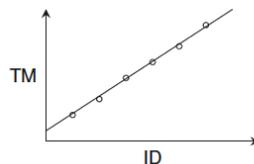
$$TM = a + b.ID \quad \text{où}$$

$$ID = \log_2\left(\frac{D}{W} + 1\right)$$

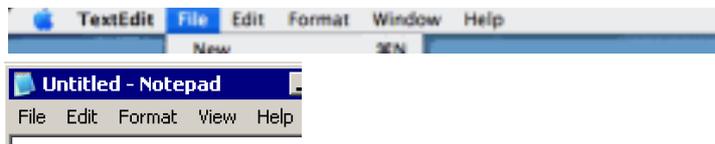
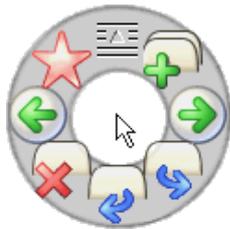
MT: Temps de Mouvement

ID: Indice de Difficulté (de la tâche)

a, *b*: constantes empiriques (qui dépendent des utilisateurs, du périphérique ...etc.)



pointage: loi de Fitts

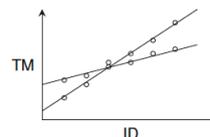
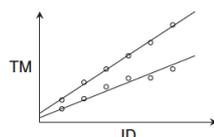


pointage: loi de Fitts et IHM

Design d'interface (clavier logiciel, interface adapté à différents types d'utilisateurs, ...)

Technique de Pointage: réduire D ou agrandir W

Pour comparer les performances entre périphériques d'entrée (souris, touchpad, tablette ...)



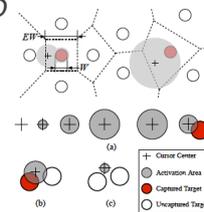
La pente de la droite comme un indice de performance

techniques de pointage

“Battre la Loi de Fitts” en réduisant D ou agrandissant W



- Agrandir la taille des cibles lorsque le curseur se trouve dessus
(Mac OS X Doc) <http://profs.loqti.etsmtl.ca/mmcquuffin/research/expandingTargets/>
- Agrandir W dans l'espace moteur [Blanch et al.]
<http://iihm.imag.fr/blanch/projects/semantic-pointing/>
- Prédiction de la position de la cible pour réduire D
<http://patrickbaudisch.com/projects/dragandpop/index.html>
- Bubble cursor [Grossman & Balakrishnan]
http://youtu.be/JUBXkD_8ZeQ
- DynaSpot [Chapuis & al.]
<http://youtu.be/itb1y5PMEOc>



architectures logicielles, MVC

structure d'un système interactive

Ce que l'on voit

- les sorties

Ce qu'on le fait

- les entrées



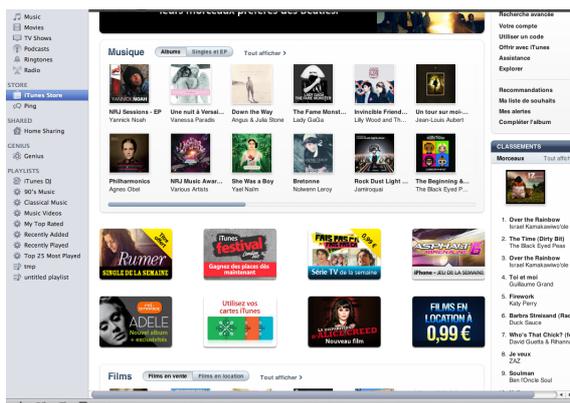
partie visible
« front end »

Ce qu'il se passe

- traitement
- calcul
- communication
- données (stockage et accès)

partie invisible
« back end »

exemple 1



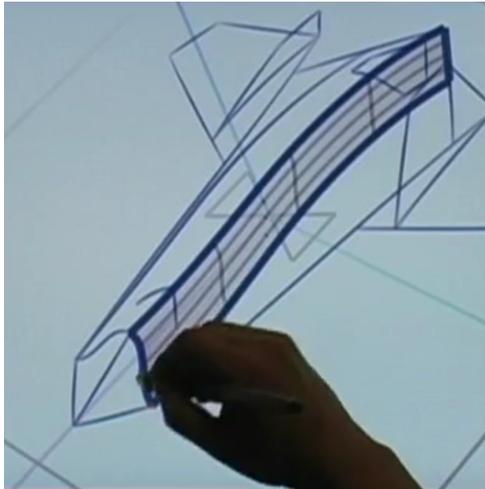
↔

- modèle de données (albums, artistes, catégories, etc.)
- communication avec le serveur de iTunes
- traitement de requêtes
- traitement des achats
- sécurité

back end

front end

exemple 2



front end

⇔

- modèles géométriques
- calcul (transformations, rendu graphique, etc.)
- stockage et accès aux dessins

back end

exemple 3

id	Name	Email	Address
6825	JUANITA LAMBERT	mbeasley@everymail.biz	139 MANNING HWY,CLYO,76604
5740	GREG CABRERA	cholder@btmail.biz	736 GENESSEE BLVD,CORDELE,17433
8699	ALISSA WISE	thweb@eyecode.net	205 ALICE RD,CAMILLA,14865
9282	SHARON WINTERS	jogang@holmafl.com	955 COHEN PIKE,TYRONE,811
2150	KRISTY FRANKS	jgates3@somemafl.com	1471 ALEXIS PKWY,BALDWIN,85
9927	JEFF RICE	die-dots@fornail.org	104 DUNDEE PKWY,HOGANSVILLE,3741
7872	TAMARA BRYANT	holwom@everymail.us	1382 WOGAN BLVD,CITY OF CATHOLIN,43790
5824	ALISHA YANG	foundwrong@holmafl.net	716 HOGANS DR,HARDING,58932
3402	JASON NGUYEN	havothers@ma12u.com	527 MICHAEL CRES,FORT STEWART,14664
3620	LINDSEY CABRERA	ashedreams@tmail.com	1429 LAZELERE HTS,FORT STEWART,21650
3511	ANTHONY MATHEWS	cfarrell@everymail.co.uk	1325 OKEY LN,THUNDERBOLT,63656
572	JARED FORD	ortrying@btmail.co.uk	722 EUCLID RD,FORSYTH,42014
9720	AUTUMN WILLIAMS	roomwhere@holmafl.net	260 HILL PARK,NORCROSS,58355
3447	MARION BROWN	hwagner@tmail.co.uk	739 MIDDLE PATH,HACON,5944
9356	HOPE HAYNES	saacrific@eyecode.com	1023 COOPERRIDERS CRES,STATENVILLE,342
2259	JEANNIE RANDOLPH	wormto@eyecode.net	672 EDISON PATH,CENTERVILLE,48580
8264	RACHAEL CONLEY	smokewhite2@eyecode.net	1223 STEVENS CT,CARROLLTON,40084
4224	TRACIE DAVENPORT	whitakerf@eyecode.net	549 TALLEY RUN,BRISTOL,53113
1836	NANCY GOFF	holdfire@gsmafl.us	147 FLEMING HWY,MORA,37823
6284	KATHY MORENO	hadcode@everymail.co.uk	1341 BYRNE DR,SUMNER,80450

front end

⇔

- structure tabulaire
- stockage et accès aux données

back end

liaison entre les deux parties **...programmation selon un modèle organisé**

organiser, structurer une application interactive en séparant :

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

Le modèle «Modèle–Vue–Contrôleur » (MVC)

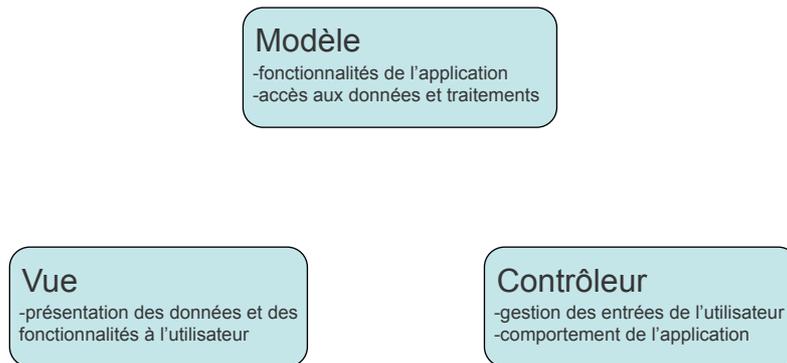
Le MVC est :

- Un *patron de conception* (solution de conception standardisé indépendante du langage de programmation)
- Une *architecture logicielle* (une manière de structurer une application ou un ensemble de logiciels)

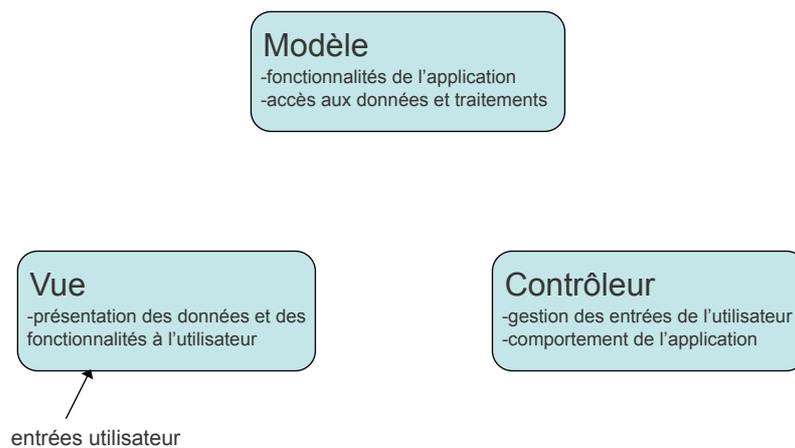
Introduit le 1979 par Trygve Reenskaug

Fortement lié aux concepts de la programmation objet (Smalltalk)

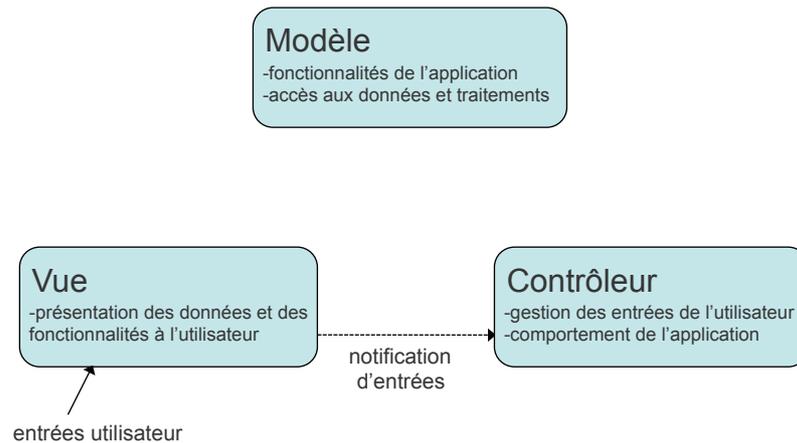
MVC : interactions entre composants



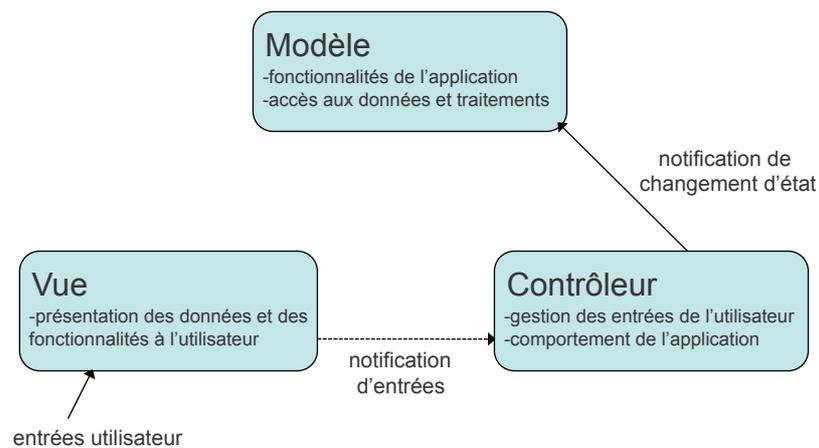
MVC : interactions entre composants



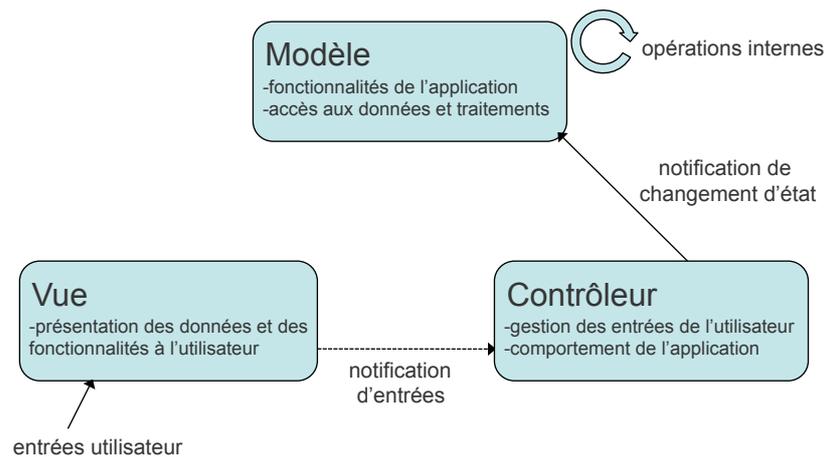
MVC : interactions entre composants



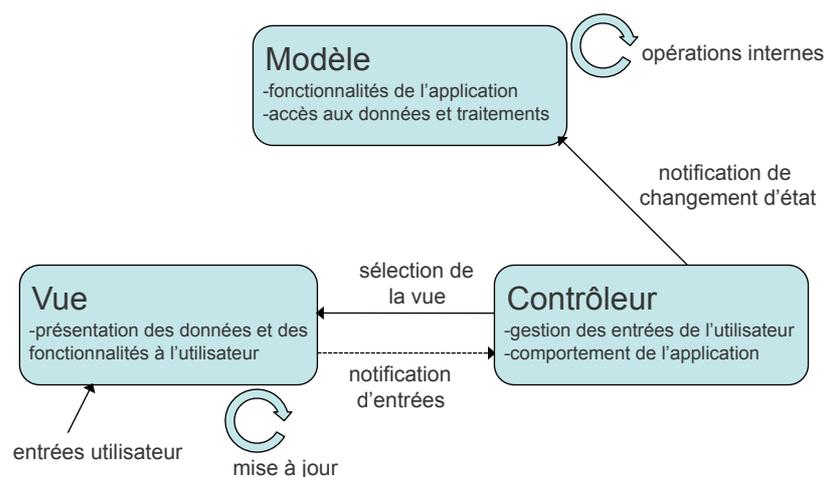
MVC : interactions entre composants



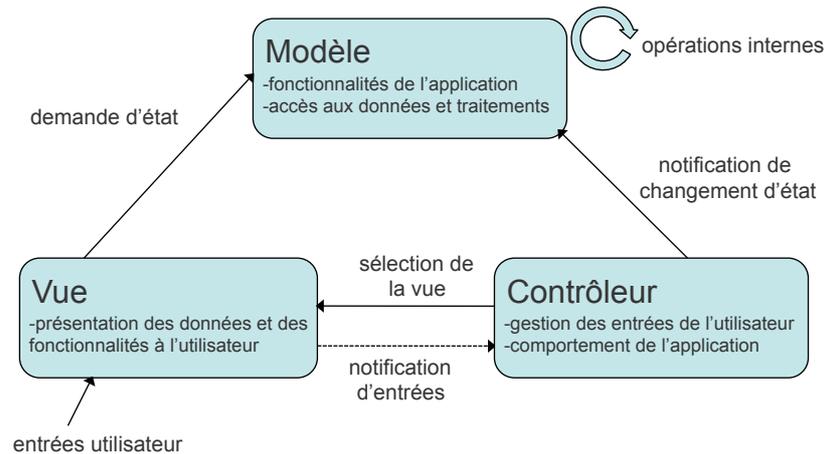
MVC : interactions entre composants



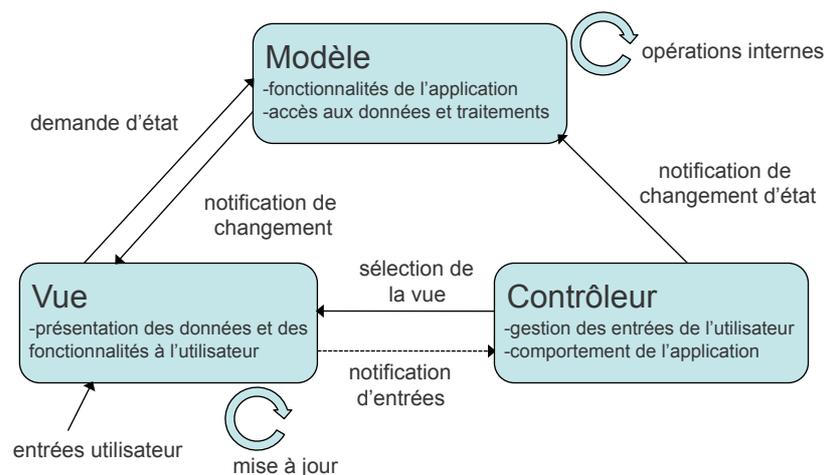
MVC : interactions entre composants



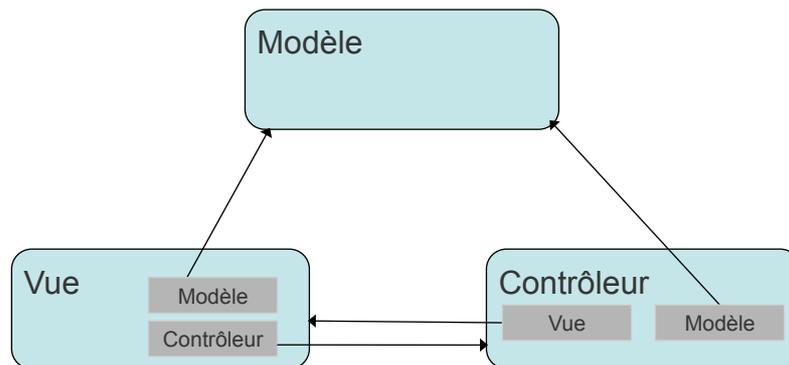
MVC : interactions entre composants



MVC : interactions entre composants



MVC : références entre composants



MVC : le modèle

Le modèle:

- Représente les données
- Fournit les accès aux données
- Fournit les traitements applicables aux données
- Expose les fonctionnalités de l'application

Noyau Fonctionnel de l'application

MVC : la vue

La vue:

- Représente la (ou une) représentation des données du modèle
- Assure la consistance entre la représentation qu'elle donne et l'état du modèle/le contexte de l'application

Sorties de l'application

MVC : le contrôleur

Le contrôleur:

- Représente le comportement de l'application face aux actions de l'utilisateur
- Fournit la traduction des actions de l'utilisateur en actions sur le modèle
- Fournit la vue appropriée par rapport aux actions de l'utilisateur et des réactions du modèle

Comportement et gestion des entrées de l'application

avantages de MVC

Structure "propre" de l'application

Adapté aux concepts de la programmation O-O

Indépendance données – représentation – comportements

Modulaire et **réutilisable**

inconvénients de MVC

Mise en place complexe dans le cas d'applications importantes

Mises à jour potentiellement trop nombreuses

- « Spaghettis » dans le code

Contrôleur et Vue restent souvent fortement liés au Modèle



adapter la réalisation au problème

MVC et Java Swing

La séparation Modèle-Vue-Contrôleur n'est pas stricte

Les widgets de base

- **Modèle** : le comportement abstrait du widget
- **Vue et Contrôleur** : Look & Feel + Listener

Exemples : JButton, JLabel, JPanel, etc.

La plupart des fois, on ne touche pas le modèle des widgets

- Swing utilise un modèle par défaut pour chaque widget

Swing : types de modèles

Look & Feel

- Interfaces : ButtonModel, ListSelectionModel
- Classes par default :
DefaultButtonModel, DefaultListSelectionModel

Données

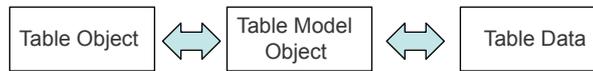
- Interfaces : ListModel, TableModel, TreeModel
- Classes par default :
DefaultListModel, DefaultTableModel, DefaultTreeModel

Look & Feel + Données

- Pour certains widgets
- Exemples : BoundedRangeModel pour JSlider

exemple

First Name	Last Name	Sport	# of Years	Vegetarian
Kathy	Smith	Snowboarding	5	false
John	Doe	Rowing	3	true
Sue	Black	Knitting	2	false
Jane	White	Speed reading	20	true
Joe	Brown	Pool	10	false



javax.swing.JTable *javax.swing.table.TableModel*

exemple

Les données

```

Object[][] data = {
    {"Kathy", "Smith", "Snowboarding", new Integer(5), new Boolean(false)},
    {"John", "Doe", "Rowing", new Integer(3), new Boolean(true)},
    {"Sue", "Black", "Knitting", new Integer(2), new Boolean(false)},
    {"Jane", "White", "Speed reading", new Integer(20), new
        Boolean(true)},
    {"Joe", "Brown", "Pool", new Integer(10), new Boolean(false)}
};
  
```

exemple

Le modèle

```
class MyTableModel extends AbstractTableModel {
    private String[] columnNames = ...
    private Object[][] data = ...

    public int getColumnCount() {
        return columnNames.length;
    }

    public int getRowCount() {
        return data.length;
    }

    public String getColumnName(int col) {
        return columnNames[col];
    }

    public Object getValueAt(int row, int col) {
        return data[row][col];
    }
    ...
}
```

exemple

La vue

```
TableModel dataModel = new MyTableModel();
JTable table = new JTable(dataModel);
JScrollPane scrollpane = new JScrollPane(table);
```