CENTRALE
L Y O N

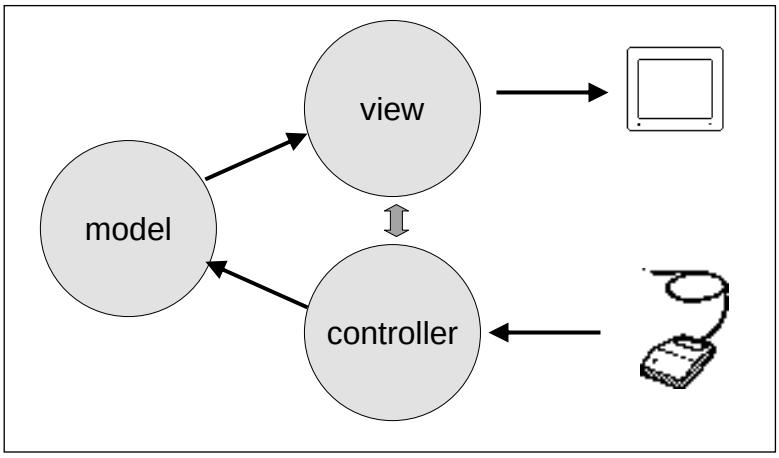
Framework STRUTS, MVC – MVC II

BTD/MAI/MVC 1

CENTRALE
L Y O N

Bertrand DAVID : Méthodologies avancées d'informatisation

MVC model - view - controller



BTD/MAI/MVC 2

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Model-View-Controller

- **Inventé en Smalltalk, vers 1980**
- **Idée : séparer les sortie de présentation (View), de la gestion des entrées utilisateur (Controller) et de la sémantique applicative (Model).**
- **Assez simple dans les principes, mais difficile à mettre en oeuvre**
- **Jamais clairement expliqué (un seul article difficile à lire)**
- **Buts**
  - ➔ **Programmer un nouveau Modèle et réutiliser les Vues et Contrôleurs existants**
  - ➔ **multiples, différents types de Vues sur le même Modèle**

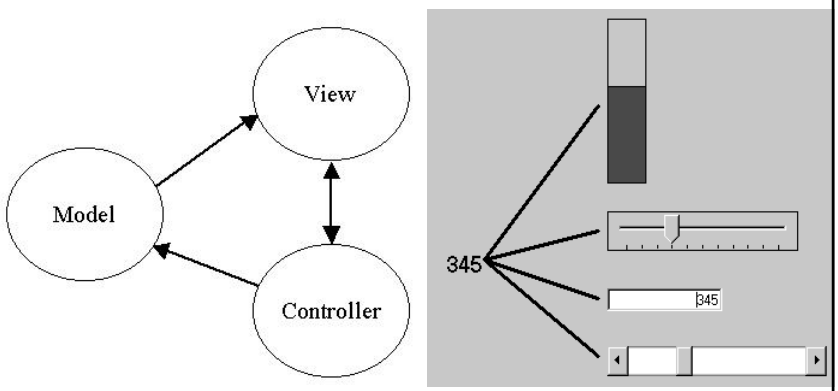
BTD/MAI/MVC

3

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

MVC



BTD/MAI/MVC

4

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

MVC

- **Les Vues sont étroitement associées aux Contrôleurs.**
- **Chaque VC a un M; un M peut avoir plusieurs VCs.**
  - ➔ **VCs ont une certaine connaissance explicite des Modèles, mais M n'en ont pas sur les Vues.**
  - ➔ **Les changements dans les Modèles sont diffusés à tous les dépendants de modèle par un protocole standard.**

BTD/MAI/MVC 5

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

MVC

- Modèle
  - ➔ **Simple comme un entier pour un compteur ; chaîne de caractères pour un éditeur**
  - ➔ **Complexe comme un simulateur nucléaire**
- Vues
  - ➔ **Tout est graphique**
  - ➔ **Organisation spatiale, sous-vues, compositions**
- Contrôleur
  - ➔ **Organisation des interactions avec les autres VCs**
  - ➔ **Un menu est un Contrôleur**

BTD/MAI/MVC 6

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

MVC

- Le cycle d'interaction standard :
  - ➔ **L'utilisateur utilise des dispositifs d'entrée, le contrôleur informe le modèle pour effectuer les changements, le modèle diffuse les notifications de changements aux vues associées, les vues changent l'affichage.**
  - ➔ **Les vues peuvent interroger le modèle.**
- Problèmes**:**
  - ➔ **Les vues et les contrôleurs sont étroitement couplés**
  - ➔ **Qu'est-ce qu'on met dans chacun ?**
  - ➔ **LA complexification des vues avec leurs parties, des contrôleurs avec les sous-contrôleurs, des modèles avec les sous-modèles.**

BTD/MAI/MVC 7

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Model-View

- **Comme in est difficile de séparer la Vue et le Contrôleur le modèle Modèle-Vue a été proposé et est utilisé dans Andrew, InterViews**
- **But principal : supporter des vues multiples des mêmes données.**
  - ➔ **Par un aiguilleur simple on pout voir les données différemment.**
- **Mettre dans le Modèle des parties qui doivent être sauvées dans un fichier"**
  - ➔ **Mais réellement il est important de sauver des parties des vues**

BTD/MAI/MVC 8

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

MVC

- **MVC divise l'application en :**
  - **➔Modèle de fonctionnalités et de données**
  - **➔Vues affichant des informations à l'utilisateur**
  - **➔Contrôleurs gérant les entrées utilisateur**
- **Vues and Contrôleurs constituent l'IHM**
- **Le mécanisme de propagation de changements assure la consistance entre le Modèle et l'IHM**

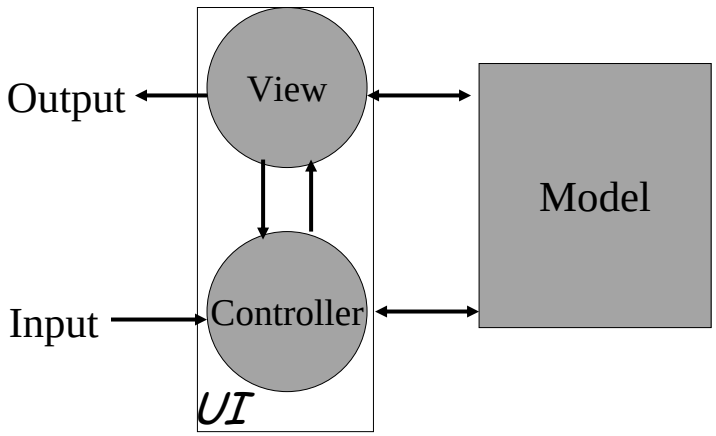
BTD/MAI/MVC 9

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

La triade MVC

- **Chaque élément est un objet**



BTD/MAI/MVC 10

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Modèle

- **Encapsule les données spécifiques à l'application et les fonctionnalités en fournissant :**
  - **➔méthodes pour éditer des données, que le Contrôleur peut appeler**
  - **➔méthodes pour accéder à l'état que la Vue et le contrôleur peuvent interroger**
- **Maintienne le registre de Vues et Contrôleurs associés pour leur notifier les changements des données**

BTD/MAI/MVC 11

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Exemples de Modèle

- **Editeur de texte : une chaîne de caractères est un modèle**
- **Ascenseur : un entier en constitue le modèle**
- **Excel : collection de valeurs reliées par des contraintes fonctionnelles**

BTD/MAI/MVC 12

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Vue

- **Mécanisme qui projette les données du modèle en présentation (vue / affichage)**
- **Quand le Modèle change, la Vue est informée**
  - **➔La vue demande des informations pertinentes venant du Modèle**
  - **➔La vue se charge de mettre à jour l'affichage**
    - **Déclare les zones à changer**
    - **Réaffiche sur demande**

BTD/MAI/MVC 13

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Exemples de Vue

- **Ascenseur : zone de texte, ligne avec poignet, jauge**
- **Excel :**
  - **➔Représentation tabulaire**
  - **➔Histogramme**

BTD/MAI/MVC 14

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Contrôleur

- Accepte les évènements d'entrée de l'utilisateur
- Traduit ces évènements en méthodes appelant le Modèle
- Active/Désactive des éléments d'IHM (griser)

BTD/MAI/MVC 15

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Exemples de Contrôleurs

- Commandes textuelles
- Commandes souris (pointage et click)
- Pas d'entrée

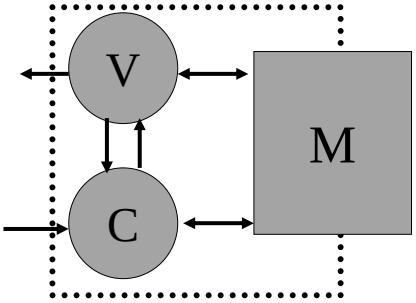
BTD/MAI/MVC 16

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Dynamique MVC

- **1. L'entrée utilisateur est acheminée par le système (Windows) vers le contrôleur approprié.**
- **2. Le Controller peut demander à la Vue de désigner l'objet destinataire (focus) d'évènement**



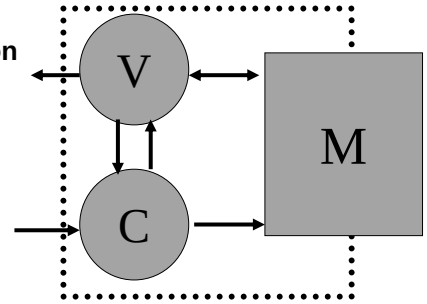
BTD/MAI/MVC 17

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Dynamique MVC

- **3. Le Contrôleur demande à une méthode du Modèle de changer son état.**
- **4. Le Modèle change son état interne**



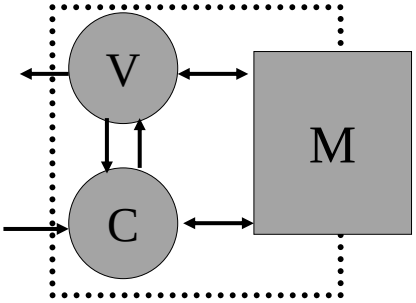
BTD/MAI/MVC 18

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Dynamique MVC

- **5. Le Modèle informe (notifie) toutes les Vues associées (dépendantes) que les données ont changées.**
- **6. La Vue demande au Modèle les valeurs réactualisées des données.**



BTD/MAI/MVC

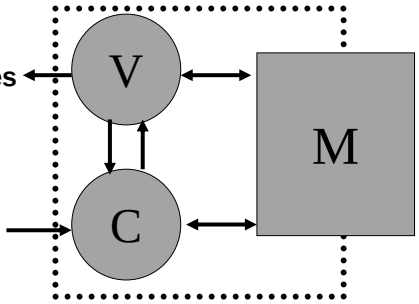
19

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Dynamique MVC

- **7. Le Modèle notifie tous les contrôleurs en relation que les données ont été changées.**
- **8. Le Contrôleur demande au Modèle les valeurs réactualisées des données.**



BTD/MAI/MVC

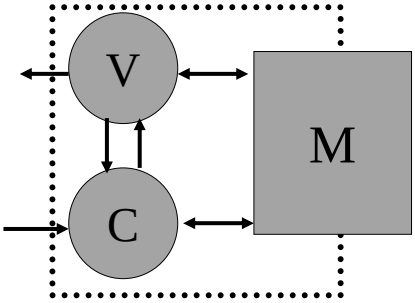
20

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Dynamique MVC

- **9. Contrôleur informe la Vue si éléments sont désactivés.**
- **10. La Vue demande le rafraichissement**



CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Association Vue + Contrôleur

- **Le Contrôleur se doit de "parler" à la Vue car,**
  - **➔ Elle a besoin de la géométrie des sorties pour interpréter les entrées (e.g., désignation)**
  - **➔ Elle a besoin de faire le feedback**
- **Comme résultat, VC tendent à être plus étroitement couplés, pour être considérés comme un seul**
  - **➔ M(VC)**

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Multiples IHM

- **Des multiples paires Vue/Contrôleur peuvent être attachées à un Modèle unique**
- **Pour des raisons d'explication nous montrons seulement une seule paire de Vue/Contrôleur.**

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

MVC : Avantages et inconvénients

- **Avantages :**
  - ➔ **Multiples Vues du même Modèle**
  - ➔ **Synchronisation des Vues**
  - ➔ **V & C "pluggable" et "look and feel"**
- **Inconvénients :**
  - ➔ **Complexité pour des interacteurs simples**
  - ➔ **Potentiellement excessives mises à jours/messages**
  - ➔ **Couplage étroit, en pratique (V-C, VC-M)**
  - ➔ **Fiable portabilité**
  - ➔ **Certains toolkits rendent MVC framework difficile**

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Un Design Pattern commun à Orienté Objet

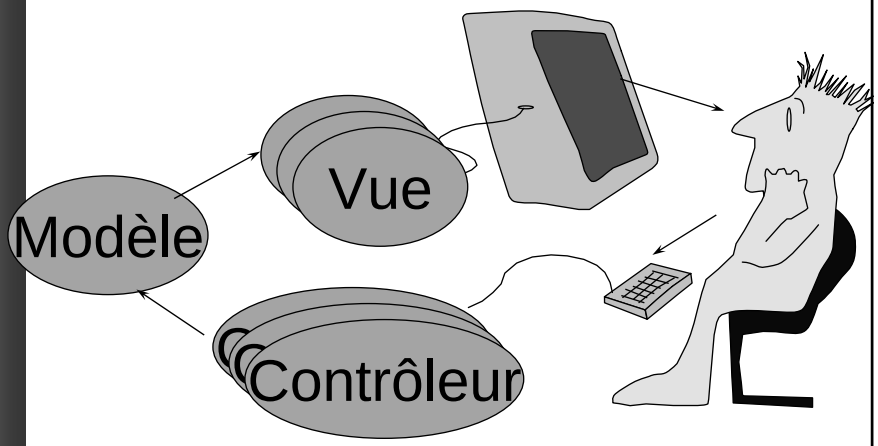
- **Le mécanisme de propagation de changements de MVC apparait près généralement connu**
  - ➔ **Pattern Observer**
  - ➔ **Pattern Publisher-Subscriber**
- **Les objets dépendants sont faiblement couplés**
  - ➔ **Pourquoi le Modèle annonce seulement le changement à la Vue et alors la Vue sollicite des valeurs courantes ?**

BTD/MAI/MVC 25

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Le Pattern Modèle - Vue - Contrôleur



BTD/MAI/MVC 26

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Objectif

- **Structurer une application interactive selon les principes du modèle de Seeheim**
  - ➔ **Séparer les considérations lexicales, syntaxiques et sémantiques**
  - ➔ **Permettre aux mêmes informations d 'être visualisées et manipulées sous différentes formes, dans différentes fenêtres**
  - ➔ **Origines : Langage Smalltalk et environnements associés. Fondement de la plupart des boîtes à outils modernes (Swing)**

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

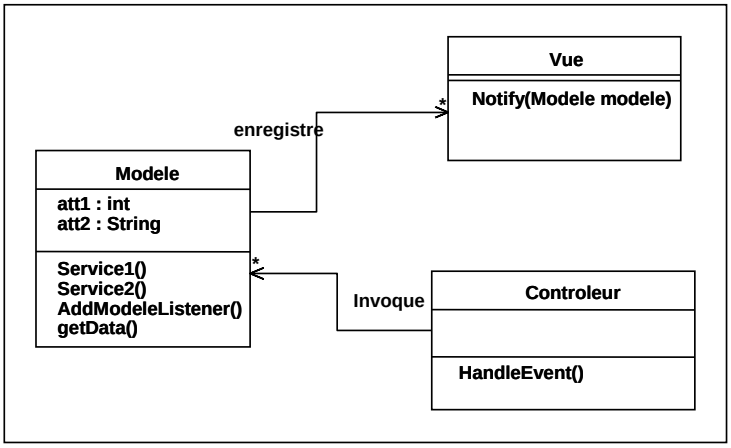
Structure générale

- **Modèle**
  - ➔ **Contient le noyau fonctionnel, indépendant de l'interface**
  - ➔ **Enregistre ses vues**
  - ➔ **Notifie ses vues de ses changements d'état (appel de « notify »)**
- **Vue**
  - ➔ **Présente le modèle à l'utilisateur**
  - ➔ **Implémente « notify »**
  - ➔ **Accède à l'état (public) du modèle (« getdata »)**
- **Contrôleur**
  - ➔ **Permet à l 'utilisateur d'interagir (indirectement) avec le modèle (reçoit les événements utilisateur)**
  - ➔ **Appelle les services sémantiques offerts par le modèle**

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Modèle de Classes

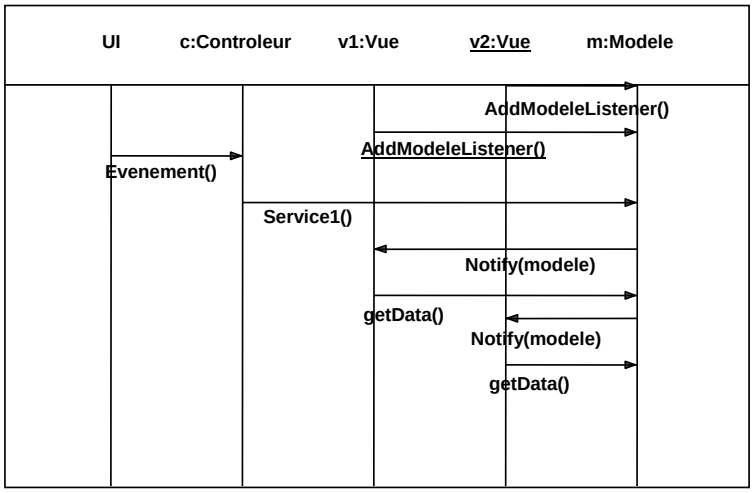


BTD/MAI/MVC 29

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Diagramme de séquence



BTD/MAI/MVC 30

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Conséquences

- **Avantages**
  - ➔ **Facilité à présenter différentes vues du même objet et à les synchroniser**
  - ➔ **Facilité d'ajout d'une nouvelle représentation**
- **Inconvénients**
  - ➔ **Multiplication des « notify »**
  - ➔ **Inefficacité de l'accès au modèle**
  - ➔ **Couplage fort entre vue et contrôleur**
    - ➢ **Impossible en général de réutiliser un contrôleur indépendamment de sa vue**
- **Variantes**
  - ➔ **Document / View, PAC**

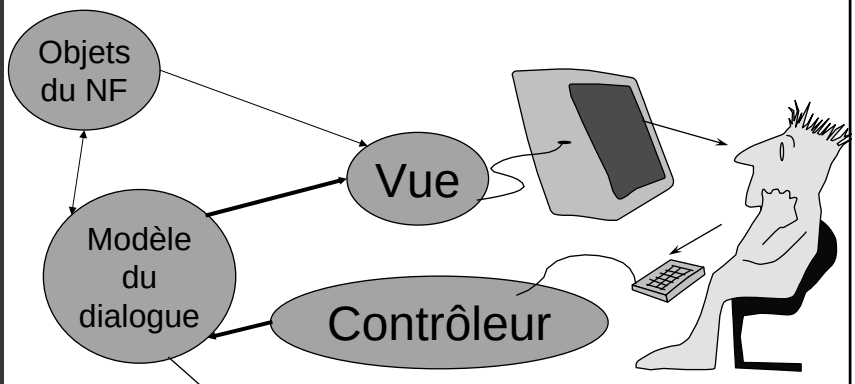
BTD/MAI/MVC

31

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

MVC étendu



Joue les rôles de contrôleur et de vue pour les objets du NF

BTD/MAI/MVC

32

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Applications WEB : MVC2 et STRUTS

- La demande de développement d'applications de plus en plus puissantes et complexes s'est multipliée au fil des années.
- Elles nécessitent un travail considérable de conception, conduisant généralement à l'intervention de multiples informaticiens.
- Elles doivent en permanence être actualisées en fonction des besoins de ses utilisateurs.
- Sa flexibilité et son évolutivité ne peuvent exister que par une structuration de son code, notamment en prenant en charge la séparation des interfaces hommes-machine et des processus métiers tout en prenant en compte le contrôle de saisie des formulaires.
- L'utilisation de framework, et notamment Struts développé au sein du projet Jakarta du groupe Apache, permet de prendre en compte ces besoins.
- Le framework Struts basé sur le modèle MVC II utilise divers outils de développement Web permettant de concevoir des applications où sont séparées les fonctionnalités du modèle, du contrôle et de la vue.

BTD/MAI/MVC 33

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Framework Struts ?

- Un framework (squelette d'application), fournit aux programmeurs un canevas de travail et a pour but de structurer, simplifier et segmenter les développements d'application. Il est composé d'un ensemble de classes et de mécanismes.
- Il existe deux types de framework :
  - ➔ les frameworks métier qui fournissent des services fonctionnels (gestion des clients, d'abonnements, de news, …),
  - ➔ les frameworks techniques qui apportent des concepts, des entités et des mécanismes permettant de s'abstraire d'un certain nombre de problématiques conceptuelles et techniques récurrentes.
- C'est dans le cadre de ce dernier type que s'inscrit le framework Struts. Struts s'appuie sur la plate-forme J2EE.
  - ➔ Plate-forme J2EE
  - ➔ J2EE (Java 2 Entreprise Edition) de Sun Microsystems est une plate-forme de développement d'applications Web. Elle permet de générer en dynamique des pages HTML qui dépendent de requêtes clientes et d'informations stockées dans des bases de données.

BTD/MAI/MVC 34

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

3 éléments pour la réutilisation

Framework	+	Boîte à outils	+	Design Pattern (modèle de conception)
Structure logicielle qui définit des cadres dans lesquels viennent s'insérer les objets et concepts spécifiques à une application		Librairie de composants réutilisables		Elément de conception réutilisable indépendant de tout langage

BTD/MAI/MVC

35

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

MVC et architectures 3-tiers

- **Une architecture 3-tiers peut être représentée comme suit :**

- **L'interface utilisateur peut être un navigateur Web ou une application autonome qui, via le réseau, envoie des requêtes HTTP et met en forme les résultats qui lui sont envoyés**
- **La logique applicative est constituée de scripts traitant les demandes de l'utilisateur.**
- **Les sources de données sont des bases de données, des annuaires LDAP, des services web distants ou tout simplement des fichiers textes ou excel.**

- **Pour faire la liaison entre ces 3-tiers, l'architecture MVC utilise trois éléments :**
- le modèle : **Objet métier qui regroupe les classes Java capables de fournir les données nécessaire à la vue. Il peut accéder directement à une base de données, une base XML ou utiliser des composants Java Beans ou EJB (Enterprise Java Bean).**
- la vue : **Objet chargé de la restitution d'informations vers l'utilisateur.**
- le contrôleur : **Objet chargé de l'acquisition d'informations en provenance de l'utilisateur. Chaque contrôle est alors associé à une vue.**

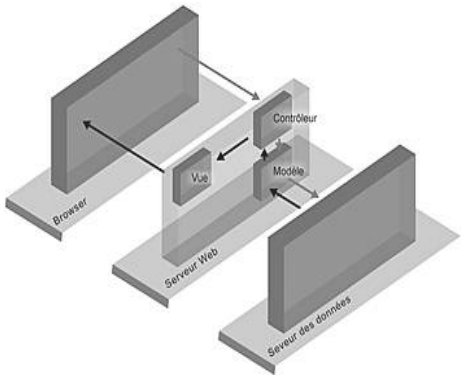
BTD/MAI/MVC

36

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

MVC et architecture 3-tiers



BTD/MAI/MVC 37

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Les Design Pattern

- **Dans le domaine de l'analyse et de la conception orientée-objet, un design pattern est une manière de construire la structure d'une classe. Plus généralement, un motif de conception est un document qui décrit une solution générale à un problème qui revient souvent. Les motifs sont basés sur des expériences passées avec les mêmes structures.**
- **Ainsi, au fil du temps, les développeurs se sont aperçus que certaines conceptions devenaient récurrentes face à certaines situations.**

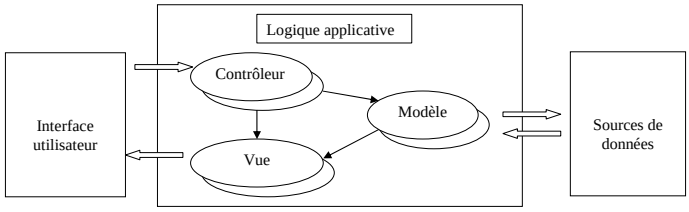
- le Design Pattern Modèle Vue Contrôleur (MVC)
- **Une architecture MVC cherche à séparer trois choses : le Modèle, les Vues et les Contrôleurs. Les contrôleurs permettent de répondre aux actions de l'utilisateur. Chaque contrôle est associé à une vue : cette dernière permet de présenter l'information retournée à l'utilisateur. Bien entendu, l'information renvoyée est dépendante des actions d'entrées de l'utilisateur (capturées par les contrôleurs, nous venons de le dire). Les liens (les traitements) sont réalisés par le modèle (la logique métier).**
- **On peut appliquer une architecture MVC à de nombreux cas de mise en œuvre de systèmes informatiques : infographie, applications, … Bien entendu, on peut aussi appliquer cette architecture à la mise en œuvre d'une application Web.**
- **Il est aussi intéressant de noter que dans une équipe de développement de site Web, on peut identifier, au moins, deux types de développeurs. Les informaticiens : ils ont pour tâche l'implémentation du système. Mais, il y aussi les infographistes : ils sont peut-être familiarisés avec HTML ou les feuilles de styles CSS, mais du code peut éventuellement les rebuter.**
- **Via ce type d'architectures, chacun intervient donc sur un type de fichier qui lui est familier.**

BTD/MAI/MVC 38

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

MVC



- **Ce type d'architecture permet de séparer deux catégories de compétences :**
  - **les développeurs qui sont chargés de l'implémentation du système,**
  - **les infographistes qui s'occupent uniquement des pages HTML et des feuilles de style.**
- **Toutefois le modèle MVC a un inconvénient majeur : il est nécessaire de coder un grand nombre de contrôleurs pour réceptionner les requêtes clientes, compte tenu qu'il faut un contrôleur par vue. Afin de pallier ce problème, certains frameworks préfèrent implémenter une architecture MVC II.**

BTD/MAI/MVC

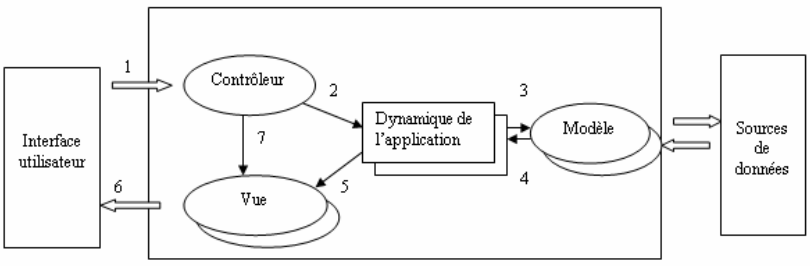
39

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

MVC II

- **Dans cette architecture, il n'existe plus qu'un contrôleur unique qui réceptionne l'ensemble des requêtes clientes. Il garantit l'unicité du point d'entrée de l'application et prend en charge une partie de son contrôle. L'autre partie est du ressort d'une nouvelle entité, appelée « dynamique de l'application ».**



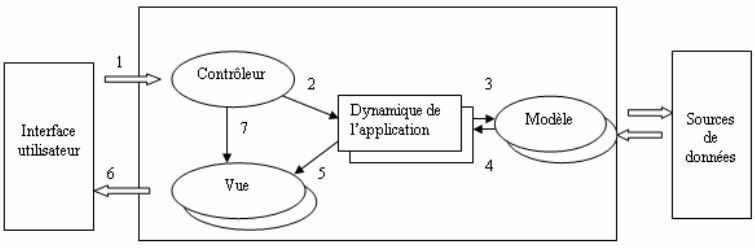
BTD/MAI/MVC

40

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Cycle de vie d'une requête :



1. **Le client envoie une requête à l'application. La requête est prise en charge par le contrôleur d'entrée.**
2. **Ce dernier analyse la requête et la réoriente vers un contrôleur adapté qui se situe dans la dynamique de l'application.**
3. **Responsable de l'exécution des traitements nécessaires à la satisfaction de la requête, le contrôleur sélectionné sollicite les objets métiers (ou modèle) si nécessaire.**
4. **Il y a transfert des données métiers.**
5. **Le contrôleur les encapsule dans des éléments Java appelés JavaBeans, sélectionne la vue qui sera en charge de la construction de la réponse et lui transmet les JavaBeans contenant les données métier.**
6. **La vue construit la réponse en faisant appel aux JavaBeans qui lui ont été transmis et l'envoie au navigateur.**
7. **Lorsque nécessaire, pour le traitement d'erreurs essentiellement, le contrôleur d'entrée traite la requête directement, sélectionne la vue de sortie et lui transmet les informations dont elle a besoin.**

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

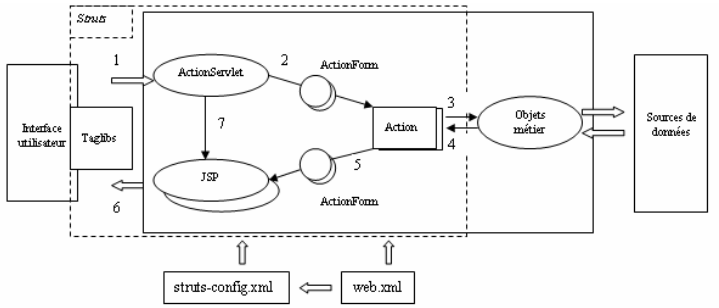
Démarche de développement Struts

- **Struts répond à l'architecture MVC II instancié avec des servlets et des JSP, c'est-à-dire qu'il répartit les responsabilités entre :**
  - ➔le modèle **qui est complètement standard selon l'architecture MVC. Struts ne fournit aucune implémentation particulière, permettant ainsi de l'appliquer à tout projet déjà existant,**
  - ➔la vue **composée de pages JSP,**
  - ➔le contrôleur **décomposé en un contrôleur d'entrée et d'une dynamique d'application. Le premier est une servlet de la classe ActionServlet et la deuxième, appelée aussi sous-contrôleur, est un ensemble d'Actions déclenchées par cette servlet.**
- **Le développement d'une application via Struts se fait par la programmation d'un ensemble d'éléments :**
  - ➔Les vues : **Ce sont les pages de présentation pour l'utilisateur. Elles utilisent des Taglibs qui permettent de limiter le code Java dans ces pages. En effet, les Taglibs sont des actions prédéfinies destinées à être utilisées sous forme de balises XML qui permettent la création aisée de pages JSP sans introduire de code Java.**
  - ➔L'ActionServlet : **C'est le contrôleur principal qui fait le lien entre l'interface utilisateur et l'application.**
  - ➔Les Actions : **Ce sont les actions à effectuer en fonction de chaque requête.**
  - ➔Les ActionForm **: Containers pour les pages JSP, ils stockent les différentes valeurs saisies dans une page JSP.**
  - ➔Le fichier de configuration struts-config.xml : **Il décrit les actions possibles et à quelles classes Actions il faut les associer.**
  - ➔Le fichier de configuration web.xml : **ce dernier contient la déclaration de l'ensemble des fichiers précédents.**

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Struts s'intègre dans le modèle MVC II selon le schéma suivant



1. Le client envoie une requête à l'application. La commande est déterminée par une partie de l'URL transmise à la servlet contrôleur, ActionServlet. La requête est prise en charge par cette dernière.
2. Le contrôleur d'entrée analyse la requête et réoriente celle-ci vers un contrôleur adapté, avec les Taglibs nécessaires, à l'aide des ActionForms.
3. Le contrôleur sélectionné par le contrôleur d'entrée est responsable de l'exécution des traitements nécessaires à la satisfaction de la requête. Les traitements sont encapsulés dans les classes Actions et sollicitent, si nécessaire, les objets métiers.
4. Les objets métiers fournissent des données.
5. L'Action encapsule les données métiers dans les JavaBeans, sélectionne la page JSP qui sera en charge de la construction de la réponse et lui transmet les JavaBeans contenant les données métiers par le biais des ActionForms.
6. La vue construit la réponse en faisant appel aux JavaBeans qui lui ont été transmis et l'envoie au navigateur.
7. Lorsque nécessaire, pour le traitement d'erreurs essentiellement, la servlet d'entrée traite la requête directement, sélectionne la vue de sortie et lui transmet les informations dont elle a besoin.

BTD/MAI/MVC

43

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Le MVC 2

- **MVC 2 : dans une telle architecture, il n'existe plus qu'un seul et unique contrôleur réceptionnant toutes les requêtes clientes.**
-
- **Mais outre les possibilités du système utilisé, il faut souvent aussi considérer sa "durée de vie" et donc sa maintenance.**
- **Le contrôleur unique devient le point d'entrée exclusif de l'application. Il devient alors très aisé de centraliser la gestion des accès, des droits, des statistiques ou de toute autre fonctionnalité transverse.**

BTD/MAI/MVC

44

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Framework Struts

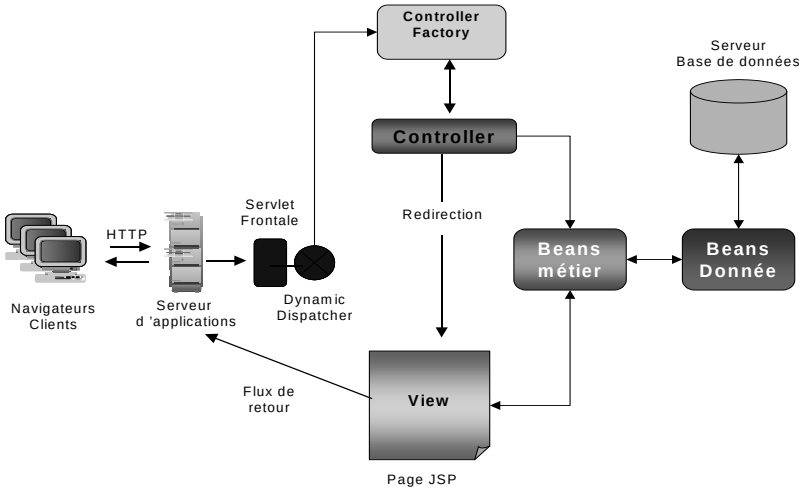
- **Il existe un grand nombre de framework basés sur une architecture MVC ou MVC II, comme par exemple Baraccuda, WebWork, Coccoon, Struts.**
- **L'appartenance du framework Struts au projet Jakarta d'Apache, auteur du serveur Web du même nom, garantit sa pérennité.**
- **De plus, étant un projet Open Source, ses sources sont publiques, ce qui le rend adaptable et extensible.**
- **Ce framework est donc à privilégier.**

BTD/MAI/MVC 45

CENTRALE LYON

Bertrand DAVID : Méthodologies avancées d'informatisation

Architecture Web



BTD/MAI/MVC 46

CENTRALE
L Y O N

Bertrand DAVID : Méthodologies avancées d'informatisation

Quelle différence entre les architectures Mvc1 et Mvc2

- MVC est un design pattern.
- Il correspond à deux modèles : MVC Modèle 1, MVC Modèle 2.
- Le framework Struts implémente Design Pattern MVC. Struts peut implémenter Modèle 1 et Modèle 2. Modèle 2 décrit plus clairement l'application de MVC dans le contexte Web.
- Les caractéristiques importantes de l'architecture MVC1 :
  - ➔ (1) les fichiers HTML ou JSP sont utilisés pour le code de présentation. L'utilisation de fichiers JSP est nécessaire pour l'accès aux données.
  - ➔ (2) Architecture MVC1 est orientée vers la conception de page. Toute la logique fonctionnelle et de traitement se trouve dans les JSP ou peut être appelée depuis les pages JSP.
  - ➔ (3) Accès aux données est assuré soit par des tags utilisateur ou par des appels via les Java bean.
- Les caractéristiques importantes de l'architecture MVC2 : Cette architecture quitte l'approche à base de pages au profit de la séparation entre la Présentation, le Contrôle et l'état de l'Application.
- Dans une architecture MVC2 :
  - ➔ Un Contrôleur qui reçoit toutes les demandes pour l'application et est responsable d'activité l'action appropriée à la sollicitation.
  - ➔ Un Modèle qui est représenté par les JavaBeans, les objets métier et la base de données.
  - ➔ Une Vue ou la page JSP prend les informations venant du Contrôleur et le Modèle et les présente à l'utilisateur.

3TD/MAI/MVC

47