

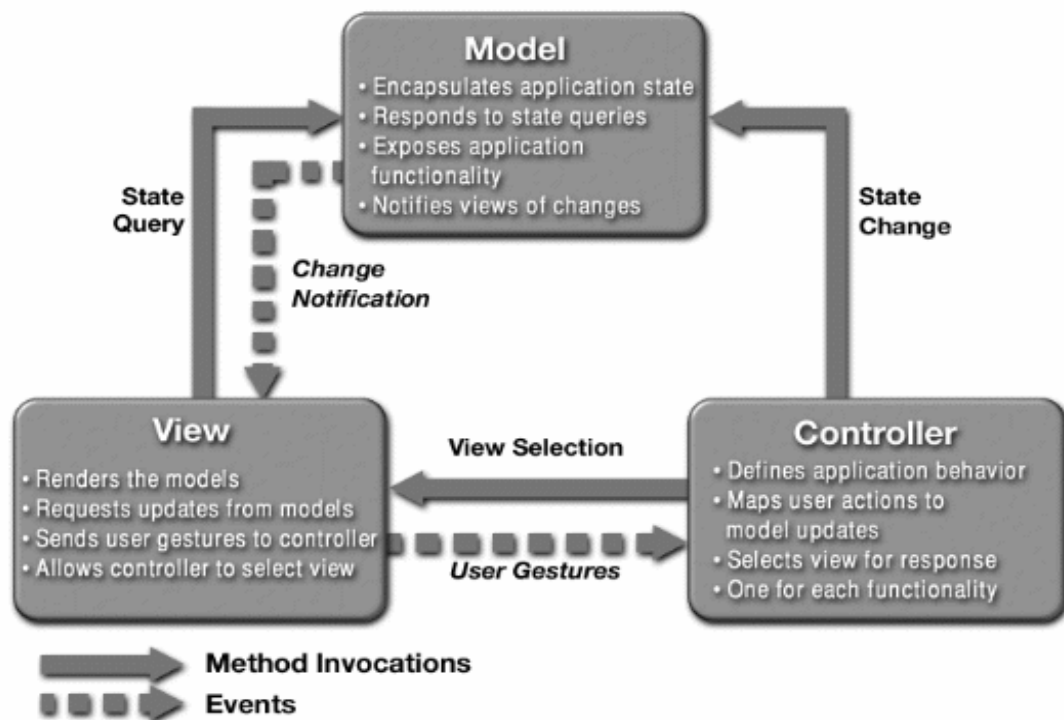
Aplicații Web cu Servleți și pagini JSP – MVC

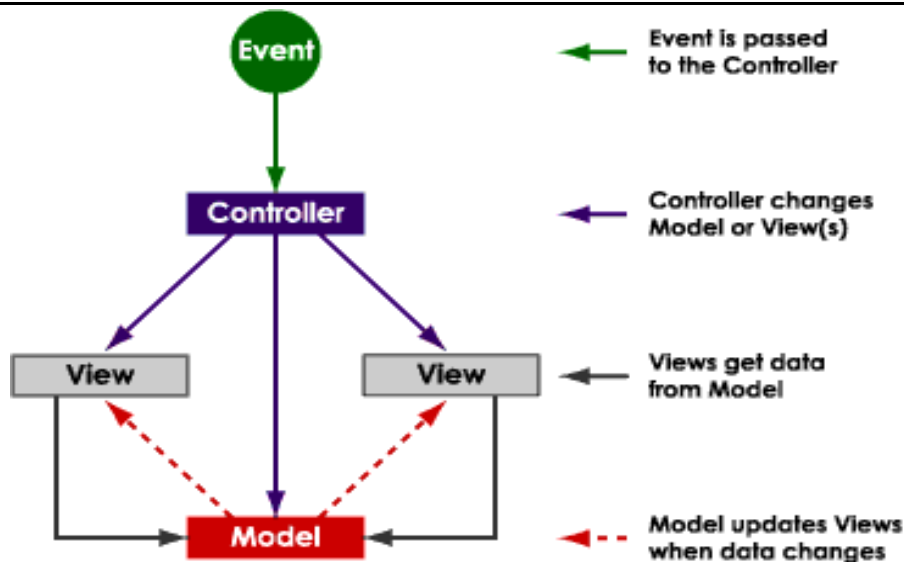
1. MVC

Arhitectura MVC (Model – View – Controller) a fost inițiată în Smalltalk, pentru a asocia introducerea datelor, prelucrarea lor și prezentarea rezultatelor cu modelul de interfață grafică utilizator.

Componentele MVC au următorul rol:

- Model: modelează datele și regulile de funcționare (acces și actualizare date).
- View: prezintă conținutul modelului în modul specificat, accesând datele din model și trebuie să păstreze consistența prezentării datelor atunci când modelul se schimbă:
 - Push, când view se înregistrează (abonează) la model pentru a fi notificat la producerea modificărilor.
 - Pull, când view apelează modelul pentru a obține datele cele mai recente.
- Controller: translatează interacțiunea cu view în acțiuni care să fie efectuate de model.
 - Interacțiunea înseamnă:
 - Pentru o aplicație desktop apăsări de butoane sau selecții de meniuri.
 - Pentru o aplicație Web cereri HTTP GET sau POST.
 - Se activează procese specifice și se schimbă starea modelului.
 - Răspunsul este selectarea unui view potrivit.





MVC are rolul de a separa nivelul de business al aplicației de interfața utilizator (nivelul de prezentare). Scopul este acela de a separa starea curentă și comportamentul aplicației de preluarea datelor și afișarea acestora utilizatorului.

MVC separă interfața cu utilizatorul în controller-e și view-uri, astfel încât modificări aduse interfeței cu utilizatorul nu afectează modul în care sunt gestionate datele, iar datele pot fi reorganizate fără a aduce modificări interfeței cu utilizatorul. MVC rezolvă această problemă prin decuplarea accesului la date și logica de business de afișarea datelor și interacțiunea cu utilizatorul. Controller-ul MVC este ilustrat în Figura anterioară:

- “Model-ul” este reprezentat de diferite componente cum ar fi entitățile de business, componentele de business, componentele pentru accesul la date și agenți de servicii.
- “View-urile” sunt componente ale interfeței cu utilizatorul cum ar fi paginile JSP / ASP.NET.
- “Controller-ul” implementează metodele pe care “View-ul” le poate apela atunci când face o cerere de date sau pornește o acțiune.

“Page Controller” reprezintă o variantă a MVC și este potrivit în aplicații client unde “View” și “Controller” sunt separate ușor. “Page Controller” poate fi folosit pentru a permite intrări dintr-o pagină, pentru a chema acțiunile necesare ale modelului și în final pentru a determina ce view trebuie folosit pentru pagina Web rezultată. Obiectul “Page Controller” primește un request (de la un end-user printr-un browser) și după ce s-au extras datele relevante (request HTTP) acesta apelează actualizările modelului și trimite cererea către “View”. La rândul său, “View-ul” depinde de model pentru a obține date ce vor fi afișate.

“Application Controller” se folosește pentru gestiunea interacțiunilor utilizatorilor multipli, prin definirea și controlul asupra fluxului dintre acestea, oferind management de stare și relaționând “View-urile” cu “Controller-ele”. În felul acesta se impune controlul asupra fluxului dintre diferite view-uri și se gestionează ușor stările acestor view-uri.

2. Exemplu folosind NetBeans

Rulați exemplul cu conturi bancare din fișierul DAI_Lab_03_MVC.pdf (secțiunea 15.5) folosind NetBeans / GlassFish. Folosiți ca fișiere de plecare cele din arhiva DAI_Lab_03_MVC.zip.

Identificați rolul servleților. Identificați rolul paginilor JSP. Stabiliți dacă folosirea lor împreună aduce vreun câștig.

3. Exemplu în linia de comandă

Reluați exemplul anterior, folosind linia de comandă și serverul Tomcat. Atenție la organizarea directoarelor.