

Tecnologie di Sviluppo per il Web

Applicazioni Web J2EE Modello 2

versione 3.2

Questo lavoro è concesso in uso secondo i termini di una licenza Creative Commons
(vedi ultima pagina)

G. Mecca – mecca@unibas.it – Università della Basilicata



Sommario

- Preliminari
- Modello 2
 - ⇒ Modello 2 con la Piattaforma J2EE
- Implementazione del Controllo
 - ⇒ Controllore Frontale (“Front Controller”)
 - ⇒ Controllo Dichiarativo



Preliminari

- Architettura Modello 1
 - ⇒ interfaccia e controllo assieme sono realizzate da pagine jsp
 - ⇒ modello e persistenza assieme sono realizzate da JavaBeans
- Difetti dell'architettura Modello 1
 - ⇒ i difetti ordinari delle architetture che mischiano interfaccia e controllo
 - ⇒ ulteriormente enfatizzati dalla piattaforma



Preliminari

- Interfaccia
 - ⇒ fatta di pagine HTML
 - ⇒ cambia abbastanza frequentemente
 - ⇒ è necessario fornire a client diversi interfacce diverse (es: browser da tavolo, dispositivi mobili, accessibilità)
 - ⇒ spesso ha impatto sul successo del sito e deve essere progettata e sviluppata da grafici professionisti



Preliminari

○ Controllo

- ⇒ codice Java mischiato al codice HTML
- ⇒ poco modulare a causa delle rigidità della piattaforma tecnologica (minore libertà nell'organizzazione in classi delle pagine jsp)
- ⇒ difficile fattorizzare blocchi di codice comune
- ⇒ deve essere sviluppato da sviluppatori professionisti



Preliminari

○ Problema

- ⇒ l'architettura basata sul modello 1 è adatta allo sviluppo rapido di applicazioni di piccole dimensioni
- ⇒ al crescere della dimensione dell'applicazione tende molto rapidamente a rendere il codice difficile da mantenere
- ⇒ per siti di grandi dimensioni è opportuno adottare un'architettura basata sul modello 2



Modello 2

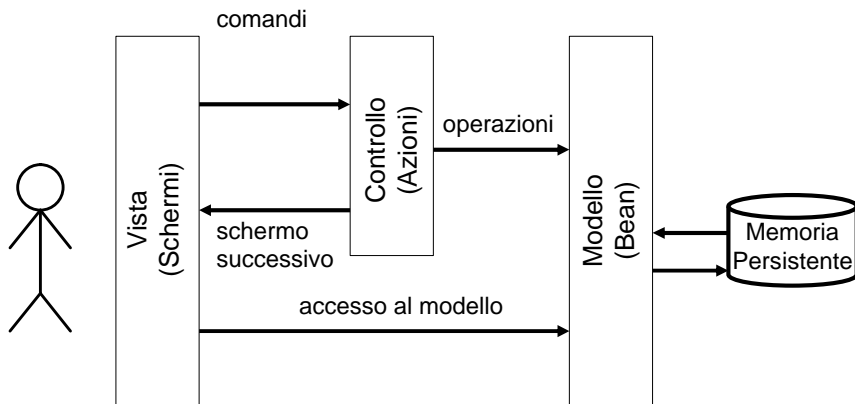
- Architettura di tipo Modello 2
 - ⇒ variante dell'architettura MVC
 - ⇒ adattata alla tecnologia delle applicazioni Web
- Architettura MVC
 - ⇒ architettura pensata per le applicazioni desktop
 - ⇒ basata sulla separazione tra gli strati di interfaccia, modello e controllo



Modello 2

- Idea alla base di MVC
 - ⇒ l'utente interagisce con gli *schermi*; i gesti dell'utente scatenano *eventi* che inviano *comandi* al controllo
 - ⇒ il controllo analizza i dati dell'utente ed effettua le *azioni*; le azioni cambiano lo stato del *modello*
 - ⇒ sulla base dell'esecuzione delle azioni, le azioni chiedono alla vista l'esecuzione dello *schermo successivo*

Architettura MVC Generale



Modello 2

- In concreto, in un'architettura MVC
 - ⇒ ogni strato è costituito da un componente di "coordinamento" e da una serie di altri componenti
 - ⇒ la vista è rappresentata da un componente Vista, più una serie di schermi
 - ⇒ il controllo è rappresentato da un componente Controllo capace di tradurre comandi nell'esecuzione di azioni, e da una serie di azioni
 - ⇒ il modello è rappresentato da un componente Modello che è una collezione di riferimenti ai JavaBean che contengono la logica applicativa



Modello 2

- Vantaggi dell'architettura
 - ⇒ interfaccia e controllo sono separati
 - ⇒ il codice è meglio organizzato
 - ⇒ si riduce l'accoppiamento
 - ⇒ è possibile cambiare l'interfaccia senza intaccare gli altri strati dell'applicazione
- Svantaggi dell'architettura
 - ⇒ maggiore complessità del codice



Modello 2

- In un'applicazione Web
 - ⇒ per la realizzazione di questa architettura sono necessarie delle varianti
- Le differenze principali
 - ⇒ gli schermi sono visualizzati dal browser; cambia la comunicazione tra vista e controllo
 - ⇒ l'applicazione è multiutente; non esiste un unico modello, ma tante "sessioni"



Modello 2

○ In concreto

- ⇒ gli schermi sono *le pagine HTML* generate dalle pagine server e visualizzate dal browser
- ⇒ i comandi inviati dalla vista al controllo sono *richieste HTTP* (unica tipologia di evento)
- ⇒ il controllo riceve i dati dell'utente attraverso la *query string* e deve effettuare le *azioni relative*
- ⇒ le azioni modificano lo stato della *sessione*



Modello 2

○ Un'ulteriore differenza

- ⇒ in un'applicazione Web gli schermi sono rigenerati *ex-novo* dal browser al termine di ogni transazione http
- ⇒ produrre lo schermo successivo richiede di *inoltrare la richiesta* alla pagina server opportuna dopo l'esecuzione delle azioni
- ⇒ di conseguenza il controllo deve essere un componente lato server capace di effettuare operazioni di *inoltro della richiesta HTTP*



Modello 2

○ Un ulteriore differenza

- ⇒ gli schermi sono rigenerati ex-novo dal browser al termine di ogni transazione http
- ⇒ produrre lo schermo successivo richiede eseguire una pagina server opportuna

○ Di conseguenza

- ⇒ è necessario al termine di ciascuna azione effettuare una operazione di *inoltra* alla pagina server responsabile dello schermo successivo



Modello 2

○ Una possibilità

- ⇒ le azioni restituiscono un *risultato*
- ⇒ sulla base del risultato, il controllo decide lo pagina server successiva che produrrà lo schermo successivo, ovvero la *risposta HTTP* e gli inoltra la richiesta
- ⇒ la pagina server accede alla sessione per la visualizzazione dei dati relativi allo stato dell'applicazione

Modello 2

○ In concreto

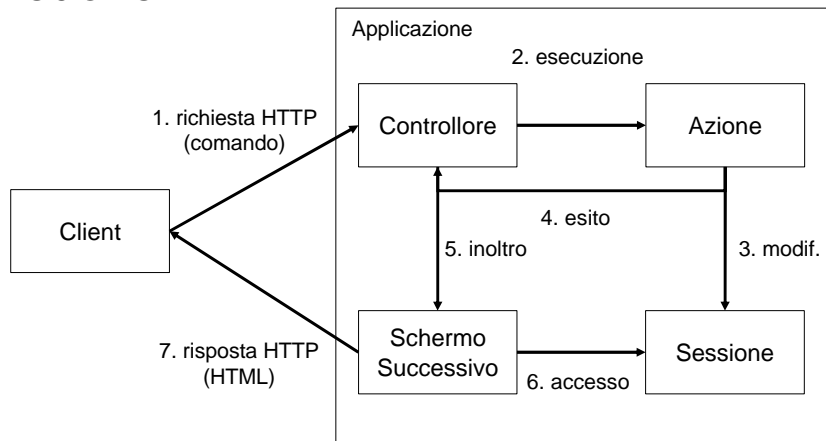
⇒ resta da decidere cosa sono i comandi e i risultati delle azioni

○ La soluzione tipica

⇒ i comandi sono stringhe e corrispondono agli URI utilizzati nelle richieste HTTP

⇒ gli esiti sono stringhe restituite dal metodo di esecuzione delle azioni

Modello 2





Modello 2

○ Nota

- ⇒ in questo approccio non necessariamente gli URI utilizzati per le richieste corrispondono a risorse fisicamente accessibili sul server
- ⇒ tipicamente sono stringhe che rappresentano i comandi (es: esci.cmd)
- ⇒ il controllo si incarica di interpretare i comandi ed eseguire le azioni, inoltrando al termine la richiesta alla pagina server opportuna



Modello 2 con la Piattaforma J2EE

○ L'architettura Modello 2

- ⇒ è un'architettura di applicabilità generale
- ⇒ implementabile con diverse tecnologie

○ In particolare

- ⇒ è molto utilizzata per lo sviluppo di applicazioni Web J2EE
- ⇒ in questo caso i componenti assumono una connotazione più specifica



Modello 2 con la Piattaforma J2EE

○ Applicazioni Web MVC con J2EE

- ⇒ la vista è realizzata attraverso pagine JSP utilizzate esclusivamente per produrre gli schermi; il coordinamento tra gli schermi è effettuato dal RequestDispatcher (che effettua gli inoltri)
- ⇒ il modello è rappresentato dai JavaBeans e da HttpSession (ogni utente ha la sua sessione di lavoro)



Modello 2 con la Piattaforma J2EE

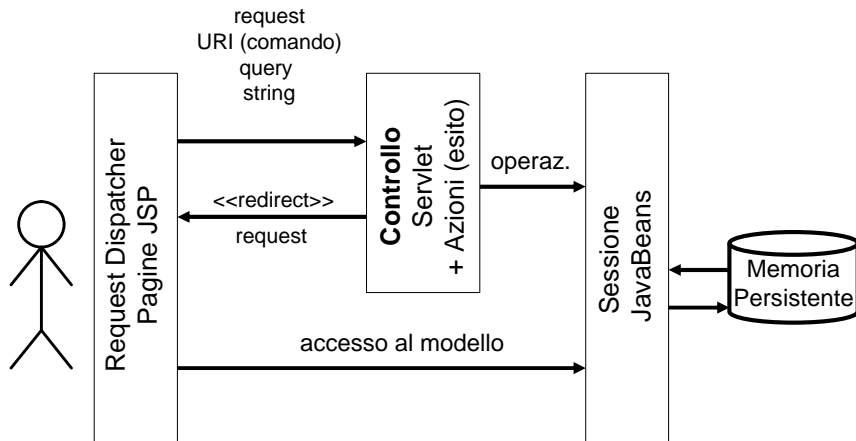
○ Il controllo

- ⇒ deve contenere le azioni
- ⇒ deve essere fatto di componenti lato server orientati alla comunicazione HTTP
- ⇒ devono ricevere i comandi attraverso richieste HTTP
- ⇒ ed effettuare gli inoltri al termine dell'esecuzione delle azioni

○ Scelta naturale

- ⇒ implementare il controllo utilizzando servlet

Modello 2 con la Piattaforma J2EE



Realizzazione del Controllo

- E' il problema centrale
 - ⇒ nel controllo si concentra il nucleo dell'applicazione
 - ⇒ è il punto principale in cui intervenire in caso di cambiamento delle specifiche
- La soluzione tipica
 - ⇒ utilizzo di un controllore frontale
 - ⇒ con controllo dichiarativo

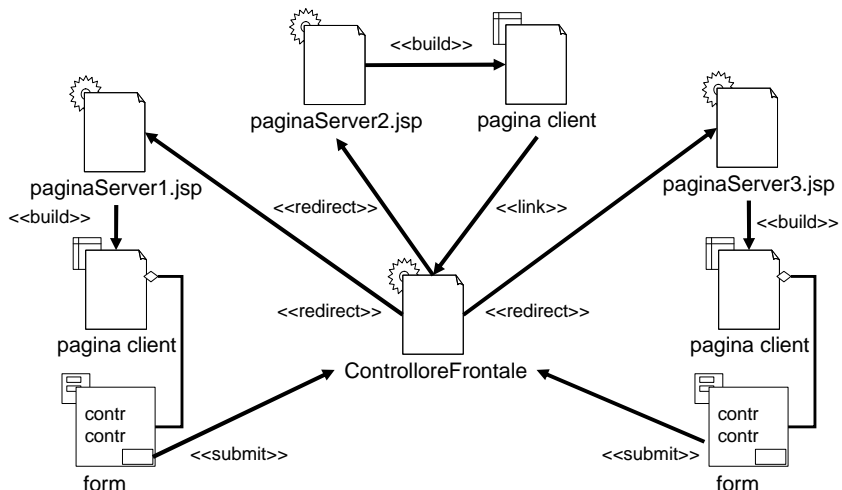


Front Controller

- Controllore Frontale (“Front Controller”)
 - ⇒ un pattern per applicazioni Web
- Idea
 - ⇒ un unico servlet per il controllo
 - ⇒ il servlet riceve i comandi dell’utente sotto forma di stringhe associate agli URI
 - ⇒ esegue le azioni corrispondenti
 - ⇒ effettua l’inoltro allo schermo successivo



Front Controller





Controllo Dichiarativo

○ Problema

- ⇒ come associare azione a comando e schermo ad esito ?
- ⇒ idea: utilizzare una “mappa delle azioni”

○ Mappa delle azioni

- ⇒ mappa associativa che associa a ciascun comando il riferimento ad un oggetto che rappresenta l'azione corrispondente, e a ciascun esito lo schermo successivo



Controllo Dichiarativo

○ Inizializzazione della mappa

- ⇒ la mappa viene caricata da un file di configurazione
- ⇒ file xml in cui è descritto dichiarativamente il flusso di controllo
- ⇒ il controllore carica la mappa all'avvio
- ⇒ e gestisce il flusso di controllo sulla base delle indicazioni nella mappa



Controllo Dichiarativo

- Il file di configurazione
 - ⇒ un descrittore per ogni comando
 - ⇒ specifica l'azione da eseguire (nome della classe corrispondente all'azione)
 - ⇒ per ciascuno dei possibili esiti delle azioni eseguite, specifica lo schermo successivo



Controllo Dichiarativo

- Esempio: descrittore di un'azione

```
<descrittoreAzione comando="inizia.cmd"  
    azione="it.unibas.aci.azioni.AzioneInizia">  
  <schermoSuccessivo esito="successo">  
    /schermoIniziale.jsp  
  </schermoSuccessivo>  
  <schermoSuccessivo esito="errore">  
    /schermoErrore.jsp  
  </schermoSuccessivo>  
</descrittoreAzione>
```



Controllo Dichiarativo

○ Teoricamente

- ⇒ per ogni applicazione Web lo sviluppatore potrebbe re-implementare il suo controllo dichiarativo
- ⇒ operazione complessa (uso intenso della riflessione, varie api complesse, es: DOM)
- ⇒ i vantaggi provenienti dall'architettura sarebbero vanificati dal costo di sviluppo
- ⇒ è opportuno utilizzare un framework



Riassumendo

- Preliminari
- Modello 2
 - ⇒ Modello 2 con la Piattaforma J2EE
- Implementazione del Controllo
 - ⇒ Controllore Frontale ("Front Controller")
 - ⇒ Controllo Dichiarativo



Termini della Licenza

- This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/1.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.
- Questo lavoro viene concesso in uso secondo i termini della licenza "Attribution-ShareAlike" di Creative Commons. Per ottenere una copia della licenza, è possibile visitare <http://creativecommons.org/licenses/by-sa/1.0/> oppure inviare una lettera all'indirizzo Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.