GenLApp
Generazione Lista di Applicazioni

Progettazione della Linea di Prodotti

Progettazione della Linea di Prodotti…

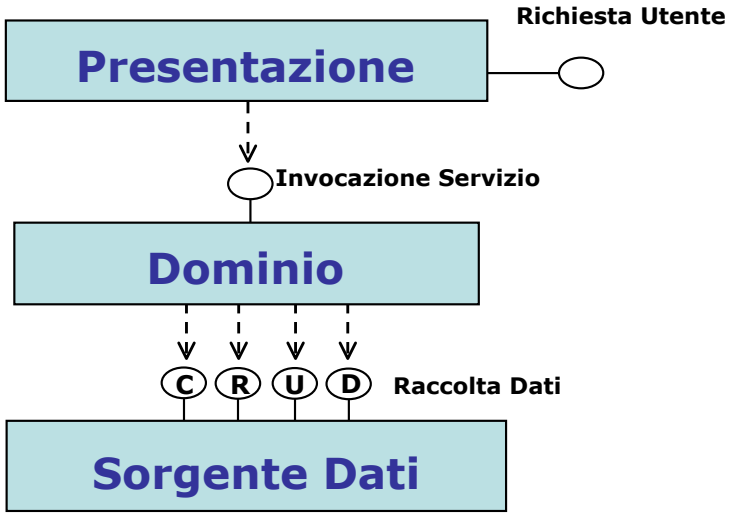
- Classi Essenziali
  - Responsabilità sui 3 Livelli Architetturali
  - Descrizione delle Responsabilità
- Modellazione Dati
  - Diagramma delle Dipendenze dei Dati
  - Modello del Database
  - Dettaglio dei Dati
  - Grammatica file XML (eventuale)

…Progettazione della Linea di Prodotti

- Architettura
  - Livelli Architetturali
  - Diagramma delle Componenti
- Progetto di Dettaglio
  - Diagramma delle Classi
  - Specifiche delle Classi
  - Diagrammi di Sequenza
- Appendice
  - Pattern Utilizzati

Design Patterns

Architettura a 3 Livelli…



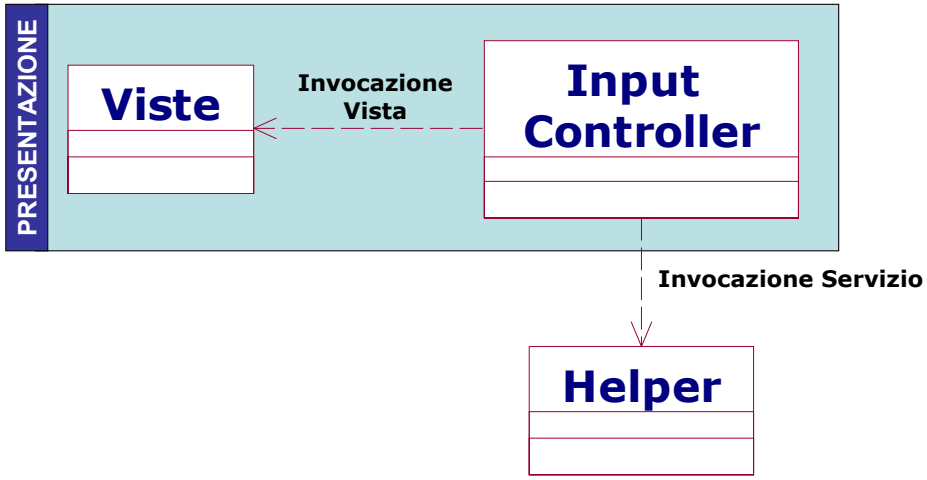
…Architettura a 3 Livelli…

- Presentazione
  - Le componenti di questo livello contengono
    - la logica di presentazione del sistema
    - la logica di navigazione delle viste del sistema
- Dominio
  - Le componenti di questo livello contengono la logica di business per l'esecuzione delle funzionalità di dominio
  - Le componenti devono poter accedere ai servizi offerti dal livello di Sorgente di Dati

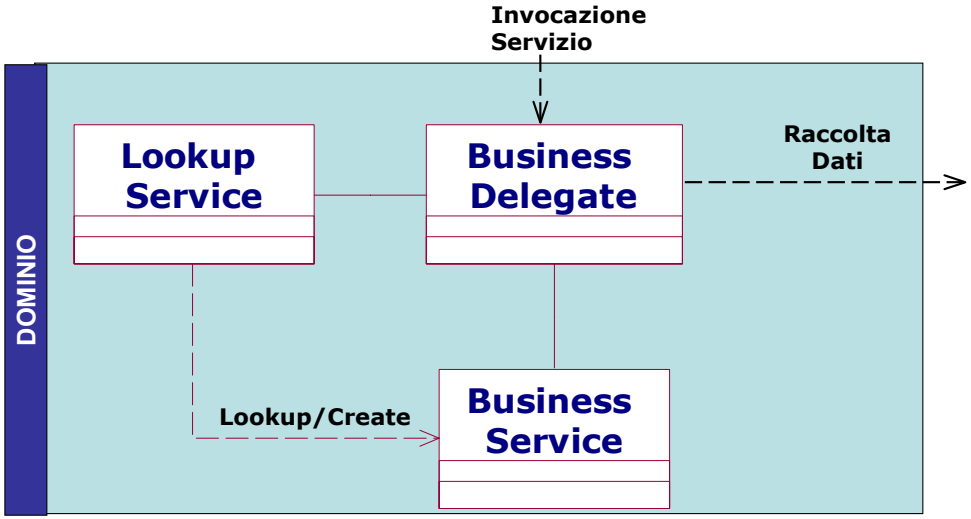
…Architettura a 3 Livelli

- Sorgente Dati
  - Le componenti di questo livello hanno la responsabilità di accedere al database appropriato ed eseguire le interrogazioni necessarie a soddisfare la richiesta del client
  - In questo livello del database si conosce
    - la struttura logica dei database
    - la struttura fisica dei database
  - Solo in questo livello è possibile eseguire le operazioni di *create*, *update*, *read* e *delete* sui database

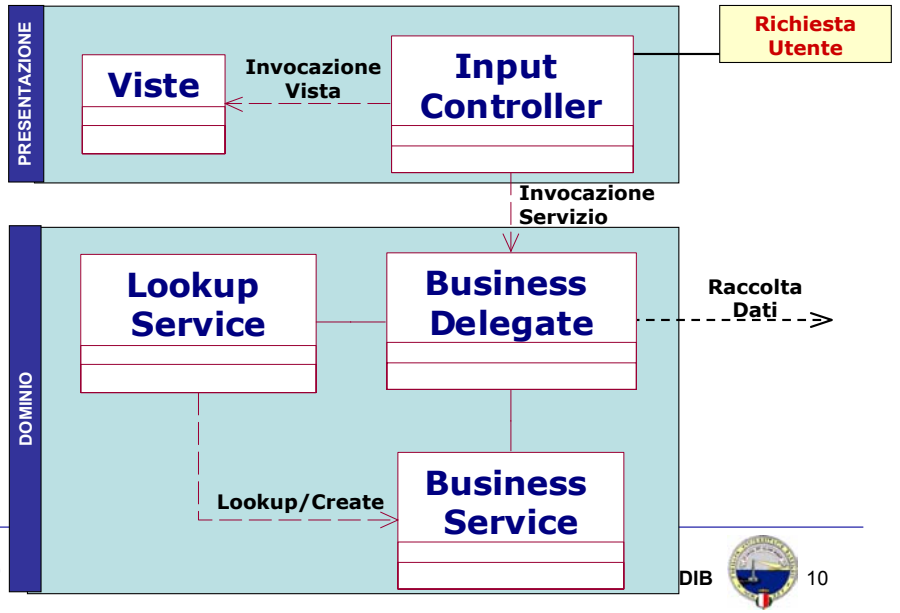
Front Controller pattern (Presentazione)



Business Delegate pattern (Dominio)



Architettura: Presentazione – Dominio



Presentazione: Responsabilità

- ⇨ Input Controller
  - È il punto di contatto iniziale per gestire tutte le richieste al sistema
  - Conosce per ogni richiesta dell'utente
    - La *vista* successiva da mostrare, oppure
    - Il *servizio* adeguato da richiedere al dominio
  - Incorpora una funzionalità per il Dispatcher
    - gestione e navigazione delle *viste*
- ⇨ Vista
  - Visualizza su un *client* le informazioni recuperate dai livelli inferiori

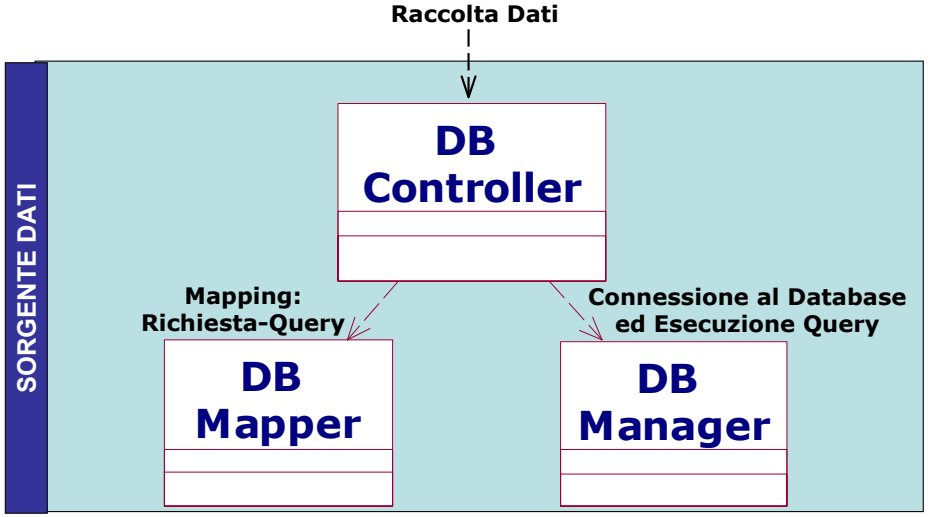
Dominio: Responsabilità…

- ⇨ BusinessDelegate
  - Fornisce controllo e protezione ai servizi di dominio
  - Conosce per ogni servizio richiesto dell'utente
    - La *sequenza di operazioni necessarie*
  - Il livello di presentazione accede ai servizi di dominio SOLO mediante il *BusinessDelegate*

…Dominio: Responsabilità

- LookUpService
  - Incorpora una funzionalità per individuare i *BusinessServices* necessari al servizio richiesto
- BusinessService
  - È un componente del livello di dominio che fornisce un servizio *tipicamente di dominio*

Architettura: Sorgente Dati



Sorgente Dati: Responsabilità…

- DBController
  - Gestisce l'esecuzione di ogni richiesta
  - Ha il compito di
    - Invocare il DBMapper
    - Invocare il DBManager
    - Restituire il risultato dell'interrogazione del database

…Sorgente Dati: Responsabilità

- DBMapper
  - Conosce la struttura logica dei database, in quali tavole sono presenti i dati obiettivo dell'interrogazione e le relazioni tra le tavole per la composizione dei dati
  - Ha il compito di
    - trasformare la richiesta in interrogazioni appropriate (query)

DBManager
- Conosce la struttura fisica dei database.
- Ha il compito di
  - effettuare la connessione al database opportuno
  - eseguire le query

Appendice
- Patterns Utilizzati -

Front Controller
- Livello di Presentazione -

Front Controller pattern…

- ⇨ Contesto
  - Il livello di presentazione necessita di un meccanismo di gestione centralizzata delle richieste dell'utente
- ⇨ Problemi
  - Codice Duplicato: ogni vista deve *auto-fornirsi* di un servizio
  - Navigazione delle Viste: è a carico delle viste stesse

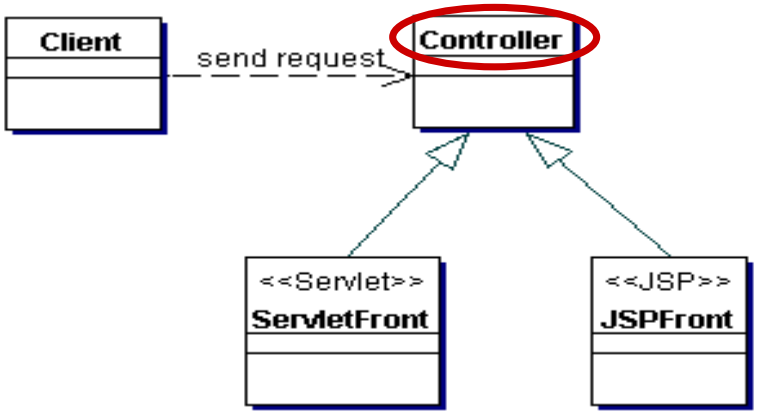
una singola modifica nel sistema può impattare su più viste

…Front Controller pattern…

- ⇨ Soluzione: usare un controller come punto iniziale per gestire le richieste dell'utente
  - Invocare un servizio di sicurezza
    - Autenticazione, autorizzazione
  - Delegare un processo di business
  - Gestire la scelta di una vista
  - Gestire gli errori
  - Gestire la scelta dei contenuti della vista

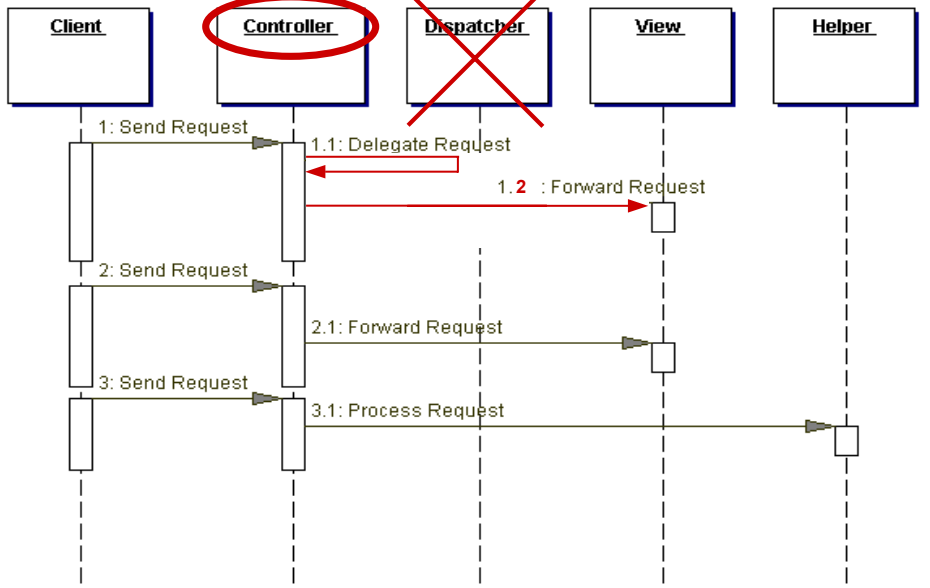
…Front Controller pattern

⇨Struttura



Sequence Diagram

Noi adottiamo una semplificazione



Partecipanti…

⇨Controller

- È il punto di contatto iniziale per gestire tutte le richieste al sistema

⇨Dispatcher

- Responsabilità
  - gestione e navigazione delle *viste*
- Implementazione
  - incorporato in un *controller* (noi faremo così)
  - componente a parte che coopera con il *controller*
  - Java: il dispatcher usa l'oggetto RequestDispatcher

…Partecipanti

⇨Helper

- Aiuta una *vista* o un *controller* a completare la sua elaborazione
- Responsabilità
  - Reperire (e conservare) i dati necessari alle viste
  - Adattare i dati necessari alle viste (opzionale)
- Implementazione
  - JavaBeans, XSLT, Delegates,…

⇨Vista

- Visualizza su un *client* le informazioni raccolte da un *helper*

Vantaggi

- ⇨ Maggiore Maneggevolezza e Sicurezza
  - Un controllo centralizzato delle richieste restringe il numero di punti di accesso del sistema (es.WebApplication) ➔ controlli di sicurezza meno onerosi
- ⇨ Maggiore Riusabilità
  - Un controllo centralizzato favorisce una progettazione dell'applicazione *più pulita*
    - Riduce il business-code nelle viste
    - Favorisce il riuso del codice

Business Delegate
- Livello di Dominio -

Business Delegate pattern…

- ⇨ Contesto
  - Un sistema distribuito e multi-livello richiede l'invocazione di metodi per inviare e ricevere dati tra i vari livelli
- ⇨ Problemi
  - Interazione diretta tra i componenti dei livelli di Presentazione ⇔ Dominio
    - I componenti del livello di presentazione sono vulnerabili ai cambiamenti dei componenti del livello di dominio

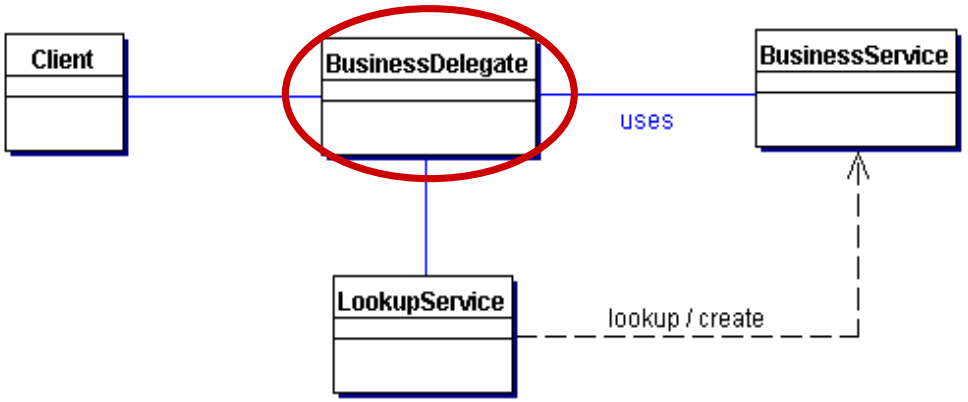
Se l'implementazione di un servizio nel dominio cambia, il codice esposto nella presentazione deve cambiare

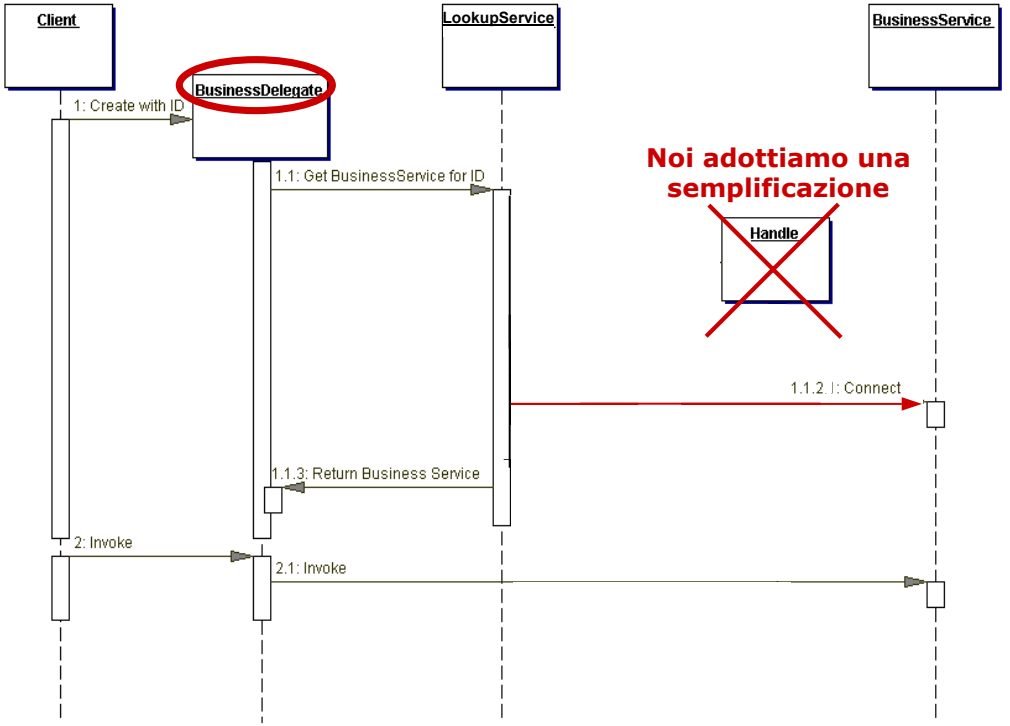
…Business Delegate pattern…

- ⇨ Soluzione: usare un Business Delegate al fine di
  - Ridurre l'accoppiamento tra i client del livello di presentazione e i servizi del livello di dominio
  - Nascondere ai client i dettagli implementativi relativi ai servizi di dominio

…Business Delegate pattern

⇨Struttura





Partecipanti…

⇨ BusinessDelegate

- Fornisce controllo e protezione ai servizi di dominio
- Usa una stringa ID per riconnettersi al servizio di dominio opportuno
- Il *BusinessDelegate* fa da *scudo*, nasconde al client i dettagli implementativi dei servizi di dominio (naming e lookup)
- Il livello di presentazione accede indirettamente ai servizi di dominio
  - mediazione del *BusinessDelegate*

…Partecipanti

⇨LookupService

- Usato dal *BusinessDelegate* per individuare il servizio di dominio (*BusinessService*)

⇨BusinessService

- È un componente del livello di dominio che fornisce il servizio richiesto dal client
  - Es. Enterprise JavaBeans

Vantaggi

- Maggiore Maneggevolezza
  - Un *BD* riduce l'accoppiamento tra il livello di presentazione e il livello di dominio
    - Nasconde i dettagli implementativi del dominio
  - È più facile gestire i cambiamenti
    - Modifiche al solo *BusinessDelegate*
- Migliori Prestazioni
  - Un *BD* può fornire al livello di presentazione un servizio di *caching* per le richieste più comuni