Java Enterprise Edition (JEE)
“Core Design Patterns”
JEE Core Design Patterns

Presentation Tier
- Intercepting Filter
- Front Controller
- Context Object
- App Controller
- View Helper
- Composite View
- Service to Worker
- Dispatcher View

Business Tier
- Business Delegate
- Service Locator
- Session Facade
- Application Service
- Business Object
- Composite Entity
- Transfer Object
- TO Assembler
- Value List Handler

Integration Tier
- Data Access Object
- Service Activator
- Domain Store
- Web Service Broker
Front Controller

Intent

- provide a single point for processing user requests

http://java.sun.com/blueprints/corej2eepatterns/Patterns/FrontController.html
Motivation

- **single processing point for all client requests**
  - across views & session
  - can be used to inject cross-cutting concerns
    - logging
    - security
  - separation of business code from presentation code

- provides logical resource mapping
  - http://server/resource.jsp
  - http://server/servlet/resourceController
  - actually we can map multiple requests to the same controller
Front Controller

Motivation

• reusability & organic growth
  – controllers can be specialized (sub-classing)
  – dynamically or declaratively mapped
  – what is declarative mapping
  – why do we care?
Intercepting Filter

Intent

- preprocessing & post processing of user requests

http://java.sun.com/blueprints/corej2eepatterns/Patterns/InterceptingFilter.html
How is this different from pipes & filters?
Intercepting Filter

Motivation

- functionality injection
- improved reusability
  - filter chains can be defined in a number of ways
- declarative configuration

Difficulties

- information sharing
- fault-tolerance
**Data Access Object (DAO)**

**Intent**
- abstract access to data repository

http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html
Concerns

Mapping

- **object to persistence mapping**
  - identifier, foreignkey
  - object to single table
  - how do we map associations
    - object to object
    - object to list of objects
    - object to map of objects
  - how do we map inheritance
Concerns

Mapping

• deletions
  – single object deletions are easy
  – deleting associated objects could cause violations

• mappers
  – allow to vary schema differently from object model
Mappers

- allow to vary schema differently from object model
ER translates into

- tables, columns
- each table row represents an entity