CS 619 Introduction to OO Design and Development

More Patterns

Fall 2012

Pattern Review

Pattern Name and Classification: A descriptive and unique name that helps in identifying and referring to the pattern.

Intent: A description of the goal behind the pattern and the reason for using it.

Also Known As: Other names for the pattern.

Motivation (Forces): A scenario consisting of a problem and a context in which this pattern can be used.

Applicability: Situations in which this pattern is usable; the context for the pattern.

Structure: A graphical representation of the pattern. Class diagrams and Interaction diagrams may be used for this purpose.

Participants: A listing of the classes and objects used in the pattern and their roles in the design.

Collaboration: A description of how classes and objects used in the pattern interact with each other.

Consequences: A description of the results, side effects, and trade offs caused by using the pattern.

Implementation: A description of an implementation of the pattern; the solution part of the pattern.

Sample Code: An illustration of how the pattern can be used in a programming language.

Known Uses: Examples of real usages of the pattern.

Related Patterns: Other patterns that have some relationship with the pattern; discussion of the differences between the pattern and similar patterns.

Classification

Creational	Structural	Behavioral
Simple Factory	Adapter	Chain of responsibility
Abstract Factory	Bridge	Command
Factory method	Composite	Interpreter
Singleton	Decorator	Iterator
Builder	Façade	Mediator
Prototype	Flyweight	Memento
	Proxy	Observer
		State
		Strategy
		Template method
		Visitor

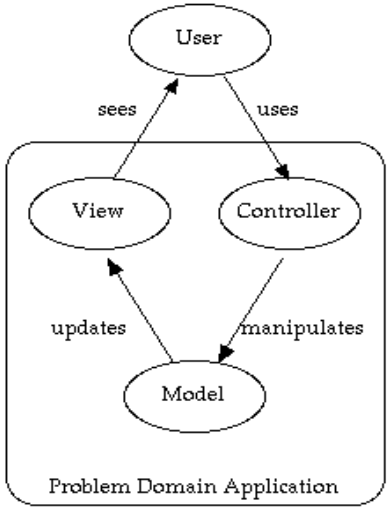
Classification II

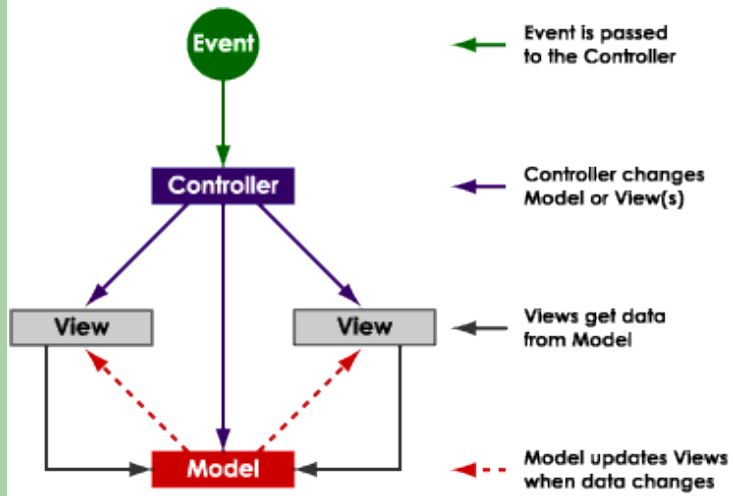
Concurrency	J2EE	Architectural
Active Object	Business Delegate	Layers
Balking	Composite Entity	Presentation-abstraction-control
Double checked locking	Composite View	Three-tier
Guarded suspension	Data Access Object	Pipeline
Monitor object	Fast Lane Reader	Implicit invocation
Reactor	Front Controller	Blackboard system
Read/write lock	Intercepting Filter	Peer-to-peer
Scheduler	Model-View-Controller	
Event-Based Asynchronous	Service Locator	Service-oriented architecture
Thread pool	Session Facade	Naked objects
Thread-specific storage	Transfer Object	
	Value List Handler	
	View Helper	

Three Modules of MVC

- The MVC pattern cleanly separates the modeling of the domain, the presentation, and the actions based on user input into three separate models
  - Model
    - The core of the application. This maintains the state and data that the application represents. When significant changes occur in the model, it updates all of its views.
  - View
    - The UI which displays information about the model to the user. Any object that needs information about the model needs to be a registered view with the model.
  - Controller
    - The UI presented to the user to manipulate the application.

Three Modules of MVC





MVC Advantages

- Clarity of design
- Efficient modularity
- Multiple views
- Ease of growth
- Distributable
- Powerful user interfaces

N tier Architecture

- The n tier architecture is based on the concept of separating a system to different layers (usually 3)
- Each layer interacts with only the layer directly below, and has specific function that it is responsible for.
- Classic for IT systems

3 tier Architecture

- **Presentation Layer**: the layer responsible for displaying user interface.
- **Business Tier**: the layer responsible for accessing the data tier to retrieve, modify and delete data to and from the data tier and send the results to the presentation tier.

  BLL and DAL
  Often this layer is divided into two sub layers: the Business Logic Layer (BLL), and the Data Access Layers (DAL). Business Logic Layers are above Data Access Layers, meaning BLL uses DAL classes and objects. DAL is responsible for accessing data and forwarding it to BLL.

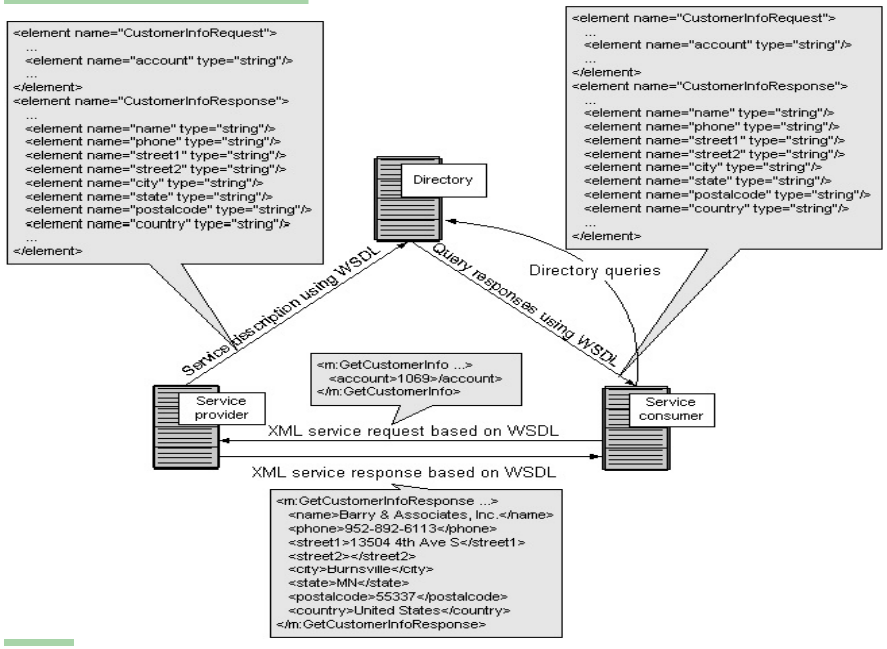
- **Data Tier**
  Data tier is the database or the source of the data itself.

Service Oriented Architecture (SOA)

- SOA is an architectural style whose goal is to achieve loose coupling among interacting software agents.
- A service is a unit of work done by a service provider to achieve desired end results for a service consumer
- in SOA, services are the mechanism by which needs and capabilities are brought together.

SOA - Principles

- Visibility – the ability for a service consumer to “see” a service provider (Awareness, willingness and Reachability)
- Interaction - the activity of using a capability. (usually message exchange - by contracts, constraints and policies, for example Web Service Description Language)
- Effect – the result of an interaction


<element name="CustomerInfoRequest">
...
<element name="account" type="string"/>
...
</element>
<element name="CustomerInfoResponse">
...
<element name="name" type="string"/>
<element name="phone" type="string"/>
<element name="street1" type="string"/>
<element name="street2" type="string"/>
<element name="city" type="string"/>
<element name="state" type="string"/>
<element name="postalcode" type="string"/>
<element name="country" type="string"/>
...
</element>
Directory
Service description using WSDL
Query responses using WSDL
Directory queries
<element name="CustomerInfoRequest">
...
<element name="account" type="string"/>
...
</element>
<element name="CustomerInfoResponse">
...
<element name="name" type="string"/>
<element name="phone" type="string"/>
<element name="street1" type="string"/>
<element name="street2" type="string"/>
<element name="city" type="string"/>
<element name="state" type="string"/>
<element name="postalcode" type="string"/>
<element name="country" type="string"/>
...
</element>
<m:GetCustomerInfo ...>
<account>1069</account>
</m:GetCustomerInfo>
Service provider
Service consumer
XML service request based on WSDL
XML service response based on WSDL
<m:GetCustomerInfoResponse ...>
<name>Barry & Associates, Inc.</name>
<phone>952-892-6113</phone>
<street1>13504 4th Ave S</street1>
<street2></street2>
<city>Burnsville</city>
<state>MN</state>
<postalcode>55337</postalcode>
<country>United States</country>
</m:GetCustomerInfoResponse>