Design Patterns

MSc Computer Science

Produced by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology
http://www.wit.ie
http://elearning.wit.ie

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning
support unit

Layering & Domain Logic

The Evolution of Layers in Enterprise Applications

- 60s-70s: Batch Systems (no layers)

- 80s-90s: Client Server (2 layers)

- 00's :Enterprise Web (3 or more layers)

Client Server

- Layers became more apparent in the '90s with the rise of client–server systems.

    - Two-layer systems: The client held the user interface and other application code, and the server was usually a relational database.

    - Common client tools were VB, Powerbuilder, and Delphi.

    - These made it particularly easy to build data-intensive applications, as they had UI widgets that were aware of SQL.

- Thus you could build a screen by dragging controls onto a design area and then using property sheets to connect the controls to the database.

3 + Layers

- Initial impetus from deploy client–server applications with a Web browser.

- However, if all your business logic was buried in a rich client, then all your business logic needed to be redone to have a Web interface.

- A well-designed three-layer system required a new 'presentation layer'

- With the arrival of Java as a main stream 'enterprise' language, we .. the tools that appeared to build Web pages were much less tied to SQL and thus more amenable to a third layer.

The Three Principle Layers

Layer	Responsibilities
Web Presentation	*Responsibilities*
Web Presentation	Provision of services, display of information (e.g., in Windows or HTML, handling of user request (mouse clicks, keyboard hits), HTTP requests, command-line invocations, batch API)
Domain Logic	Logic that is the real point of the system
Data Source	Communication with databases, messaging systems, transaction managers, other packages

Web Presentation

- Handle the interaction between the user and the software.

- May be a rich-client graphics UI or an HTML-based browser UI.

- Primary responsibilities are to display information to the user and to interpret commands from the user into actions upon the domain and data source.

Domain Logic

- Also also referred to as business logic.

- The work that this application needs to do for the domain you're working with.

- It involves calculations based on inputs and stored data, validation of any data that comes in from the presentation, and figuring out exactly what data source logic to dispatch, depending on commands received from the presentation.

Data Source

- Communicating with other systems that carry out tasks on behalf of the application.

- These can be transaction monitors, other applications, messaging systems, etc…

- For most enterprise applications this is a database that is primarily responsible for storing persistent data.

Enterprise Patterns

Session State
- Client Session
- Server Session
- Database Session

Offline Concurrency
- Optimistic Lock
- Pessimistic Lock
- Coarse Grained Lock
- Implicit Lock

Distribution
- Remote Facade
- Data Transfer Object

O/R Metadata Mapping
- Metadata Mapping
- Query Object
- Repository

Domain Logic
- Transaction Script
- Domain Model
- Table Module
- Service Layer

O/R Structure
- Mapping
  - Identity Field
  - Foreign Key
  - Association
  - Dependent
  - Inheritance
- Embedded Value
- Serialized LOB
- Inheritance
  - Single Table
  - Class Table
  - Concrete Table

Base
- Special Case
- Money
- Plugin
- Mapper
- Separated Interface
- Service Stub
- Gateway
- Registry
- Value Object
- Record Set
- Layer Supertype

Web Presentation
- View
  - Transform
  - Template
  - Two Step
- Controller
  - Application
  - Page
  - Front
- MVC

Data Source
- Gateway
  - Table Data
  - Row Data
- Active Record
- Data Mapper

O/R Behaviour
- Lazy Load
- Identity Map
- Unit of Work

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning
support unit