This Lecture

E Gamma, R. Helm, R. Johnson and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

Distribution

- Distributed Objects vs. Clustering
- Patterns:
  - *Remote Facade*
  - *Data Transfer Object*
- Use of Web Services
Layers (as usual)

- Top-down:
  - Presentation
  - Domain
  - Data Source

- Order of design decisions:
  - Domain
  - Data Source
  - Presentation
Some Patterns

- Domain Layer
  - Transaction Script
  - Domain Model
  - Table Module
- Data Source Layer
  - Row Data Gateway
  - Table Data Gateway
  - Record Set
  - Active Record
  - Data Mapper
Some Patterns

- Presentation Layer
  - Model View Controller
  - Controller
    - Page Controller
    - Front Controller
  - View
    - Template View
    - Transform View
Table Data Gateway

An object that acts as a gateway to a database table. One instance handles all rows in the table.

<table>
<thead>
<tr>
<th>Person Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>find(id) : RecordSet</td>
</tr>
<tr>
<td>update(id, lastName, firstName, numDependents)</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>
Row Data Gateway

An object that acts as a gateway to a database table. One instance for each row in the table

Finder object to look up instances

<table>
<thead>
<tr>
<th>Person Finder</th>
</tr>
</thead>
<tbody>
<tr>
<td>find(id)</td>
</tr>
<tr>
<td>findByLastName(String)</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Person Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>lastname</td>
</tr>
<tr>
<td>firstname</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>insert</td>
</tr>
<tr>
<td>update</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>
Active Record

An object that wraps a row in a database table, and *adds domain logic* onto that data.

Separate finder class is optional
Data Mapper

A layer that moves data between objects and a database while keeping them independent of each other.
Domain Logic Patterns

- Revenue Recognition Problem

<table>
<thead>
<tr>
<th>Product</th>
<th>Contract</th>
<th>Rev Recog</th>
</tr>
</thead>
<tbody>
<tr>
<td>type (W,D,S)</td>
<td>date signed revenue</td>
<td>amount date</td>
</tr>
</tbody>
</table>

Rules for different product types:

W: all right away
S: 1/3 now, 1/3 60 days, 1/3 90 days
D: 1/3 now, 1/3 30 days, 1/3 60 days
Transaction Script

Organize business logic by procedures

Each procedure handles a single request from the presentation

```
Recognition Svc

recognizedRevenue(contractNo, effectiveDate): Money
computeRecognitions(contractNo)

DB1
```
Domain Model

An object model of the domain that incorporates both behavior and data
Table Module

A single instance that handles the business logic for all rows in a database table
Model-View-Controller

Web Svr → Ctrlr → HTTP Req

Req → Get data
Create with data → Execute domain logic
Add data for view → Model Objects
Fwd to View → Generate reply
Reply → Get data
Fwd to View → View