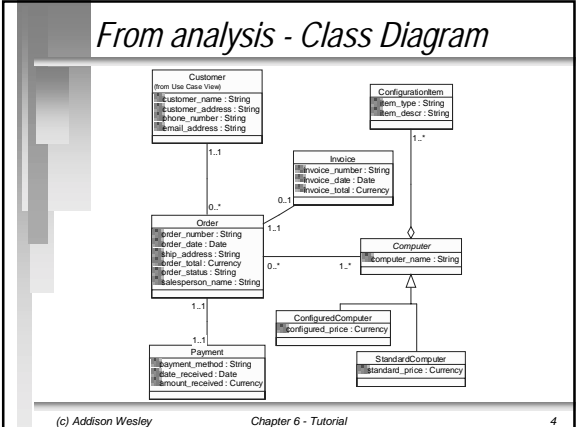


MACIASZEK, L.A. (2001): *Requirements Analysis and System Design. Developing Information Systems with UML*, Addison Wesley

Chapter 6 - Tutorial
Guided Tutorial in Design Modeling
OnLine Shopping

Copyright © 2000 by Addison Wesley
Version 1.0



Topics

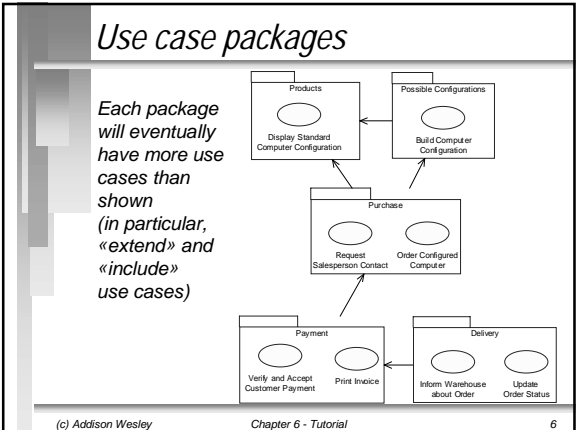
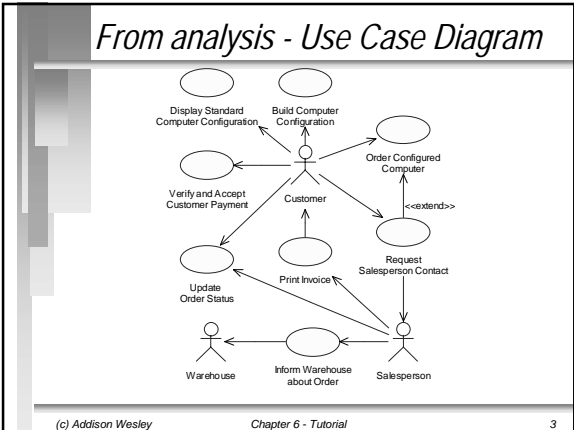
- **Package Design**
 - Use Case Packages
 - Class Packages
- **Component Design**
- **Deployment Design**
- **Collaboration Design**

(c) Addison Wesley Chapter 6 - Tutorial 2

Package design

- **Package - groups classes, use cases or other modeling elements**
- **Useful in large systems**
- **We distinguish between:**
 - Use Case packages – emphasized in analysis
 - Class Packages – emphasized in design

(c) Addison Wesley Chapter 6 - Tutorial 5



Boundary class packages

- Most classes that we defined in analysis represented **persistent database objects** ("business objects")
- BCED application program classes** need to be considered as well

The functions of configuring computers and entering orders require Boundary Packages

```

classDiagram
    class ConfigurationGUI["<<boundary>> Configuration GUI"]
    class OrderGUI["<<boundary>> Order GUI"]
  
```

(c) Addison Wesley Chapter 6 - Tutorial 7

DB interface class packages

- To mediate between entity classes and the database
- To handle connections, authorizations, transactions
- To hold "meta-information" about DB schema

```

classDiagram
    class CRUD["<<db interface>> CRUD"]
    class Connection["<<db interface>> Connection"]
    class Schema["<<db interface>> Schema"]
    CRUD --> Connection
    CRUD --> Schema
  
```

(c) Addison Wesley Chapter 6 - Tutorial 10

Entity class packages

- Persistent database classes** correspond to Entity Classes in the application program
- Entity Packages** represent in-memory run-time structure for persistent database classes

```

classDiagram
    class Computers["<<entity>> Computers"]
    class Customers["<<entity>> Customers"]
    class Orders["<<entity>> Orders"]
    Computers --> Customers
    Customers --> Orders
    Orders --> Customers
  
```

(c) Addison Wesley Chapter 6 - Tutorial 8

Component design

- Components** – physical parts of the system
- Component design** refers to the implementation platform for the system
- OnLine Shopping** – Web application with database server
- Web application**
 - “... Web system that allows its users to execute business logic with a web browser.”
 - Business logic can reside on the server and/or on the client
 - Client/Server system with a Web site

(c) Addison Wesley Chapter 6 - Tutorial 11

Control class packages

- Control classes** → Control Packages
 - represent application logic
 - “glue” boundary and entity classes

```

classDiagram
    class ConfigureProcess["<<control>> Configure Process"]
    class OrderPlacement["<<control>> Order Placement"]
  
```

(c) Addison Wesley Chapter 6 - Tutorial 9

Implementing Web applications

- Web pages**
 - Rendered in Internet client **browser**
 - Delivered by **Web server**
- Web page document**
 - can be static (unmodifiable) or dynamic
 - can be a **form** that a user fills in
- Frames**
 - divide the screen's "real estate" so that the user can view multiple Web pages at the same time
- Application server**
 - to manage the application logic
 - to monitor the application state
 - By storing **cookies** in the browser
 - Session timeouts

(c) Addison Wesley Chapter 6 - Tutorial 12

Implementing Web applications

- **Dynamic client pages**
 - **Script** – program interpreted by the browser
 - **Applet** - compiled component that executes in the browser's context
- **Server pages** - Web pages with **scripts** executed by the server
 - Have access to DB server
 - Manage client sessions
 - Place cookies on the browser
 - Build client pages

(c) Addison Wesley Chapter 6 - Tutorial 13

Deploying Web applications

- **Four tiers of computing nodes**
 - **Client with browser**
 - Static and dynamic pages
 - Scripted pages and applets downloaded and run within the browser
 - **Web server**
 - Page requests from the browser
 - Generation of pages and code for execution on the client
 - **Application server**
 - Necessary with distributed objects
 - **Database server**
 - Data storage
 - Multi-user access

(c) Addison Wesley Chapter 6 - Tutorial 16

Component diagram

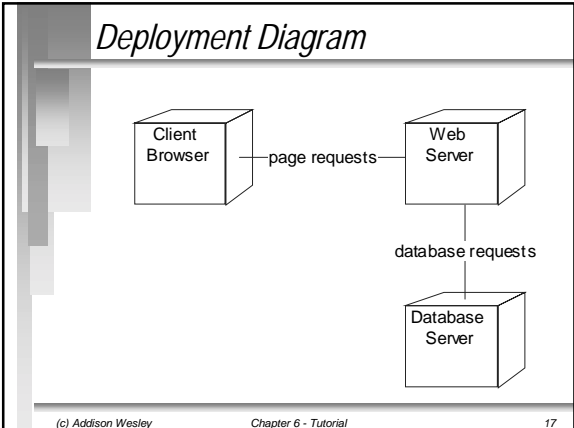
- **Component**
 - Cohesive functional unit with clear interfaces
 - Replaceable part of the system
 - Can correspond to implementation of one or more Web pages
 - Can parallel Use Case Packages

```

classDiagram
    class ProductList
    class ProductDisplay
    class Configuration
    class Purchase
    class OrderTracking

    ProductList ..> ProductDisplay
    Configuration ..> ProductList
    Purchase ..> Configuration
    OrderTracking ..> Purchase
  
```

(c) Addison Wesley Chapter 6 - Tutorial 14



Deployment design

- **Assignment of objects to computing nodes**
- **Difficulties related to Web applications**
 - Connectionless nature of Internet
 - Session management
 - Cookies
 - Distributed objects (CORBA, DCOM, EJB)
 - Application server between Web server and DB server
 - Web server as the routing point between all client browsers and the database
 - Security
 - Network loads, backups, etc.

(c) Addison Wesley Chapter 6 - Tutorial 15

Collaboration design

- **Architectural design** ≡ packages, components
- **Detailed design** ≡ collaboration design
- **Collaborations** define the **realization** of
 - Use cases
 - Operations
- **Collaboration design** is conducted in parallel with the **elaboration** of
 - Use case models
 - Class models
 - most other models

(c) Addison Wesley Chapter 6 - Tutorial 18

Elaborating use cases

Use Case Specification: Order Configured Computer

1. [UC15 Order Configured Computer]

1.1 Brief Description
A customer fills in and submits a purchase order form. The system verifies the details and confirms or rejects the order.

2. Flow of Events

2.1 Basic Flow

2.1.1 [UC15.1 The system displays the Order Entry form in the customer's Web browser. The form contains the following items:]

- [UC15.1.1 The title of the form is "Order Your Computer".]
- [UC15.1.2 Explanatory information is displayed below the title.

The text for explanatory information is:
"Please fill out the boxes in the form. Prompts for required items are in red color and in boldface. Press the Submit button to submit the form or Cancel button if you decide not to proceed with your order. You can cancel your order without penalty within 24 hours from the order's submission. You can cancel submitted order by using Web, email, fax or phone."

The document "Use Case Specification: Update Order Status" describes how order can be

(c) Addison Wesley Chapter 6 - Tutorial 19

Using BCED approach

- Recommended practice - **prefixing the class names with letters**
 - b (Boundary), e.g. b_OrderClientPage
 - c (Control)
 - e (Entity)
 - d (Database Interface)
- Association and aggregation relationships to link BCED classes**
- Instantiation relationships to signify messages that instantiate objects**
 - User events leading to object instantiation can be named

(c) Addison Wesley Chapter 6 - Tutorial 22

Requirements management

(c) Addison Wesley Chapter 6 - Tutorial 20

Boundary classes

- For use case "Order Configured Computer"

(c) Addison Wesley Chapter 6 - Tutorial 23

Structure of collaboration

- Structure of collaboration \equiv collaboration Class Diagram extended with application program classes (BCED classes)
- Adheres to the enabling technology chosen for the application
- Difficulty:
 - The enabling technology may not be OO
- OnLine Shopping
 - Boundary classes – client pages, forms
 - Control classes – server pages

(c) Addison Wesley Chapter 6 - Tutorial 21

Control and entity classes

- For use case "Order Configured Computer"

(c) Addison Wesley Chapter 6 - Tutorial 24

