Chapter 6 - Tutorial

Guided Tutorial in Design Modeling
OnLine Shopping

Topics

- Package Design
  - Use Case Packages
  - Class Packages

- Component Design

- Deployment Design

- Collaboration Design

From analysis - Use Case Diagram
**From analysis - Class Diagram**

- **Customer**
  - `customer_name : String`
  - `customer_address : String`
  - `phone_number : String`
  - `email_address : String`

- **Configured Computer**
  - `configured_price : Currency`

- **Standard Computer**
  - `standard_price : Currency`

- **Payment**
  - `payment_method : String`
  - `date_received : Date`
  - `amount_received : Currency`

- **Invoice**
  - `invoice_number : String`
  - `invoice_date : Date`
  - `invoice_total : Currency`

- **Configuration Item**
  - `item_type : String`
  - `item_descr : String`

- **Computer**
  - `computer_name : String`

- **Order**
  - `order_number : String`
  - `order_date : Date`
  - `ship_address : String`
  - `order_total : Currency`
  - `order_status : String`
  - `salesperson_name : String`


---

**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

---

**Use case packages**

Each package will eventually have more use cases than shown (in particular, «extend» and «include» use cases)
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

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

Control class packages
- Control classes → Control Packages represent application logic
  "glue" boundary and entity classes

MACIASZEK (2001): Req Analysis & Syst Design
**DB interface class packages**

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

**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

**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
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

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 Case Packages

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.
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

Deployment Diagram

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
Elaborating use cases

Use Case Specification: Order Configured Computer

1. [UC15 Order Configured Computer]

1.1 Brief Description
A customer (BC) and seller (syst) purchase order items. This use case handles the AddItem and CancelItem commands to
modify the order.

2. Flow of Events
2.1 Basic Flow

2.1.1 [Begin] The system displays the Order Form to the customer (BC) and seller (Syst).
2.1.2 The system displays a selection of items for each category.
2.1.3 The system displays the order form.
2.1.4 The system displays the order form again.

2.2 Extended Flow

2.2.1 [Begin] The user selects an item from the Order Form.
2.2.2 The system displays the selected item.
2.2.3 The system displays the selected item again.

Requirements management

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
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

**Association relationship**

**Aggregation relationship**

**Boundary classes**

- **For use case “Order Configured Computer”**

**Control and entity classes**

- **For use case “Order Configured Computer”**
Behavior of collaboration

For use case "Order Configured Computer"

Boundary and control objects

- Note from structural collaboration that b_OrderClientPage contains b_ConfigurationClientPage.

Entity and DB Interface objects

- e_Configuration
- e_Customer
- e_Order
- e_Payment
- d_Transaction
  - beginTransaction
  - commitTransaction
  - link(in ConfOID)
  - newPayment
  - newCustomer
  - emailCustomer
  - newOrder(in ConfOID)
  - deleteOrder

[over 15 min] rollbackTransaction