Week 10: Morphing Robustness
Diagrams into Sequence Diagrams

Adapted from ``UML for e-Commerce'' by Doug Rosenberg
http://www.iconixsw.com/uml_for_e-commerce.ppt
Sequence Diagrams

We allocate methods to classes as we draw sequence diagrams.
Before we do sequence diagrams, though...

- We need to have a good idea about what objects will be performing in which use case, and what functions the system will perform as a result of user actions.

- We get this information from robustness diagrams, the result of robustness analysis.
Robustness Diagrams -- the missing link!

We discover new objects, and add attributes to classes, as we draw robustness diagrams.
But we can’t draw robustness diagrams before...

- We describe system usage *in the context of the object model*.
- This means that we don’t write abstract, vague use cases that we can’t design from.
- Instead, we need to write use case text that references the names of objects in the problem domain.
- We also reference the names of "boundary objects" in the use case text.
First, though...

- We need to identify the main abstractions that are present in the problem domain.
- In other words, we need a domain model.
- We show our domain model on class diagrams.
Refining our class diagrams

We'll refine our (static) analysis level class diagrams (our domain model) continuously as we explore the dynamic behavior of the system in more and more detail during analysis and design.

This will ultimately result in our design-level class diagrams, which we can code from.
The ICONIX Process

GUI Prototype → Use Case Model → Robustness Diagram → Sequence Diagram

Dynamic

Static

Domain Model → Class Diagram → Code
The Internet Bookstore Example

- Domain Model
- Use Case Model
- 2 use cases: Login, Edit Shopping Cart
- Robustness and Sequence Diagrams for each use case
- Show common errors (Wrong way / Right way)
Requirements

- The bookstore shall accept orders over the Internet.
- The bookstore shall maintain a list of accounts for up to 1,000,000 customers.
- The bookstore shall provide password protection for all accounts.
- The bookstore shall provide the ability to search the master book catalog.
- The bookstore shall provide a number of search methods on that catalog, including search by author, search by title, search by ISBN number, and search by keyword.
- The bookstore shall provide a secure means of allowing customers to pay by credit card.
- The bookstore shall provide a secure means of allowing customers to pay via purchase order.
- The bookstore shall provide a special kind of account that is preauthorized to pay via purchase order.
- The bookstore shall provide electronic links between the Web and database and the shipping fulfillment system.
- The bookstore shall provide electronic links between the Web and database and the inventory management system.
- The bookstore shall maintain reviews of books, and allow anyone to upload review comments.
- The bookstore shall maintain ratings on books, based on customer inputs.
Domain Model
Use Case Model

- Browse List of Books
- Edit Contents of Shopping Cart
- Log In
- Cancel Order
- Search by Author
- Check Out
- Open Account
- Track Recent Orders
- Customer Actor
- Shipping Clerk
- Shipping Station
- Ship Order
- Shipper
Login: use case text

- Basic Course: The Customer enters his or her user ID and password, and then clicks the Log In button. The system validates the login information against the persistent Account data, and then returns the Customer to the Home Page.

- Basic Course: The Customer enters his or her user ID and password, and then clicks the Log In button....
Login: robustness diagram

[Diagram showing the login process with nodes and arrows indicating the steps: Customer, Login Page, Validate, Display, Open Account, Reminder Word Dialog Box, Home Page.]
Login: bad sequence diagram

Basic Course

The Customer clicks the Log In button on the Home Page. The system displays the Login Page. The Customer enters his or her user ID and password, and then clicks the Log In button. The system validates the login information against the persistent Account data, and then returns the Customer to the Home Page.

Alternate Courses

If the Customer clicks the New Account button on the Login Page, the system invokes the Open Account use case.

If the Customer clicks the Reminder Word button on the Login Page, the system displays the reminder word stored for that Customer, in a separate dialog box. When the Customer clicks the OK button, the system returns the Customer to the Login Page.

If the Customer enters a user ID that the system does not recognize, the system displays a message to that effect and prompts the Customer to either enter a different ID or click the New Account button.

If the Customer enters an incorrect password, the system displays a message to that effect and prompts the Customer to reenter his or her password.

If the Customer enters an incorrect password three times, the system displays a message...
Login: good sequence diagram
Edit Shopping Cart use case text

- **Basic Course:** On the Shopping Cart Page, the Customer modifies the quantity of an Item in the Shopping Cart, and then presses the Update button. The system stores the new quantity, and then computes and displays the new cost for that Item....

- **Alternate Course:** If the Customer changes the quantity of the Item to 0, the system deletes that Item from the Shopping Cart.
Robustness diagrams bridge the “what/how” gap

Most current UML texts do not address crossing this what/how gap.
Edit shopping cart robustness diagram
Starting a sequence diagram

Use case text is refined during robustness analysis and reviewed during the preliminary design review.

1. Copy the use case text to the left margin of the sequence diagram.
2. Add the entity objects.
3. Add the boundary objects.
4. Work through the controllers, one at a time, and figure out how to allocate the behavior among the collaborating objects.

The user requirements are always visible as we work through the design of the system.
Edit shopping cart sequence diagram

Use Case View: Customer

<< Abstraction >>

Basic Course
On the Shopping Cart Page, the Customer modifies the quantity of an Item in the Shopping Cart, and then presses the Update button.
The system stores the new quantity, and then computes and displays the new cost for that Item.
The Customer presses the Continue Shopping button. The system returns control to the use case from which it received control.

Alternate Courses
If the Customer changes the quantity of the Item to 0, the system deletes that Item from the Shopping Cart.
If the Customer presses the Delete button instead of the Update button, the system deletes that Item from the Shopping Cart.
Use the Sequence Diagram to Allocate Behavior

- *Which class does an operation belong in?*

Halbert and O’Brien criteria:
- Reusability: does it make this class more general?
- Applicability: does it fit? Is it relevant?
- Complexity: is it easier to build it here or elsewhere?
- Implementation knowledge: does it rely on internal details?
Update your static model, again