CS48 – UML and Sequence Diagram Examples

Chandra Krintz
Department of Computer Science
UC Santa Barbara
http://www.cs.ucsb.edu/~ckrintz/
Parnas’ Modularization

- Define your set of data structures

Foreach data_structure
  - Define the set of possible operations on it (as functions)
    - Encapsulate code and data
    - Make public the set of functions that other modules or users employ to interact with the data structure
    - Make everything else (code and data) private

- Make each data structure reusable and extensible (inheritance)
  - And customizable (polymorphism)
The Unified Modeling Language (UML)

- A tool for all phases of software development and design
  - Use it for designing your modules and interfaces
- Many books on UML, some good ones are:
  - “UML Distilled,” Martin Fowler
  - “UML Explained,” Kendall Scott
  - “Using UML,” Perdita Stevens
- The Object Management Group (OMG, a computer industry consortium) defines the UML standard
  - The current UML language specification is available at:
    http://www.uml.org/

- Tools:
  - http://www.visual-paradigm.com/solution/freeumldesigntool/
  - http://yuml.me (online tool)
UML Diagram Relationships

**Association**
- From: http://creately.com/blog/diagrams/class-diagram-relationships/

**Directed Association**
- A container/contained directional flow

**Reflexive Association**
- Multiple functions + relationship between 2 instances of same class

**Multiplicity**
- Relationship with 0-to-many Passenger instances

**Aggregation**
- A container/contained directional flow
- Built as a collection

**Composition**
- A child/parent directional flow

**Inheritance**
- A child/parent directional flow

**Realization**
- A child/parent directional flow
Annotations

For any relationship (edge between classifiers), we can annotate:

- The name of the relationship (may be directional – indicated with a solid arrowhead in the direction the relationship holds)
- The role of target instance in the source
- Cardinality constraints (1:N, N:M, etc.) at either end
- Possible ordering at either end

1 
* 
0..1 
m..n 
{ordered}* 

Exactly one 
Many (any number) 
Optional (zero or one) 
Specified range 
Ordered
Nontrivial Example
Nontrivial Example

Copyright 2002-2006 Scott W. Ambler

Small solid triangle placed in front of association name to show order of the association
Sequence Diagrams: Basic Idea

• Illustrate interactions between objects over time
• Show behavior as opposed to static design
• Dashed lines are called Object Lifetimes
• Messages between objects are method/function calls
  • Can be asynchronous or synchronous
Common Stereotypes

- Non-objects that are intrinsic to the interactions
- Can include users, databases, etc.
Example 1

The ShoppingCart and Order objects already exist when the interaction starts.

- **addItem(i)**
  - `new(i)`
  - `o1:OrderLine`
  - An OrderLine object is created.
  - The lifeline indicates its lifetime.

- **remove(o1)**
  - <<destroy>>
  - The OrderLine object is destroyed.
Example 3

1. Opens the iBooklet
   1.1: Displays Log In Page
2. Selects Sign Up
   2.1: Displays Sign Up Page
   3: Enters Credentials
   3.1: Sends Credentials Info
      3.1.1: Checks if Valid
      3.1.2: Creates New User
      3.1.3: Sends Creation Confirmation
   3.1.3.1: Displays Log In Page
   4: Enters Log In Credentials
      4.1: Sends Log In Info
         4.1.1: Checks if Valid
         4.1.2: Sends Login Confirmation
   4.1.2: Sends Login Confirmation
   4.1.2: Sends Login Confirmation
5. Redirects To Book Selection Screen