INF 117
Project in Software Engineering

Lecture Notes – Winter Quarter, 2008

Michele Rousseau

Set 7 – Going from Design to Code

Announcements

FEBRUARY 2008

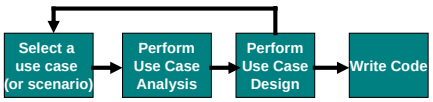
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
10 Week 6	11 Des. Iter: #2	12 Lincoln's B-Day	13	14 Valentines Day	15 Cust. Milestone: Des. Approved Des Iter: #3 Test-Plan It #2 (Incl Des)	16
17 Week 7	18 President's Day Code Iter #1 Proj. Plan #3	19	20	21 Stud. Pres-Des Order 3,4,1,2 Sub. Ass: #3 Peer Eval #2	22 Team Log #3	23

Reminder:

Can submit Subjective Assessments online (prior to due date)

Happy Valentine's Day

From Use Cases to Code



Use Case Analysis

For each use case in an iteration…

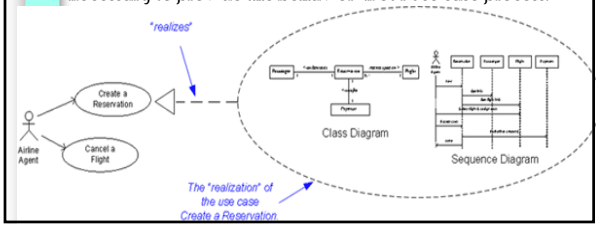
1. Create a use case realization
2. Supplement the Use-Case descriptions
   - if necessary
3. Find Analysis Classes from Use-Case Behavior
4. Distribute Behavior to Analysis Classes

1. Use-Case Realization

A ***use-case realization*** is a collection of UML diagrams which together validate that we have

- the classes → Class Diagrams (static relationships)
- responsibilities
- object interactions → Interaction Diagrams (dynamic relationships) – could be Sequence or Collaboration

necessary to provide the behavior in our use case process.



2. Supplement the Use-Case descriptions (if necessary)

- Beef up your use-case descriptions
  - Can include internal or non-visible behavior of the system
    - Careful not to include how… but what it will do
  - Do you need to do this for all of them?

    No! → Include just enough detail to understand the classes you will need

3. Find Analysis Classes from Use-Case Behavior &

- identify a candidate set of analysis classes
- Analysis Class
  - 3 Categories
    - Entity → Business level
      - Banking system → Customer, account, transaction (e-commerce or old school)
    - Controller → process & sequence aware
      - Control & direct the flow of control on an execution sequence
    - Boundary → I/O required by the s/w system

Describe the Class's Responsibilities

- Use nouns to determine classes

Class Name	Description	Responsibilities

4. Distribute Behavior to Analysis Class

- Sequence Diagrams
- Collaboration Diagrams

Next

For each resulting analysis class
Describe the Class's Responsibilities

- Describe the Class's Attributes and Associations
  - Define Class Attributes
  - Establish Associations between Analysis Classes
  - Describe Event Dependencies between Analysis Classes
- Establish Traceability
- Evaluate the Results of Use-Case Analysis

Other Notes

- Simplify your diagrams using subsystems
  - Packages can be used anywhere
- Use some underlying concepts
  - Abstraction
  - Encapsulation → Information hiding
    - Hide design decisions most likely to change
  - Polymorphism
    - Use Operations/functions in different ways