

Requirements Analysis

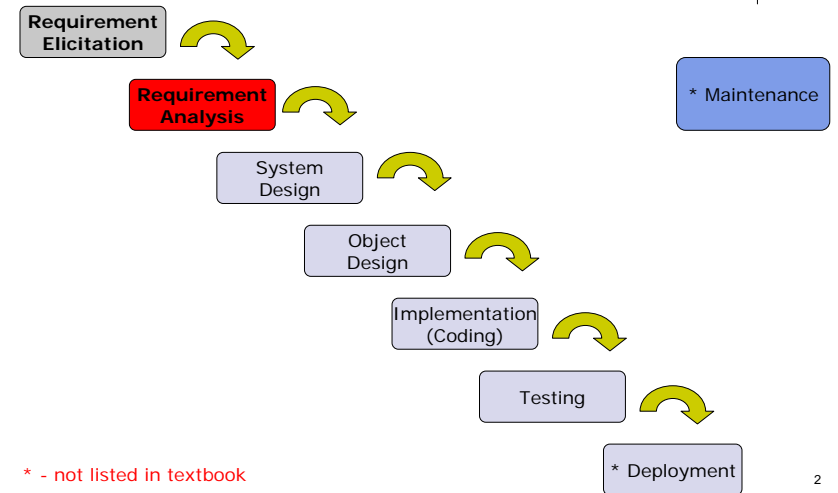
Lecture #06
Software Engineering and
Project Management

Instructed by Steven Choy on Nov 6, 2006



1

Where are we NOW?



2

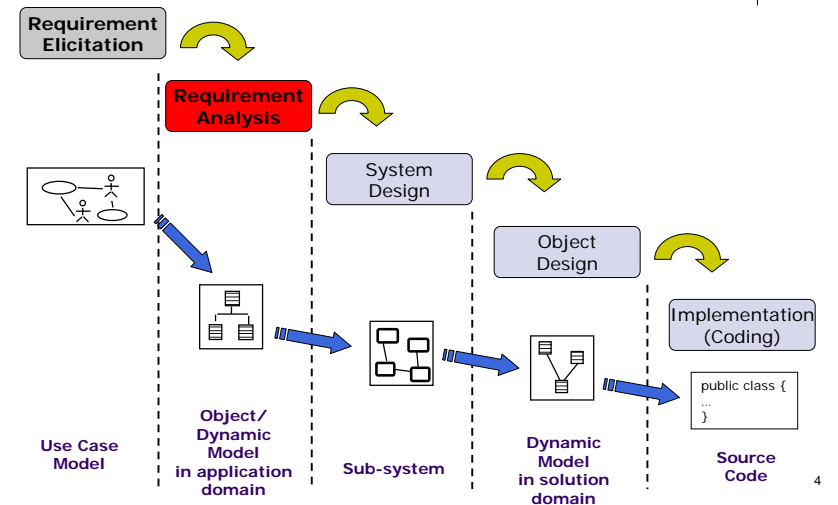
What we will discuss today?

- The Requirement Process
 - Requirement Elicitation (Chapter 4)
 - Gather requirement from our users
 - Requirement Analysis (Chapter 5)
 - Analyze the requirement and build a model for the potential system



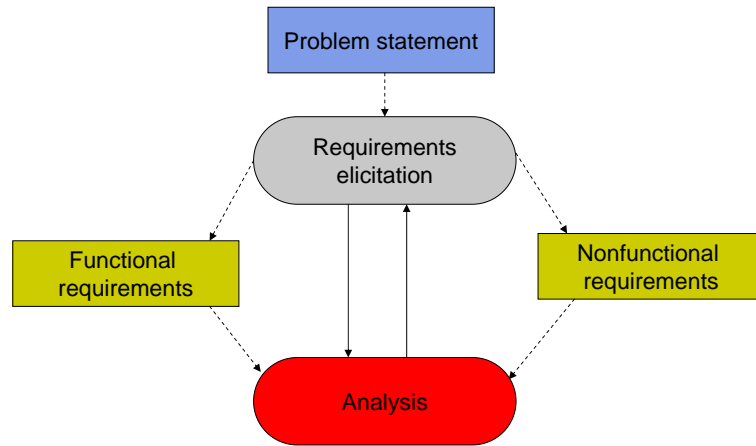
3

Software Development Activities



4

What does requirement elicitation consume and produce?



5

Requirement Analysis



- Focus on producing a model of the system
- Types of model to produce:
 - Functional Model – represented by use cases and scenarios
 - Analysis Object Model – represented by class diagrams
 - Dynamic Model – represented by sequence diagram
- **The focus of these models should be on user-level, not actual software components/classes for implementation**

6

Who need the Models?



- Is it for user or client?
 - Both probably are not interested in object model and class diagrams.
 - But by developing object model, you can develop a more thorough understanding of the future system and find out some hidden/ambiguous requirement, which need further clarification.
- Developer
 - Aid communication with developers
 - Give yourself a big picture

7

Object Model

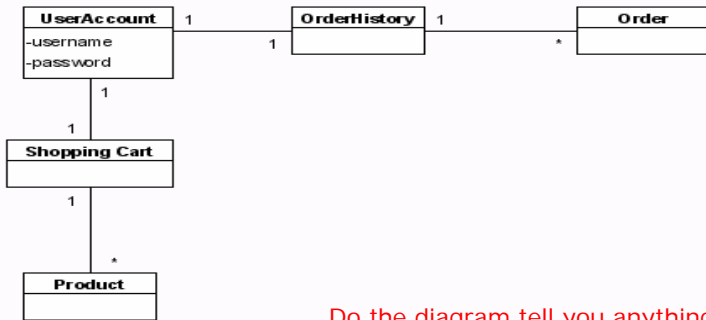


- Model the individual concept of the system, their relationships and properties using class and object diagram
- The goal of object modeling is to find classes

*The focus of these models should be at user-level, not implementation level.
That is, the actual software components/classes should not be considered*

8

Analysis Object Model



Do the diagram tell you anything
About implementation?

9

Steps for Object Modeling



- Class Identification
- Find the attributes
- Find the methods
- Find the associations between classes

10

How to find out Classes?



- Not easy if you are lack of experience
- But here are some guidelines to follow:
 - Start with use cases
 - Scan out the **nouns** from the "Flow of event" of use cases you developed
 - Not all nouns are actual classes
 - Filter those nouns using the filtering rules on the next page
- Abbott Textual Analysis
 - Nouns are good candidate of classes
 - Verbs are good candidate of operations
 - Adjectives are good candidate of attributes

11

Guidelines on Class Filtering



- Irrelevance
- Redundancy
 - Two nouns represent the same thing.
 - E.g. credit card, visa card and master card are referring to the same thing
 - Analyze those nouns that their primary roles are the same
- An attribute
 - Noun that is actually an attribute of a class.
 - E.g. credit card number
- An operation
 - Noun that actually describes the behaviour of a class.
 - E.g. Payment calculation

12

Guidelines of Class Filtering



- An association
 - Nouns that describes the relationship between classes
 - E.g. A shopping cart contains a list of product items
 - “List” is not a class
- An event
 - Nouns that describe a time frequency
 - E.g. Print payment report every mid-night
 - Mid-night should not be a class
- An implementation construct
 - Nouns that refer to a hardware
 - E.g. Printer, monitor

13

Mapping nouns to Class - Example



Part of speech	Model component	Examples
Proper noun	Instance	Peter
Common noun	Class	Customer
Doing verb	Operation	Submit payment
Being verb	Inheritance	Is a kind of
Having verb	Aggregation	Has, consists of

14

Entity, Control & Boundary Objects



15

Types of Objects



- In OO system, objects are generally grouped into 3 types:
 - Entity Objects
 - Represent the persistent information of the system (e.g. Player, Customer)
 - Boundary Objects
 - Serve as a boundary between the actors and the entity objects
 - Allow an actor to interact with the system
 - Control Objects
 - Coordinators of activity in the application domain

16



Why do we need to categorize objects into these three types?

ANS: Modularity



Let's consider this...

Sign Up

Already have a Slide account? [Sign In](#)

Email Address:
Don't worry, we don't spam people.

First Name:

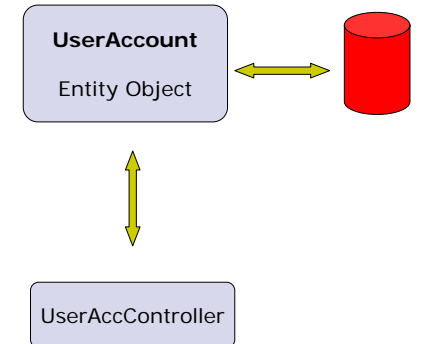
Last Name:

Username:
You will use the username to sign in to your Slide account.

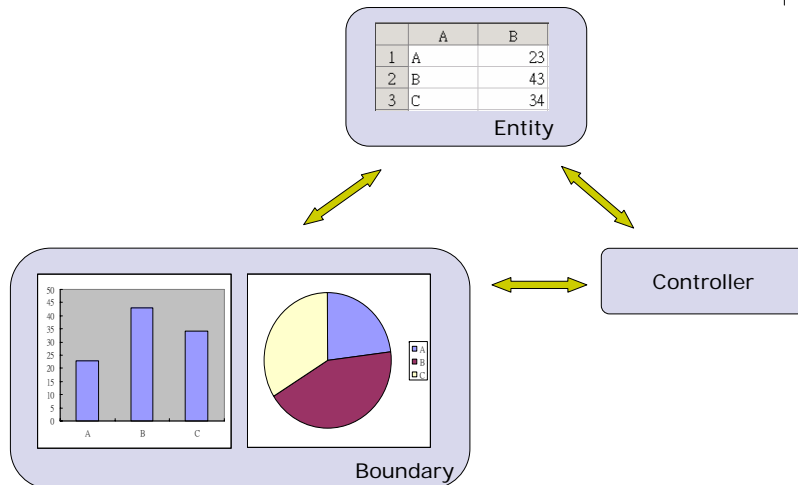
Password:

Confirm Password:

UserAccountForm



Again, let's consider this...



Identify Entity Object

- Use the methodology as described previously to look for classes
- But focus only on those noun phrases that are related to the application domain

ARENA: Identifying Entity Object



Use case name	AnnounceTournament
Flow of events	<ol style="list-style-type: none"> 1. The LeagueOwner requests the creation of a tournament. 2. The system checks if the LeagueOwner has exceeded the number of tournaments in the league or in the arena. If not, the system presents LeagueOwner with a form. 3. The LeagueOwner specifies a name, application start and end dates during which Players can apply to the tournament, start and end dates for conducting the tournament, and a maximum number of Players. 4. The system asks the LeagueOwner whether an exclusive sponsorship should be sought and, if yes, presents a list of Advertisers who expressed the desire to be exclusive sponsors. 5. If the LeagueOwner decides to seek an exclusive sponsor, he selects a subset of the names of the proposed sponsors. 6. The system notifies the selected sponsors about the upcoming tournament and the flat fee for exclusive sponsorships. 7. The system communicates their answers to the LeagueOwner. 8. If there are interested sponsors, the LeagueOwner selects one. 9. The system records the name of the exclusive sponsor and charges the flat fee for sponsorships to the Advertiser's account. From now on, all advertisement banners associated with the tournament are provided by the exclusive sponsor. <p>....</p>

21

* Word highlighted in bold are potential classes

ARENA: Identifying Entity Object



- Try to apply the “Rules of Class Filtering”, find out the real classes

LeagueOwner	Tournament	Number of tournaments
League	Arena	Name
Application start and end dates	Start and end dates	Maximum number of Players
List of Advertisers	Exclusive sponsors	names
Proposed sponsors	Flat fee	Answers
Name	Advertiser's account	Advertisement banner

Potential Classes

22

ARENA: The Real Entity Objects



Account	• Balance, charge history (?), payment history (?)
Advertiser	• Name, account, sponsored tournament
Advertisement	• Associated game (?)
Arena	• max. number of tournaments, flat fee
League	• max. number of tournaments
LeagueOwner	• name
Player	• name
Tournament	• name, application start date, application end date

23

Identifying Boundary Object



- Boundary objects are NOT explicitly described in use case.
- How do you find out those hidden boundary objects? Ask **HOW** questions about interface and interaction between actors and the system
- Consider our classic example about Online Shopping system, ask:
 - How can customer places a payment order for product purchase?
 - How can customer view his/her order history?
 - How can the accounting system interface with this system?

24

Identifying Boundary Objects



- Examples of Boundary Objects for **AnnounceTournament** use case:
 - TournamentForm
 - RequestSponsorshipForm
 - SponsorshipRequest
 - SponsorshipResponse
- Examples of Boundary Objects for **BuyProduct** use case:
 - PaymentForm
 - ViewOrderHistoryForm

25

Refer to Table. 5-7 of textbook for complete description

Identifying Control Objects



- Control objects coordinate activities among boundary and entity objects
- Usually (but not always) a single control object is designed for one particular use case

26

ARENA: Identifying Control Objects



- Consider **AnnounceTournament** use case
- We need an object to control activities such as
 - Sending and collecting notice to advertisers
 - Checking resource availability
- → **AnnounceTournamentControl** object

27

Review: Identifying entity objects



- Entity object: an application domain concept that has information stored in it
- Heuristics for identifying entity objects:
 - Terms that developers or users need to clarify in order to understand the use case
 - Recurring nouns in the use cases
 - Real-world entities that the system needs to track
 - Real-world activities that the system needs to track
 - Data sources or sinks

28



Review: Identifying boundary objects

- Boundary objects: represent the system interface with the actors
- In each use case, each actor interacts with at least one boundary object
- Collect information from the actor and translates it into a form that can be used by both entity and control objects
- Model the user interface at a course level → do not describe visual aspects of the user interface
- Heuristics for identifying boundary objects:
 - More on P. 183 of the textbook

29



Review: Identifying control objects

- Control objects: responsible for coordinating boundary and entity objects
- Collect information from the boundary objects and dispatch it to entity objects
- Heuristics for identifying control objects:
 - Identify one control object per use case
 - Identify one control object per actor in the use case
 - The life span of a control object should cover the use case or the extent of a user session

30

Dynamic Model

- Model the operations of the system
- Depicted by sequence diagram and statecharts

Recap:

1. Sequence diagram represents the interaction among a set of objects, mostly within a single use case.
2. Statechart represents the transition of states of a single object.

31



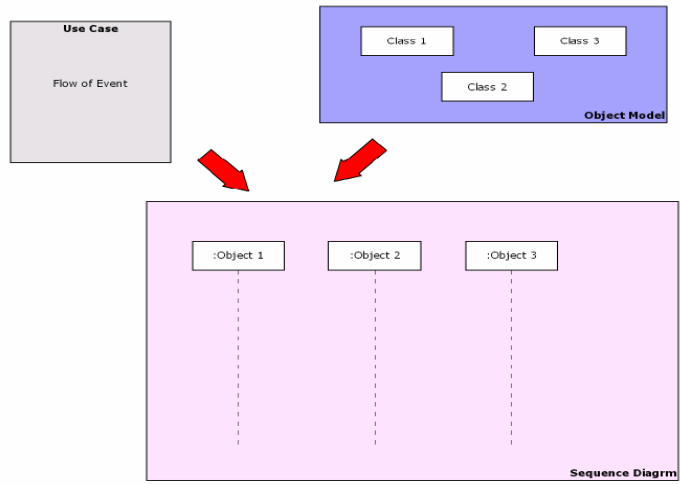
Working out the Model

- How do we produce the dynamic model? How do we get start?
 - Start with “Flow of events” of use cases and scenarios
 - Review the classes you developed in the Object Model
 - Find out the interactions between objects → Sequence Diagram
 - Model the dynamic behaviour of a single object → Statechart

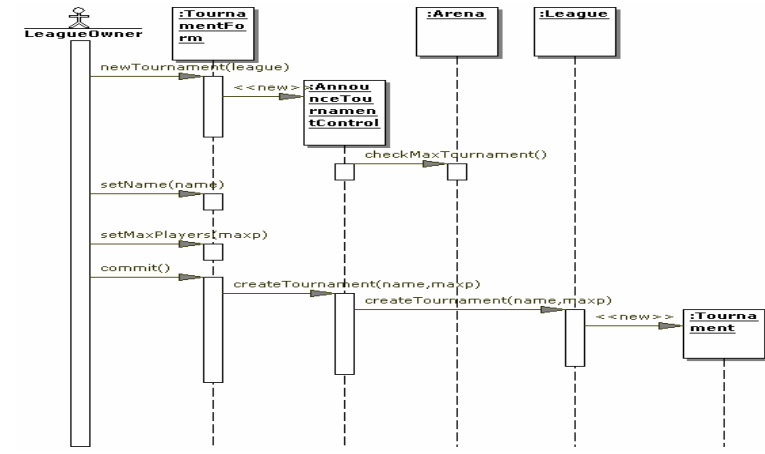
32



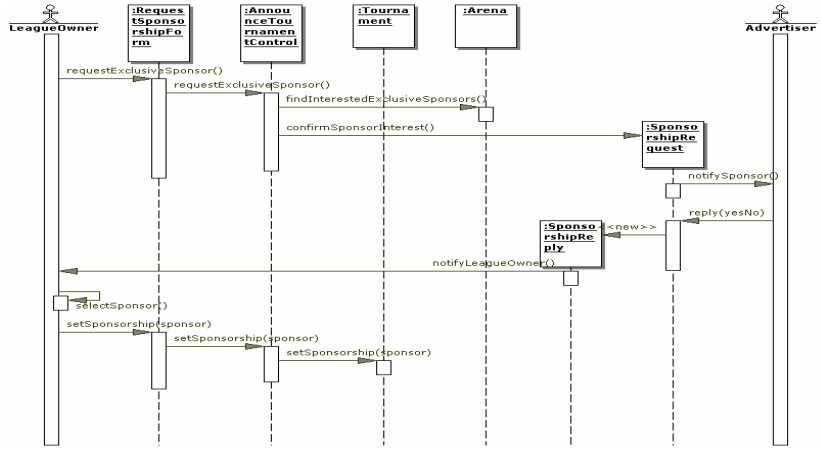
Working out the Sequence Diagram



ARENA: AnnouceTournament (Tournament Creation)



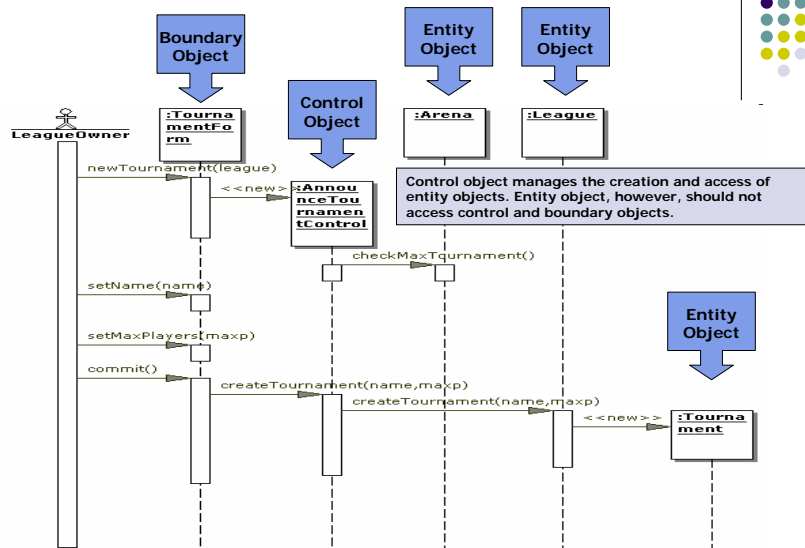
ARENA: AnnouceTournament (Sponsorship Workflow)



Guideline for drawing Sequence Diagram

- Object Layout
 - The first column should correspond to the actor who initiated the use case
 - The second column should be a boundary object (that the actor used to initiate the use case)
 - The third column should be the control object that manages the rest of the use cases





37

Requirement Analysis & Dynamic Model

- How does dynamic model aid your requirement process?
 - Help you generate questions on some uncertain or ambiguous requirements
 - Let developers have a high-level view of how the future system works

38

The Deliverable

- The deliverable of both requirement elicitation and requirement analysis processes is a **Requirement Analysis Document (RAD)**
- RAD documents all the information you gather and develop in the requirement process including:
 - Problem statement
 - Function/Non-functional Requirement
 - Use cases/Scenarios
 - Object Model/Dynamic Model
 - User Interface – mock up screen

39

Sample RAD Outline

1. Introduction
 - 1.1 Purpose of the system
 - 1.2 Scope of the system
 - 1.3 Project Objectives
 - 1.4 Definition, acronyms and abbreviation
 - 1.5 Overview
2. Proposed System
 - 2.1 Overview
 - 2.2 Functional Requirements
 - 2.3 Non-functional Requirements
 - 2.3.1 Usability
 - 2.3.2 Reliability
 - 2.3.3 Performance
 - 2.3.4 Supportability
3. System Models
 - 3.1 Scenarios
 - 3.2 Use case model
 - 3.3 Object model
 - 3.4 Dynamic model
 - 3.5 User interface – mockup screen

40