



MIT

E-Commerce Architecture Project

by Daniel Greenwood

UML and Use Cases

for

**Object-oriented
Analysis**



Topics of Discussion

- OOA
- UML
- Use Cases & Business Transaction Scenarios
- Use Case Models



Object-oriented Analysis

“Object-oriented Analysis (OOA) is a method of analysis that examines requirements from the perspective of the classes and objects found in the vocabulary of the problem domain”

- Grady Booch



Object-oriented Analysis

- Analysis Model provides the foundation for the Design Model
- Focus on Hi-level Business Objects
- Concentrate on activities of the User of the business process
- Avoid detailed design tasks



Requirements Analysis

- **Who** are the Users and the Customers?
- **Why** do they want this system?
- Define **what** the business needs to accomplish
- Define Constraints on **how** a solution is manifested but not on **how** system it is designed
- *What is accomplished conceptually*
- *What is required to interface to the system*
- *What is required to operate it*



Enterprise-wide Vs Project-Specific

- Enterprise-wide requirements provide Re-Use
- Requirements common to a project can be obtained by referring to enterprise-wide requirements
- Project-specific requirements should be evaluated for re-factoring into enterprise-wide requirements



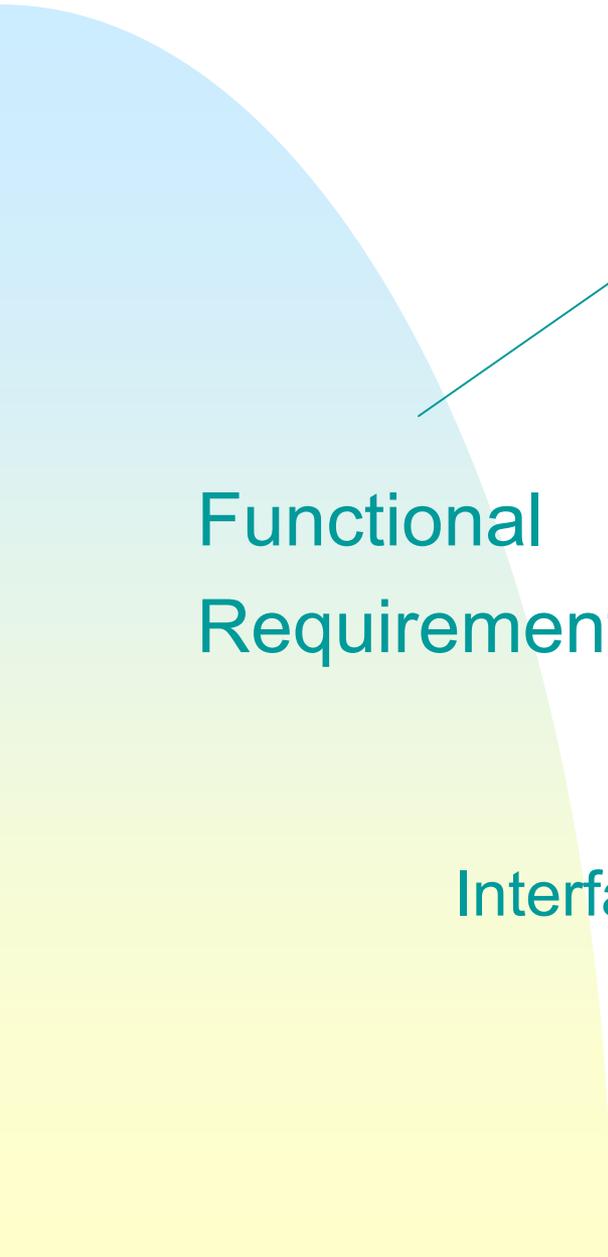
Requirements

Functional
Requirement

Non-Functional
Requirement

Interface Constraint

Operational Constraint





The Big Process Picture

- Requirements Analysis process fits into other processes within Integrated Requirements
- Deliverables output from one process become inputs to other processes
- Integrated Requirements provide the glue between the business side and the technology side



Essential Elements for Requirements Analysis

- Clarity
- Efficiency
- Priority
- Quality
- Traceability
- Completeness
- Accuracy



Guidelines for Requirements Analysis

- Problem Vs Solution Evolution
- Abstraction
- Iteration
- Modeling
- Re-Use



UML

- Unified Modeling Language
- Successor to methods of Booch, Rumbaugh & Jacobson
- A modeling language and not a method



The Unified Modeling Language (UML) is the industry-standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. It simplifies the complex process of software design, making a "blueprint" for construction. The UML definition was led by Rational Software's industry-leading methodologists: Grady Booch, Ivar Jacobson, and Jim Rumbaugh.



Use Cases

- A typical interaction a user has with a system to achieve a goal
- An essential tool in Requirements Capturing
- Provides User-visible function
- Use Cases are part of UML



Some Definitions

- Rectangles

Indicate a computer system boundary (Human are always outside “systems”)

- Ovals

Indicate a “use case” and straddle the boundary when a human Actor is involved, or can be inside a system when non-humans are interacting

- Actors

An actor is a role that an external object or user plays vis the System

- Arrows

Indicate activity or flow of information



Business Transaction Scenarios

- Business Transaction Scenarios describe all the possible interactions between the system and the external objects of the outside world. BTS are modeled as Use Cases
- Normal Scenario captures the normal interaction between the actor and the system
- Abnormal Scenario captures interaction that occurs during exceptions or error conditions



Sequence Diagrams

A Sequence Diagram provides a diagrammatic representation of a specific instance of a Use Case (a scenario)



Format of Use Cases

[Optional Information]

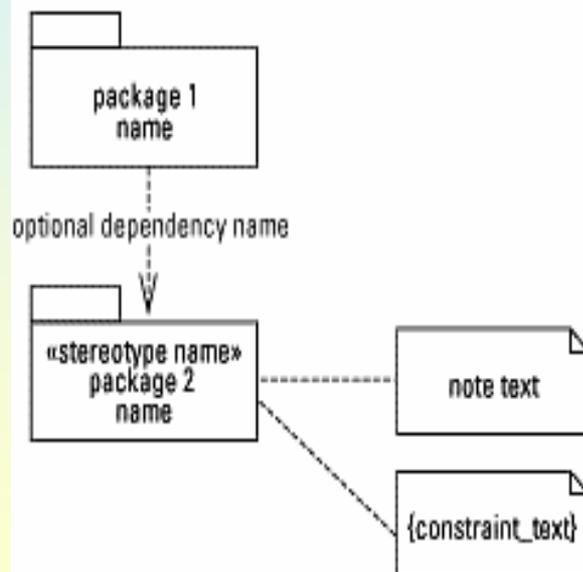
Scenarios and Use Cases will have the following sections in this order:

- .Purpose
- .Assumptions
- .Actors
- .Use Cases Used
- .Use Cases Extended
- .Preconditions
- .Postconditions
- .Basic Course
- .Alternate Course
- .Rules
- .Interface Constraints
- .Operational Constraints

GENERAL-PURPOSE CONCEPTS

Can be used on various diagram types

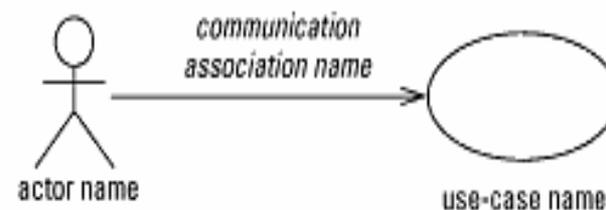
Package, dependency, note



USE-CASE DIAGRAM

Shows the system's use cases and which actors interact with them

Actor, use case, and association





Business Transaction Scenario: Learning Administration System

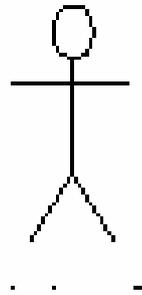
Draft 0.2



1. Scenario: Learning Administration System

The Learning Administration System (LAS) depicts the scenario where a student enrolls for a Program or Courses at a Learning Institution, attends the courses scheduled and after completion of the same, applies for various job positions at different companies.

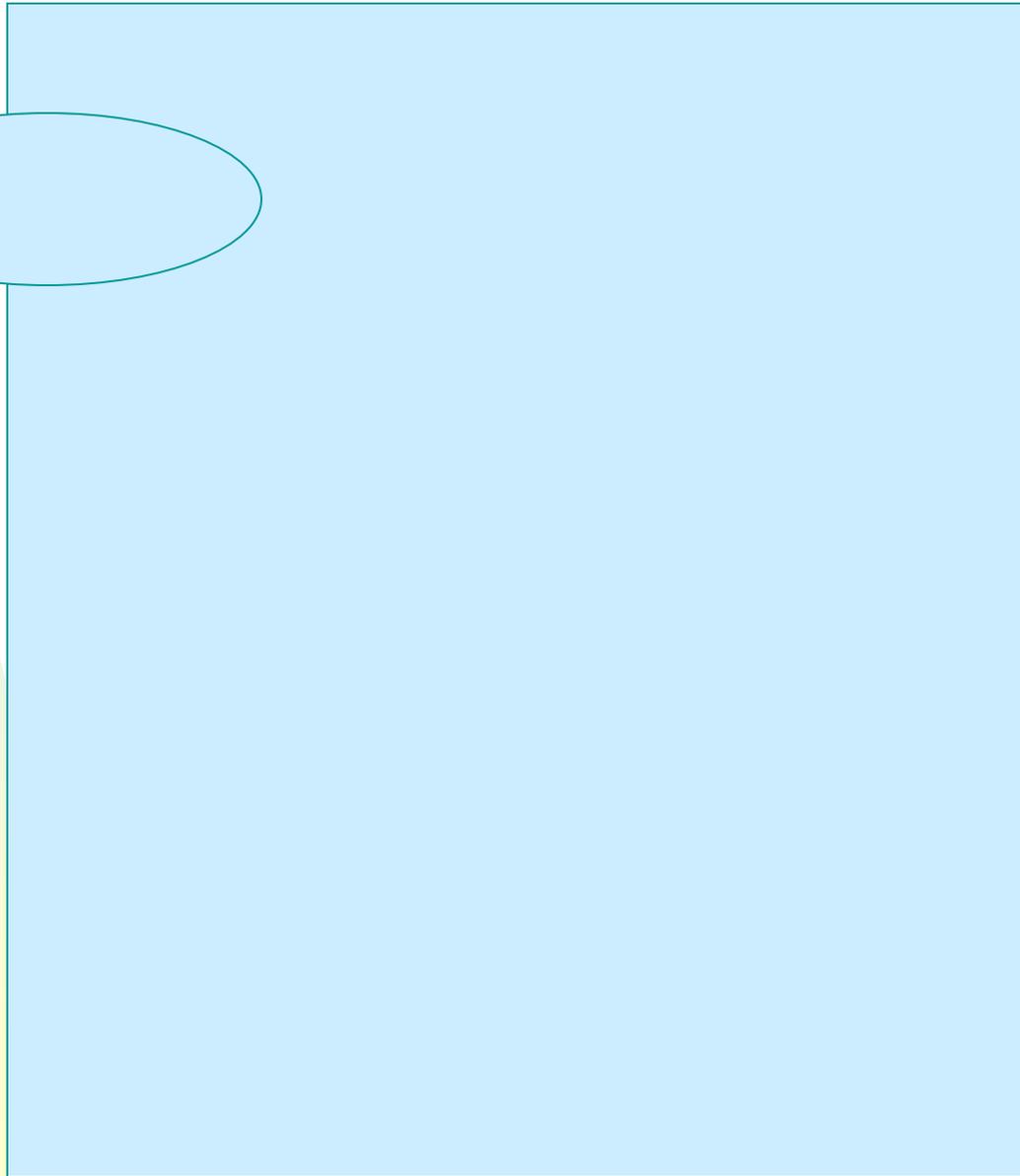
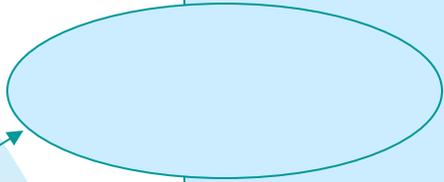
Who are the Actors?

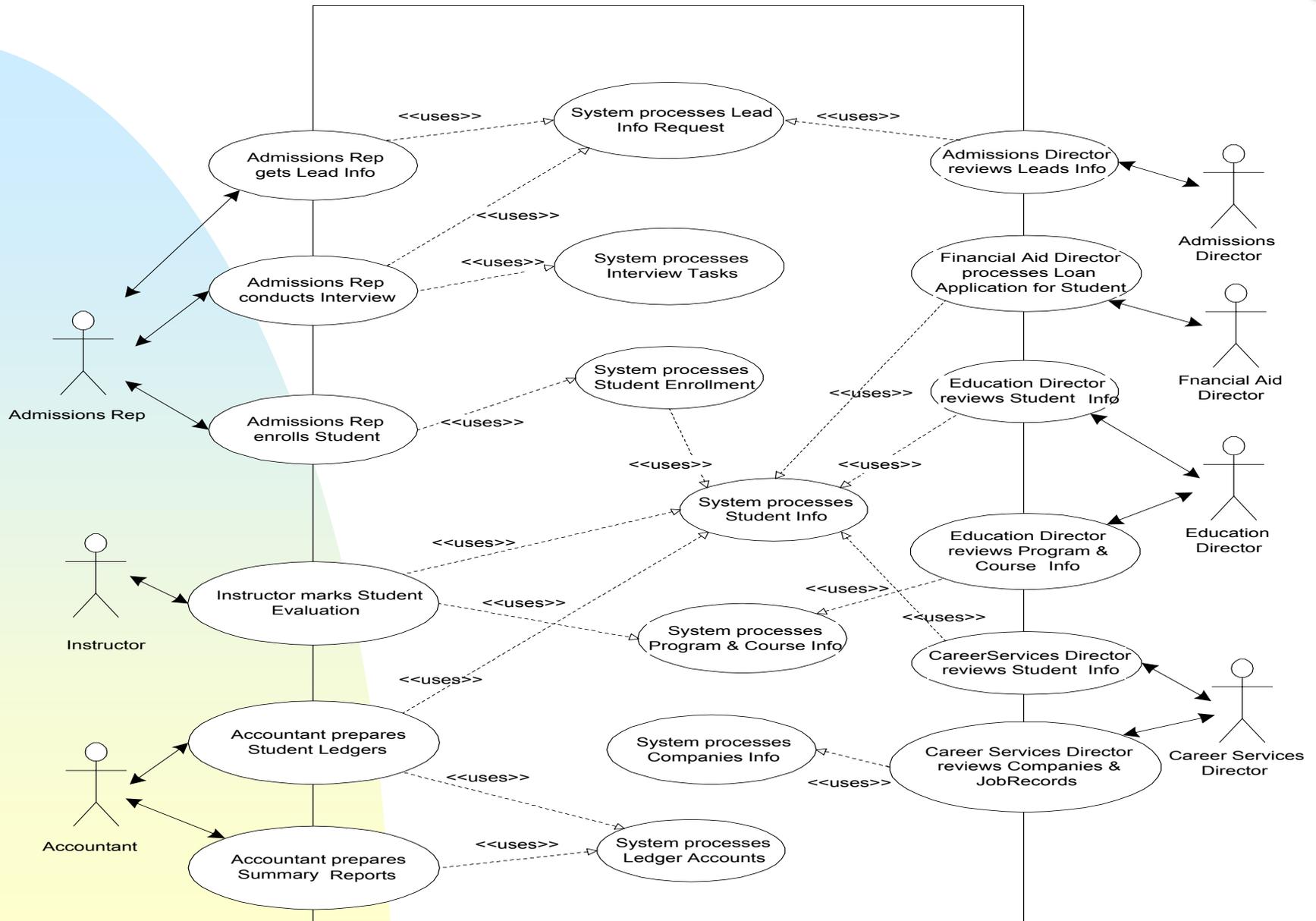


Admissions Rep
Admissions Director
Financial Aid Director
Education Director

Instructor
Career Services Director
Accountant

Let us model the system







Next Step ...

Let's get Hands-On

Thank You