Module 16

Course Review

Objectives

Upon completion of this module, you should be able to:

- Review the key features of object orientation
- Review the key UML diagrams
- Review the Requirements Analysis (Analysis) and Design workflows

Overview

During this module, your instructor will facilitate an interactive review in which you will recall and review the key topics covered in the course.

This is your opportunity to review the key topics covered in the course.

This course covered three main aspects that ran concurrently throughout the course:

- Object orientation
- UML diagrams
- The development process, focusing on the Analysis and Design workflows

Reviewing Object Orientation

Discussion – In this instructor-facilitated discussion, you will recall and review the key object-oriented (OO) concepts and terminology covered in the course.

This page is intentionally left blank

OO Concepts and Terminology: A Recap

The following are the key OO concepts and terminology that were covered in this course:

- Object

  Represents a runtime instance of a class.
- Class

  Represents a template for describing objects that share common data, structure, and behavior.
- Abstraction

  Represents a real world object with irrelevant details either hidden or removed.
- Encapsulation

  Represents an object's data and methods that are not directly accessible from outside of the object.
- Inheritance

  Represents a class that can be defined in terms of extending an existing class.
- Abstract class

  A class from which objects cannot be instantiated, often because the class is not fully defined in terms of its implementation.
- Interface

  Represents a classification of a set of methods that have no method implementation.
- Polymorphism

  Represents a mechanism where the runtime behavior is determined by the form of the object. However, there are other less object-oriented definitions of polymorphism.
- Cohesion

  Represents a relative measure of how well a class, component, or method supports a single purpose.
- Coupling

  Represents a relative measure of the dependence of one component on other components.

- Class associations and object links

  Represent the relationships between classes and objects respectively.
- Delegation

  Represents the delegation of the behavior of a method to other methods that are in the same class or in another class.
- Design pattern

  Represents a known solution to a common design problem. This solution can be customized to meet the specific requirements of the problem.

Reviewing UML Diagrams

Discussion – In this instructor-facilitated discussion, you will recall and review the key UML diagrams and their purposes.

This page is intentionally left blank

UML Diagrams: A Recap

The following are the key UML diagrams that were covered in this course:

- Use Case diagram

  Represents the set of high-level behaviors that the system must perform for a given actor.
- Class diagram

  Represents a collection of software classes and their interrelationships.
- Object diagram

  Represents a runtime snapshot of software objects and their interrelationships.
- Communication diagram (formerly Collaboration diagram)

  Represents a collection of objects that work together to support some system behavior.
- Sequence diagram

  Represents a time-oriented perspective of communication between objects.
- Activity diagram

  Represents a flow of activities that might be performed by either a system or an actor.
- State Machine diagram

  Represents the set of states that an object might experience and the triggers that transition the object from one state to another.
- Component diagram

  Represents a collection of physical software components and their interrelationships.
- Deployment diagram

  Represents a collection of components and shows how these are distributed across one or more hardware nodes.
- Package diagram

  Represents a collection of other modeling elements and diagrams.

The following UML diagrams were not formally covered in this course:

- Interaction Overview diagram

  Represents a form of activity diagram where nodes can represent interaction diagram fragments. These fragments are usually sequence diagram fragments, but can also be communication, timing, or interaction overview diagram fragments.
- Timing diagram

  Represents changes in state (state lifeline view) or value (value lifeline view). It can also show time and duration constraints and interactions between timed events.
- Composite Structure diagram

  Represents the internal structure of a classifier, usually in form of parts, and can include the interaction ports and interfaces (provided or required).
- Profile diagram

  Might define additional diagram types or extend existing diagrams with additional notations.

Figure 16-1 shows a pictorial representation of the 14 UML 2.2 diagrams.

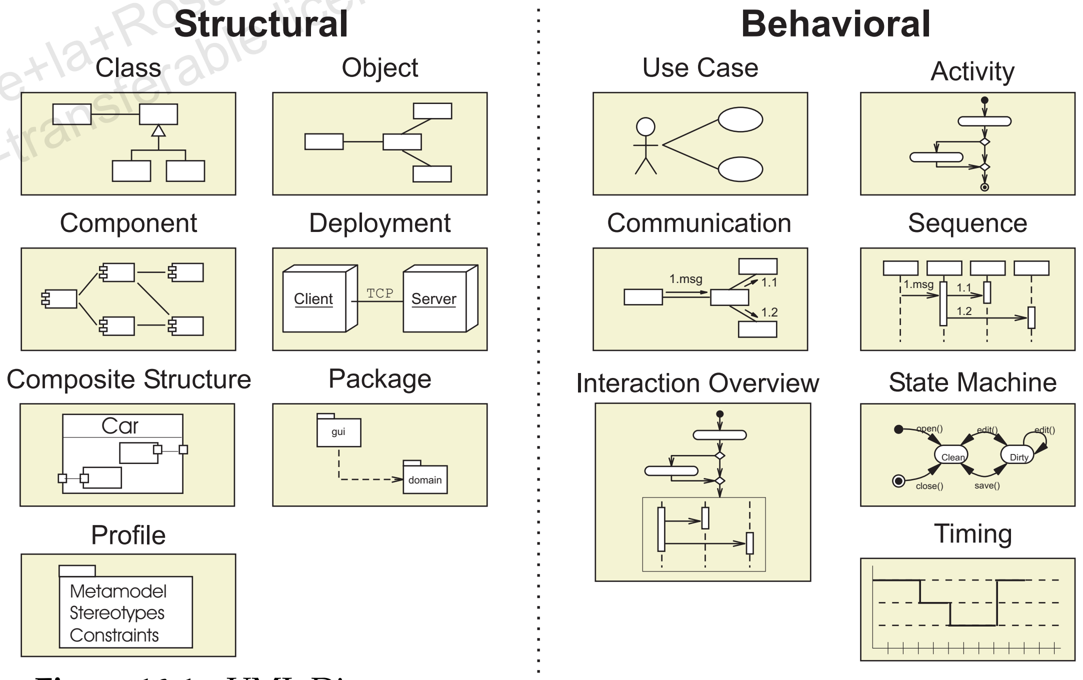

Figure 16-1 UML Diagrams

Reviewing the Development Process

Discussion – In this instructor-facilitated discussion, you will first recall and review the key workflows of the entire development process. Then, you will focus on reviewing the key activities and artifacts of the Analysis and Design workflows.

This page is intentionally left blank

The Development Process: A Recap

The following workflows were covered in the course:

- Requirements Gathering

  Determine the requirements of the system by meeting the business owner and users of the proposed system.
- Requirements Analysis (or just Analysis)

  Analyze, refine, and model the requirements of the system.
- Architecture

  Identify risk in the project and mitigate the risk by modeling the high-level structure of the system.
- Design

  Create a Solution model of the system that satisfies the system requirements.
- Implementation

  Build the software components defined in the Solution model.
- Testing

  Test the implementation against the expectations defined in the requirements.
- Deployment

  Deploy the implementation into the production environment.

Although the course discussed all of the above workflows, it focused mainly on the Analysis and Design workflows.

The Analysis and Design Workflows: A Recap

The following diagrams are just one interpretation of how to implement a development process.

Figure 16-2 shows a high-level view of one iteration covering the Analysis, Design and Architecture workflows. The Architecture workflow has been shown as a running concurrently alongside the Analysis and Design workflows.

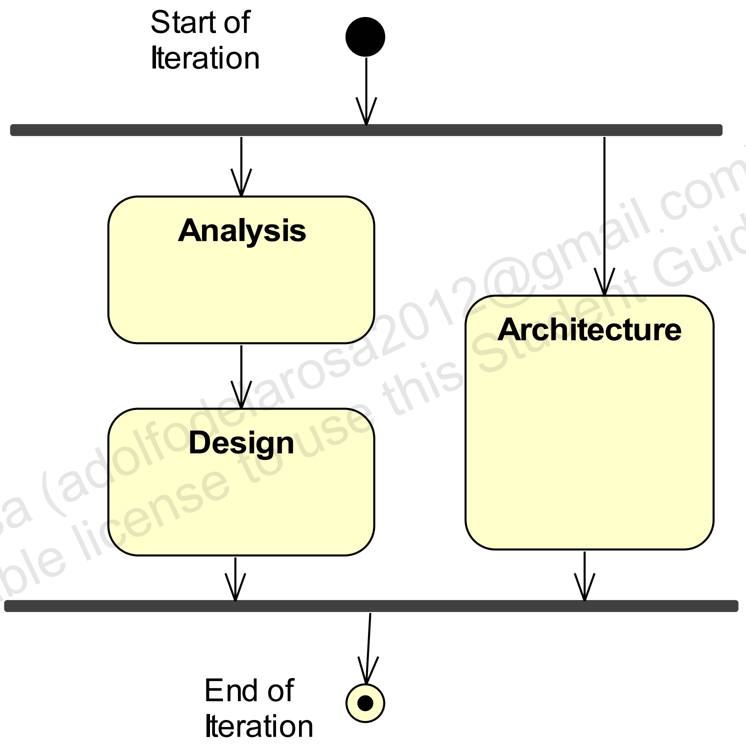

Figure 16-2 Example of a High-Level View of the Analysis, Design, and Architecture Workflows

Figure 16-3 shows an example of the activities and artifacts covered in one iteration of the Analysis workflow.

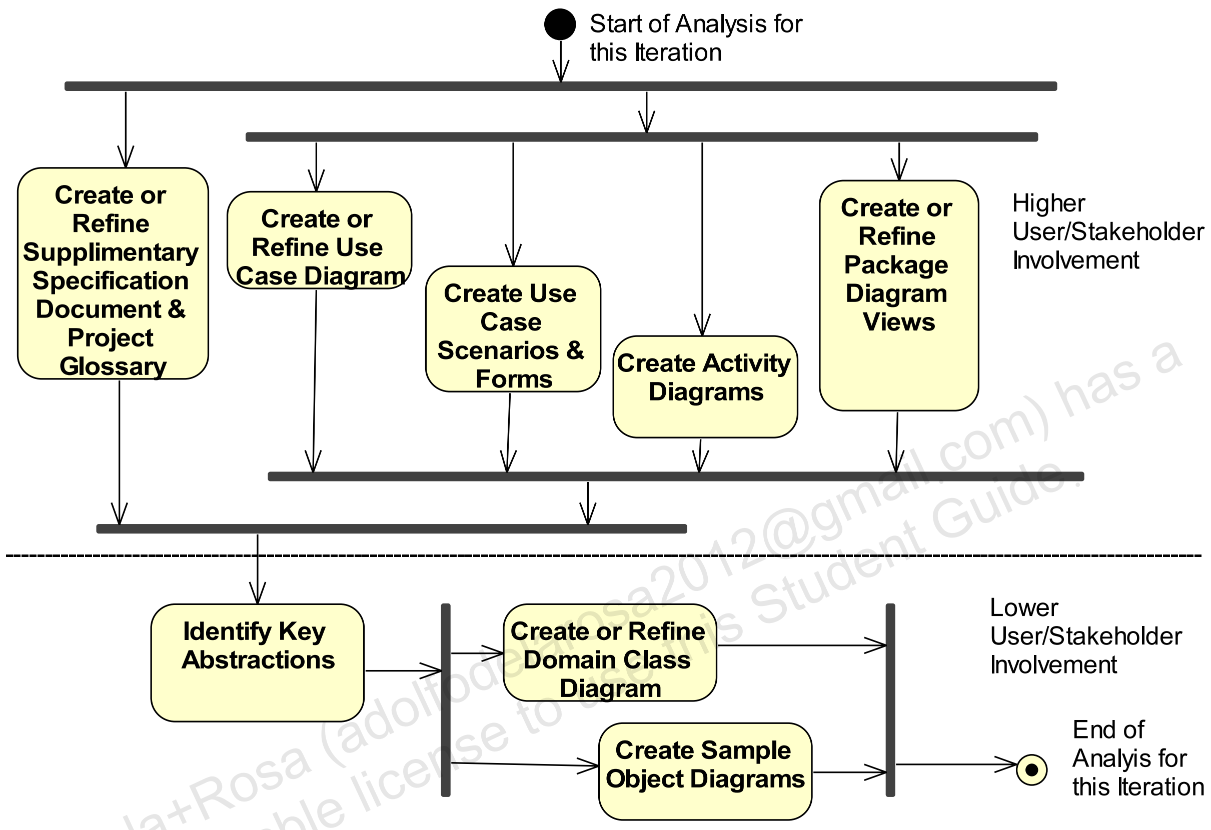
Start of Analysis for this Iteration
Create or Refine Supplimentary Specification Document & Project Glossary
Create or Refine Use Case Diagram
Create Use Case Scenarios & Forms
Create Activity Diagrams
Create or Refine Package Diagram Views
Higher User/Stakeholder Involvement
Identify Key Abstractions
Create or Refine Domain Class Diagram
Create Sample Object Diagrams
Lower User/Stakeholder Involvement
End of Analyis for this Iteration

Figure 16-3 Example of Activities in the Analysis Workflow

Figure 16-4 shows an example of the subactivities that occur during the Identify Key Abstraction activity shown in Figure 16-3 on page 16-15.

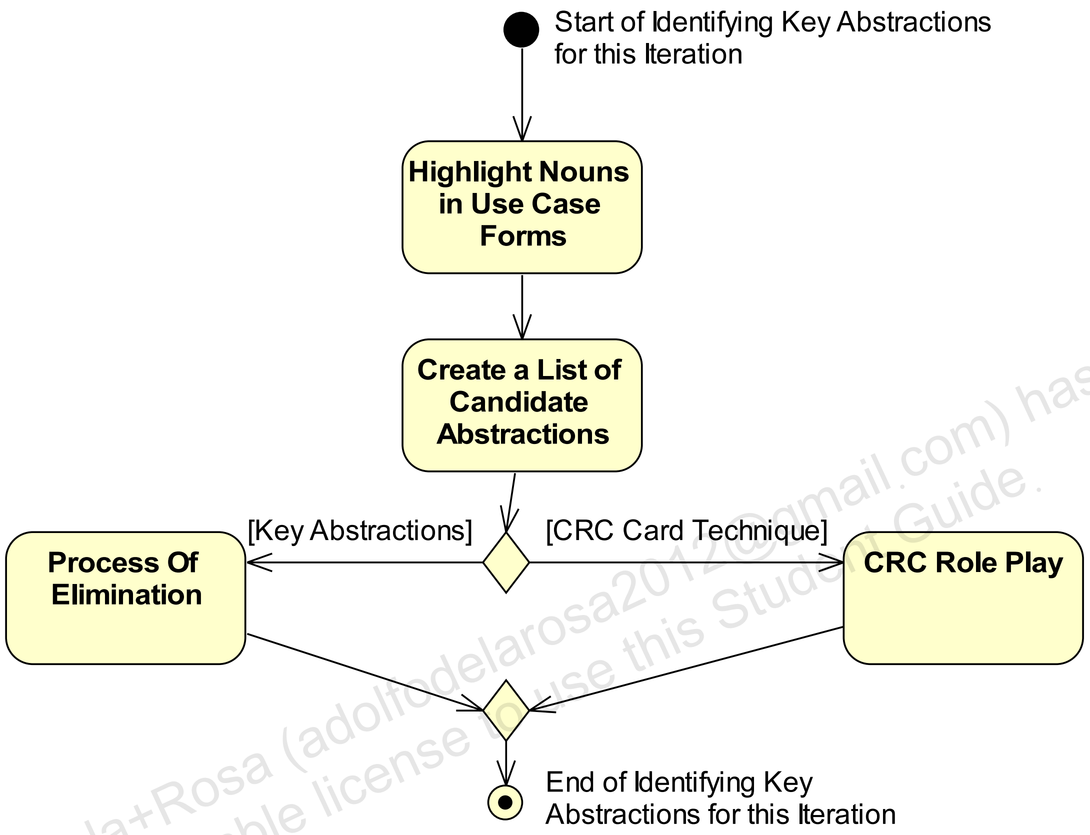
Start of Identifying Key Abstractions for this Iteration
Highlight Nouns in Use Case Forms
Create a List of Candidate Abstractions
[Key Abstractions]
[CRC Card Technique]
Process Of Elimination
CRC Role Play
End of Identifying Key Abstractions for this Iteration

Figure 16-4 Example of the Identify Key Abstractions Activity

Figure 16-5 shows an example of the activities and artifacts covered in one iteration of the Design workflow.

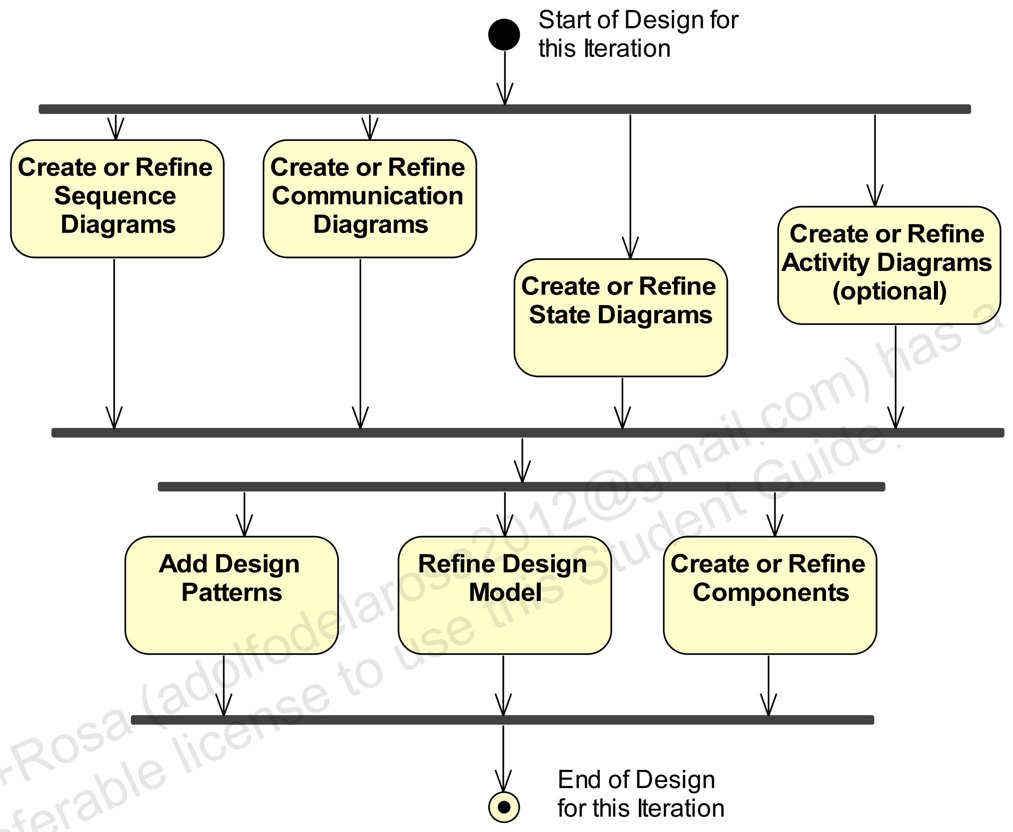
Start of Design for this Iteration
Create or Refine Sequence Diagrams
Create or Refine Communication Diagrams
Create or Refine State Diagrams
Create or Refine Activity Diagrams (optional)
Add Design Patterns
Refine Design Model
Create or Refine Components
End of Design for this Iteration

Figure 16-5 Example of Activities in the Design Workflow

Summary

In this module, you reviewed the three main aspects that ran concurrently throughout the course:

- Object orientation
- UML diagrams
- The development process, focusing on the Analysis and Design workflows