IBM Software Group

Mastering Object-Oriented Analysis and Design with UML

Module 2: Requirements Overview

Rational. software

Objectives: Requirements Overview

- Describe the basic Requirements concepts and how they affect Analysis and Design
- Demonstrate how to read and interpret the artifacts of Requirements that are used as a starting point for Analysis and Design

IBM

Requirements Overview Topics

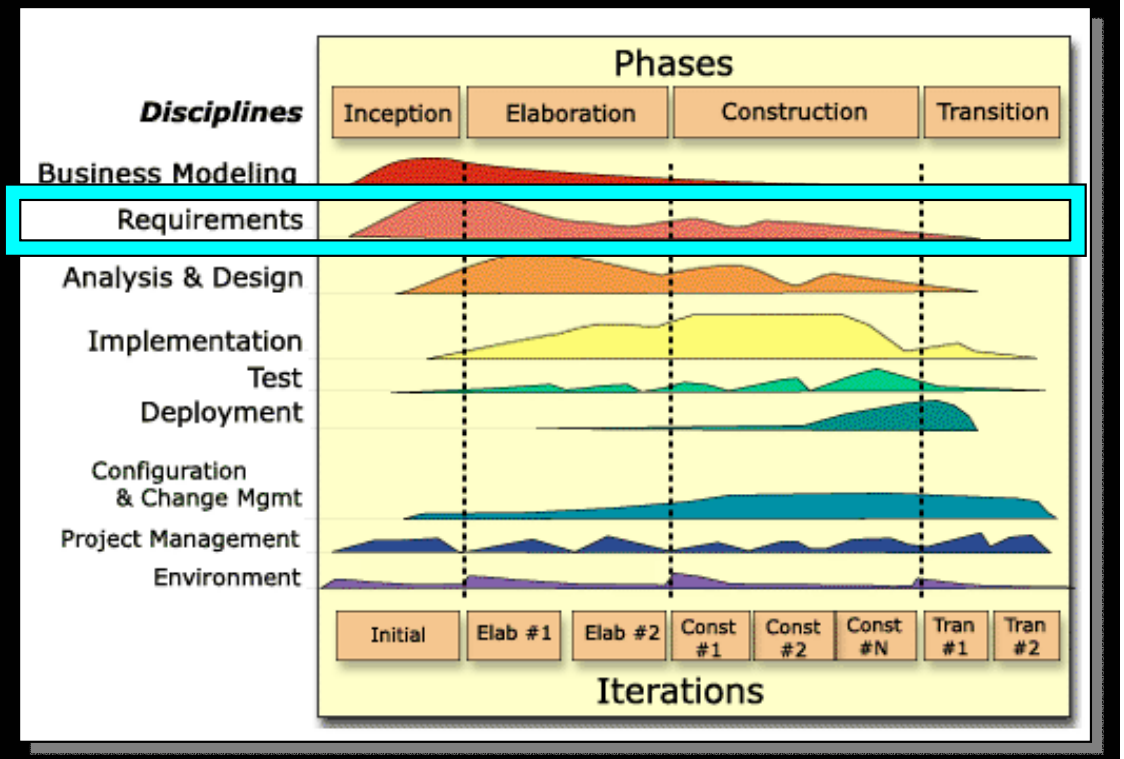
- ★ Introduction
- Key Concepts
- Use-Case Model
- Glossary
- Supplementary Specifications
- Checkpoints

IBM

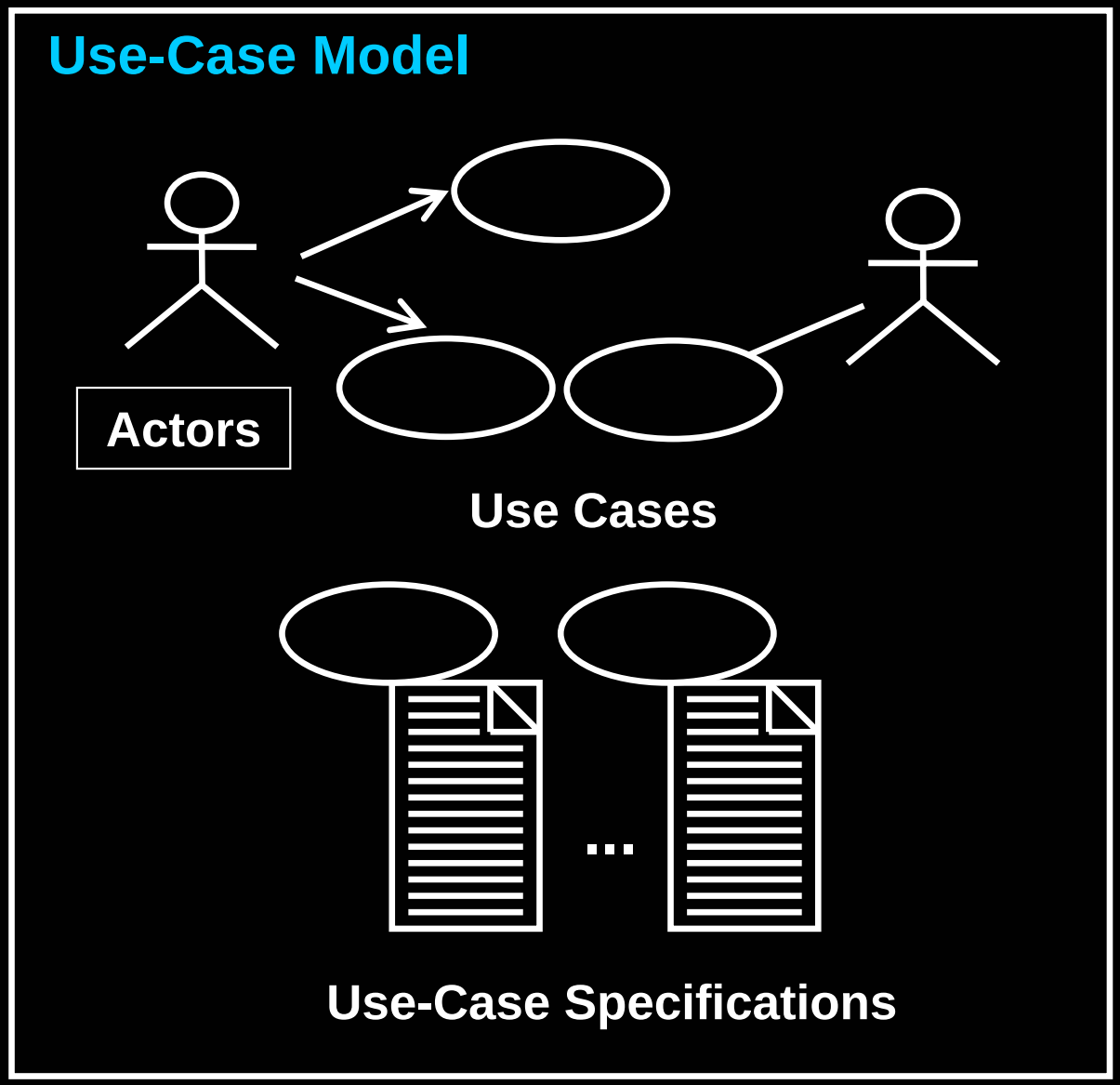
Requirements in Context

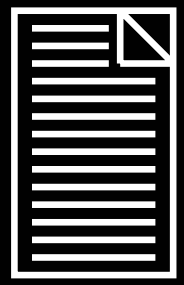
The purpose of Requirements is to:

- **Establish and maintain agreement with the customers and other stakeholders on what the system should do.**
- **Give system developers a better understanding of the requirements of the system.**
- **Delimit the system.**
- **Provide a basis for planning the technical contents of the iterations.**
- **Provide a basis for estimating cost and time to develop the system.**
- **Define a user interface of the system.**

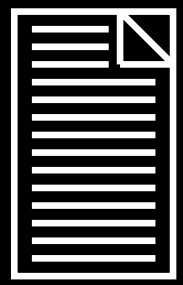


Relevant Requirements Artifacts





Glossary



Supplementary Specification

Case Study: Course Registration Problem Statement

- Review the problem statement provided in the Course Registration Requirements Document.

Course Registration
Requirements Document

IBM

Requirements Overview Topics

- Introduction
- ★ Key Concepts
- Use-Case Model
- Glossary
- Supplementary Specifications
- Checkpoints

IBM

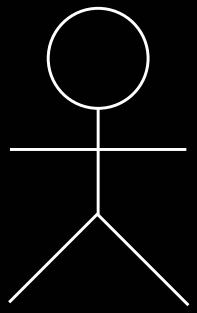
What Is System Behavior?

- System behavior is how a system acts and reacts.
  - It is the outwardly visible and testable activity of a system.
- System behavior is captured in use cases.
  - Use cases describe the system, its environment, and the relationship between the system and its environment.

IBM

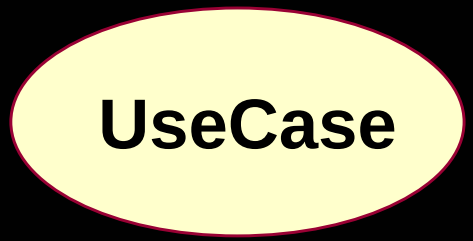
Major Concepts in Use-Case Modeling

- An actor represents anything that interacts with the system.



Actor

- A use case is a sequence of actions a system performs that yields an observable result of value to a particular actor.



IBM

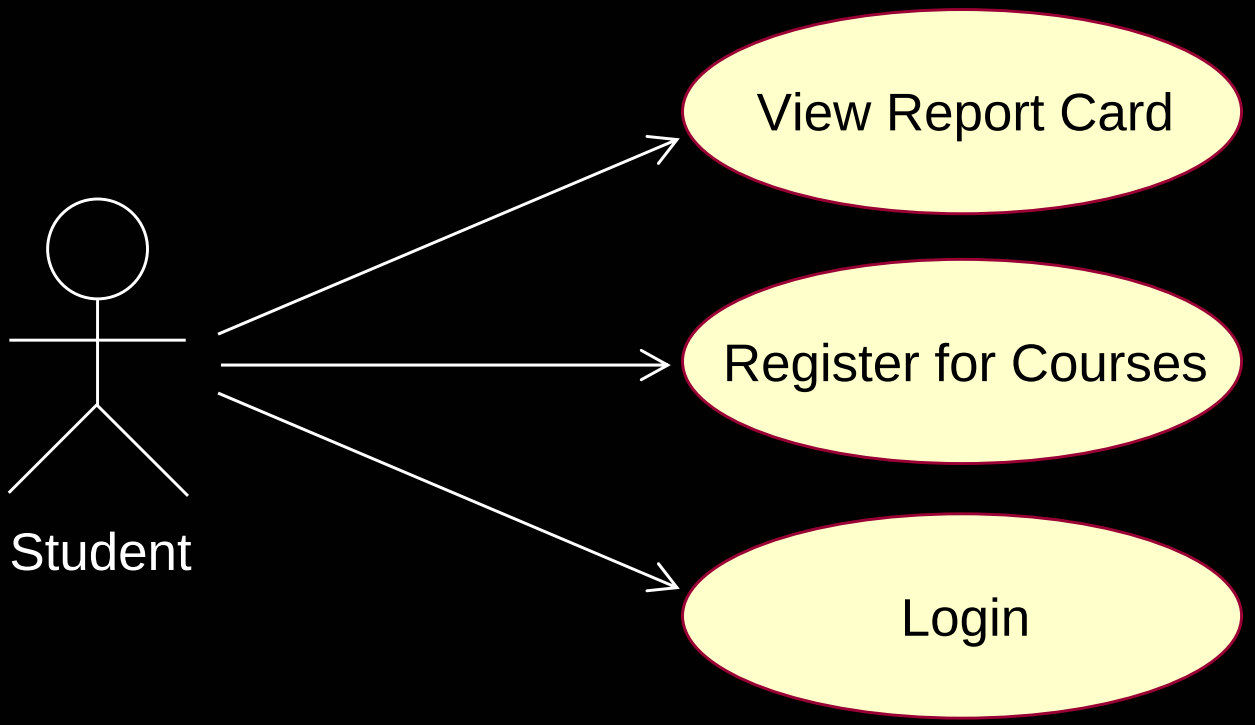
Requirements Overview Topics

- Introduction
- Key Concepts
- ★ Use-Case Model
- Glossary
- Supplementary Specifications
- Checkpoints

IBM

What Is a Use-Case Model?

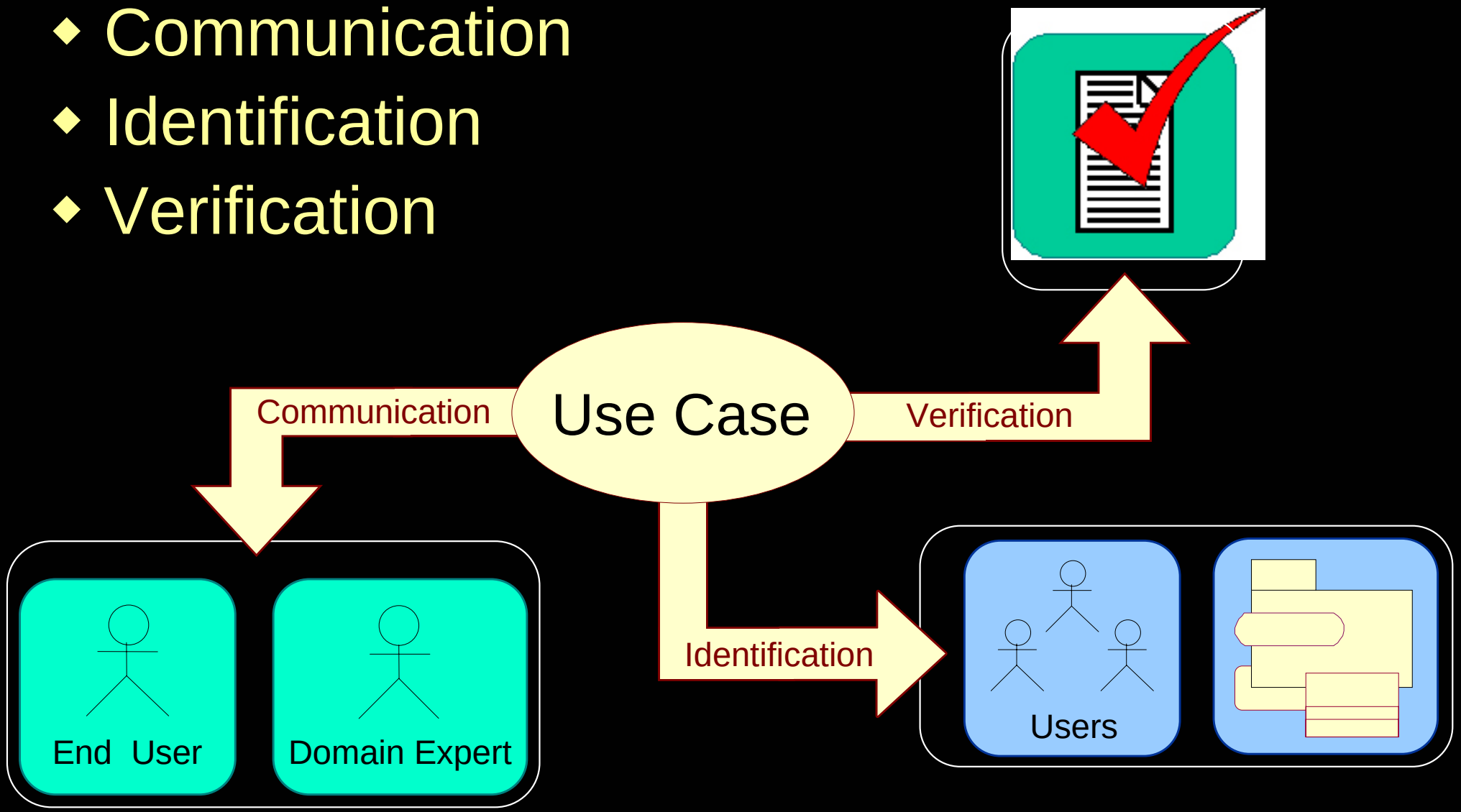
- A model that describes a system's functional requirements in terms of use cases
- A model of the system's intended functionality (use cases) and its environment (actors)



IBM

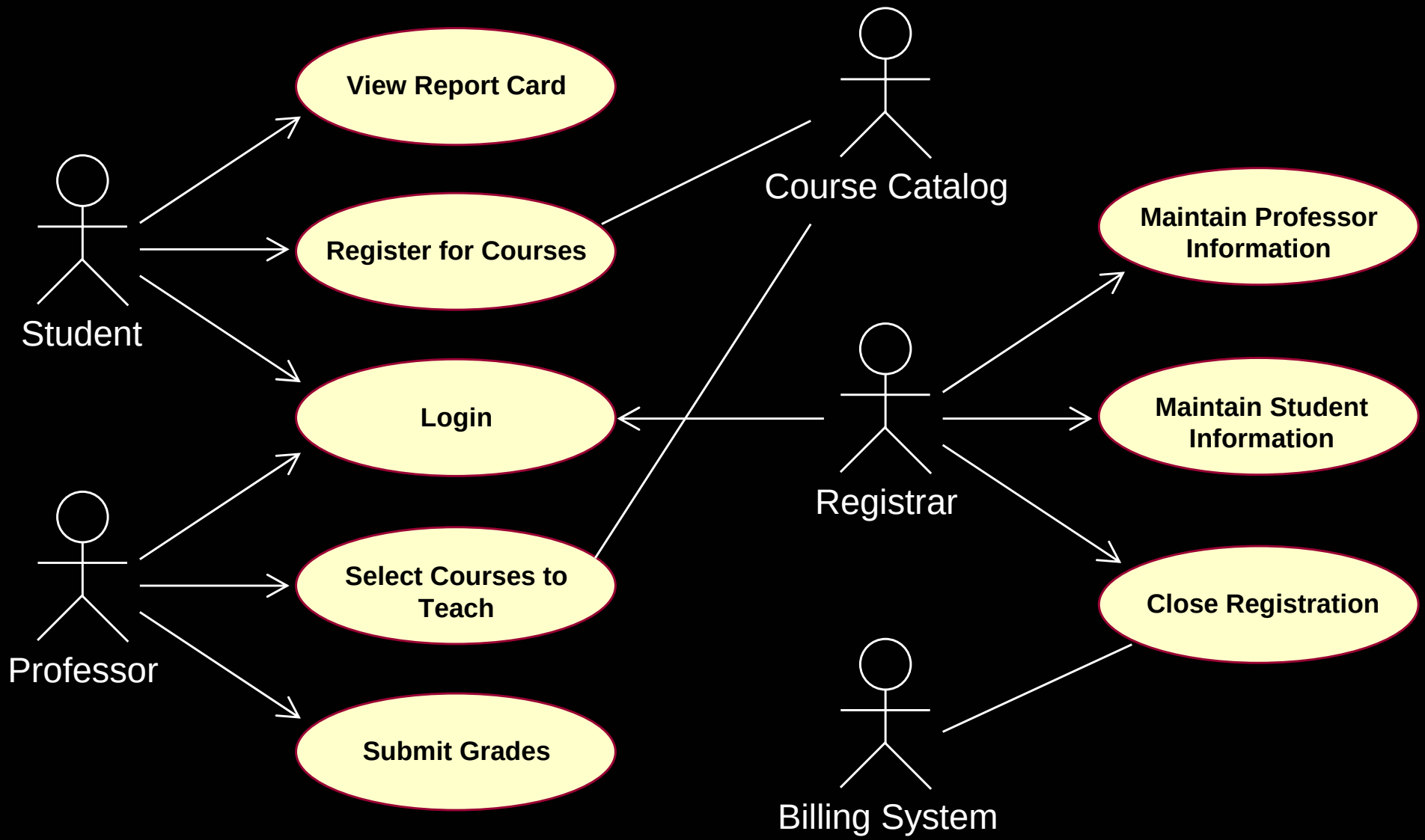
What Are the Benefits of a Use-Case Model?

- Communication
- Identification
- Verification



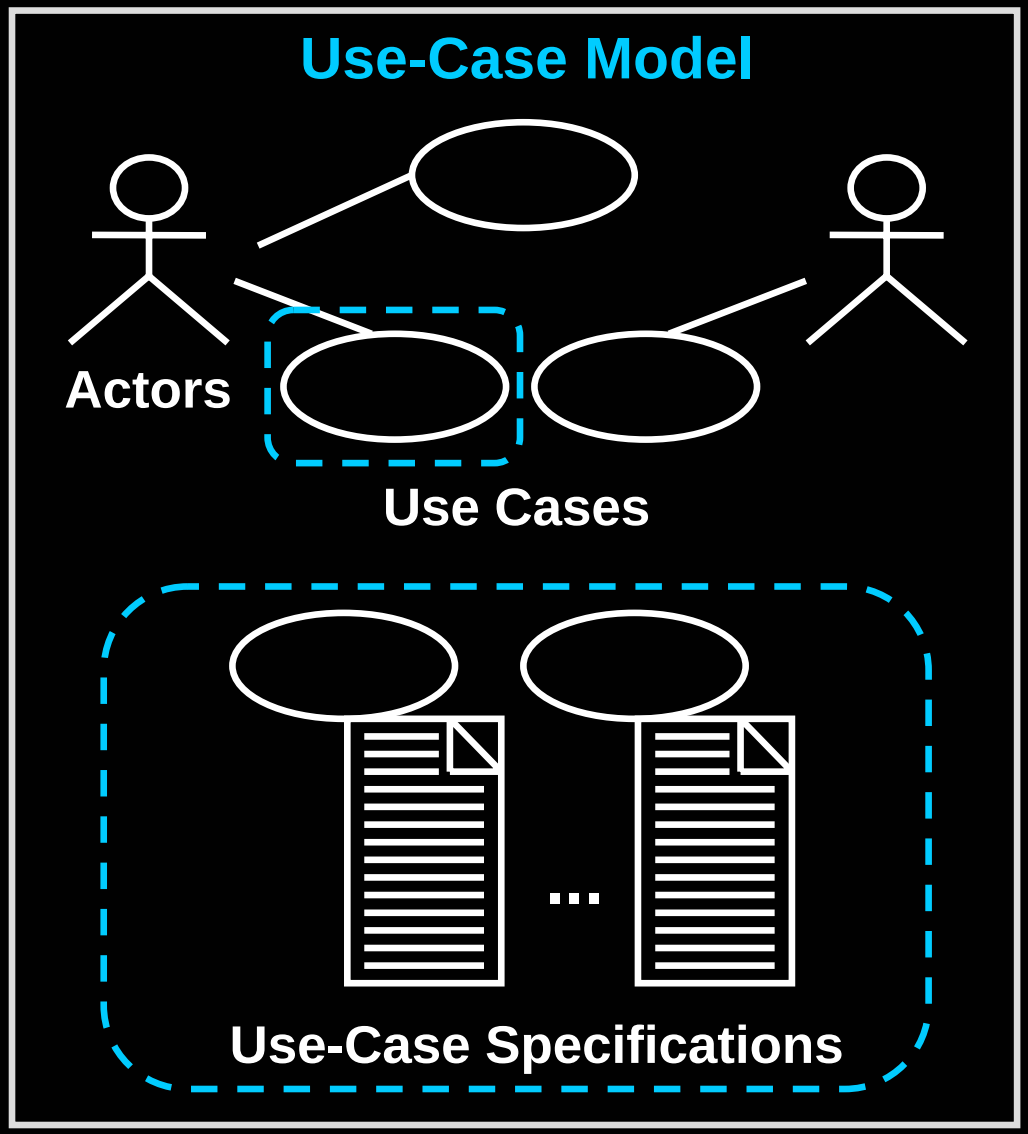
IBM

How Would You Read This Diagram?



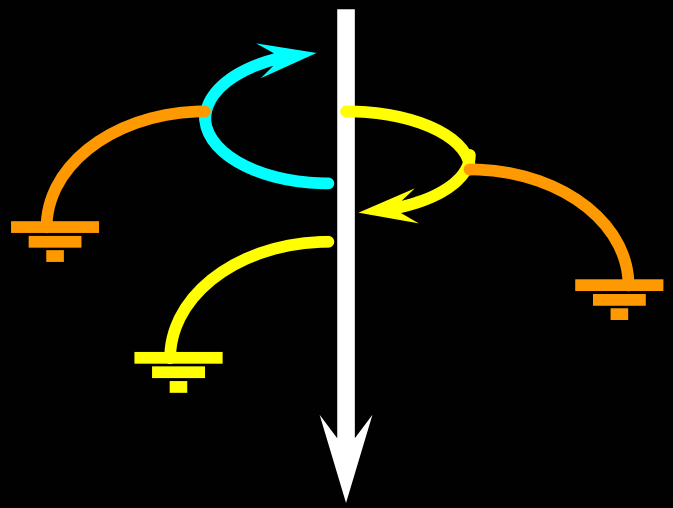
Use-Case Specifications

- Name
- Brief description
- Flow of Events
- Relationships
- Activity diagrams
- Use-Case diagrams
- Special requirements
- Pre-conditions
- Post-conditions
- Other diagrams



Use-Case Flow of Events

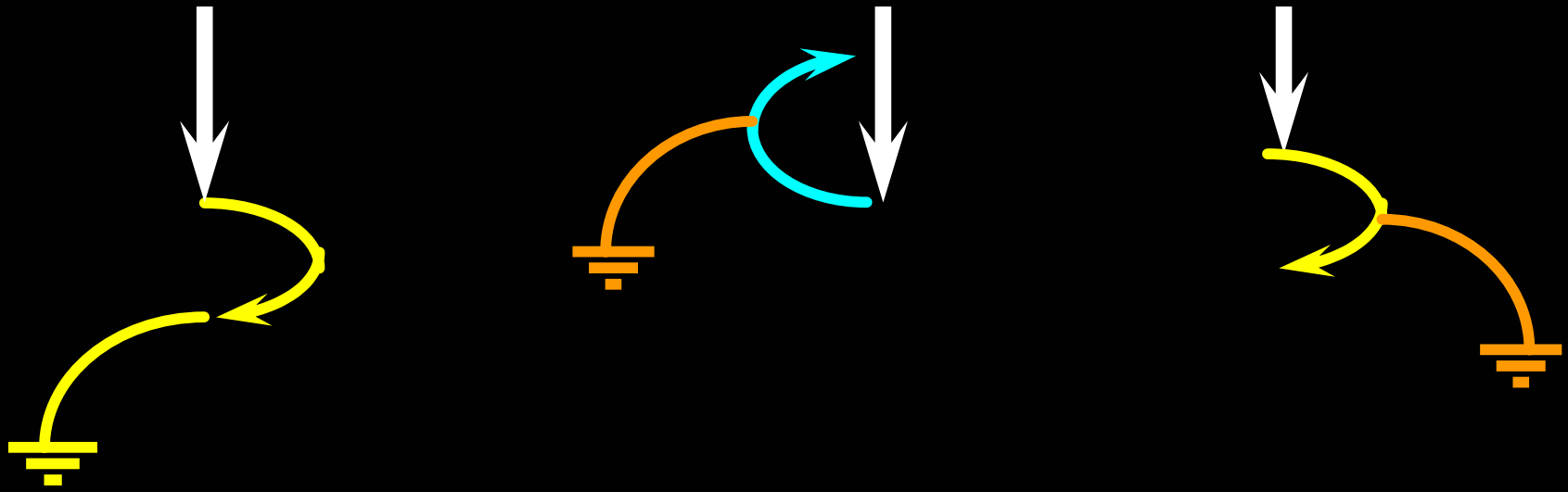
- Has one normal, *basic flow*
- Several *alternative flows*
  - Regular variants
  - Odd cases
  - Exceptional flows for handling error situations



IBM

What Is a Scenario?

- A scenario is an instance of a use case.



IBM

What Is an Activity Diagram?

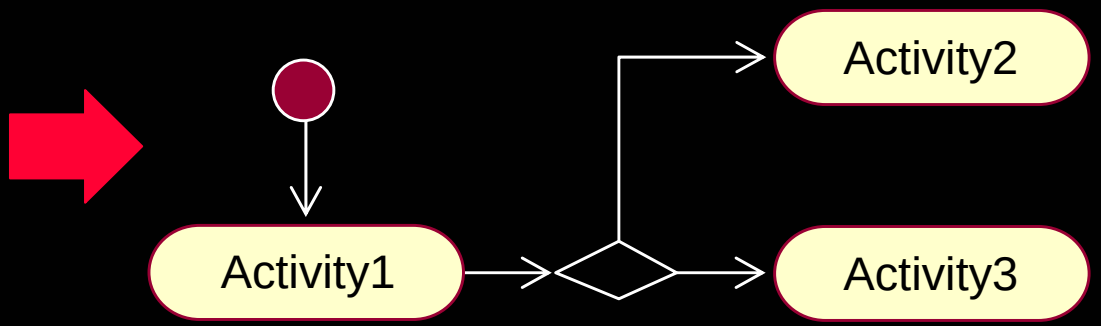
- An activity diagram in the Use-Case Model can be used to capture the activities in a use case.
- It is essentially a flow chart, showing flow of control from activity to activity.

Flow of Events

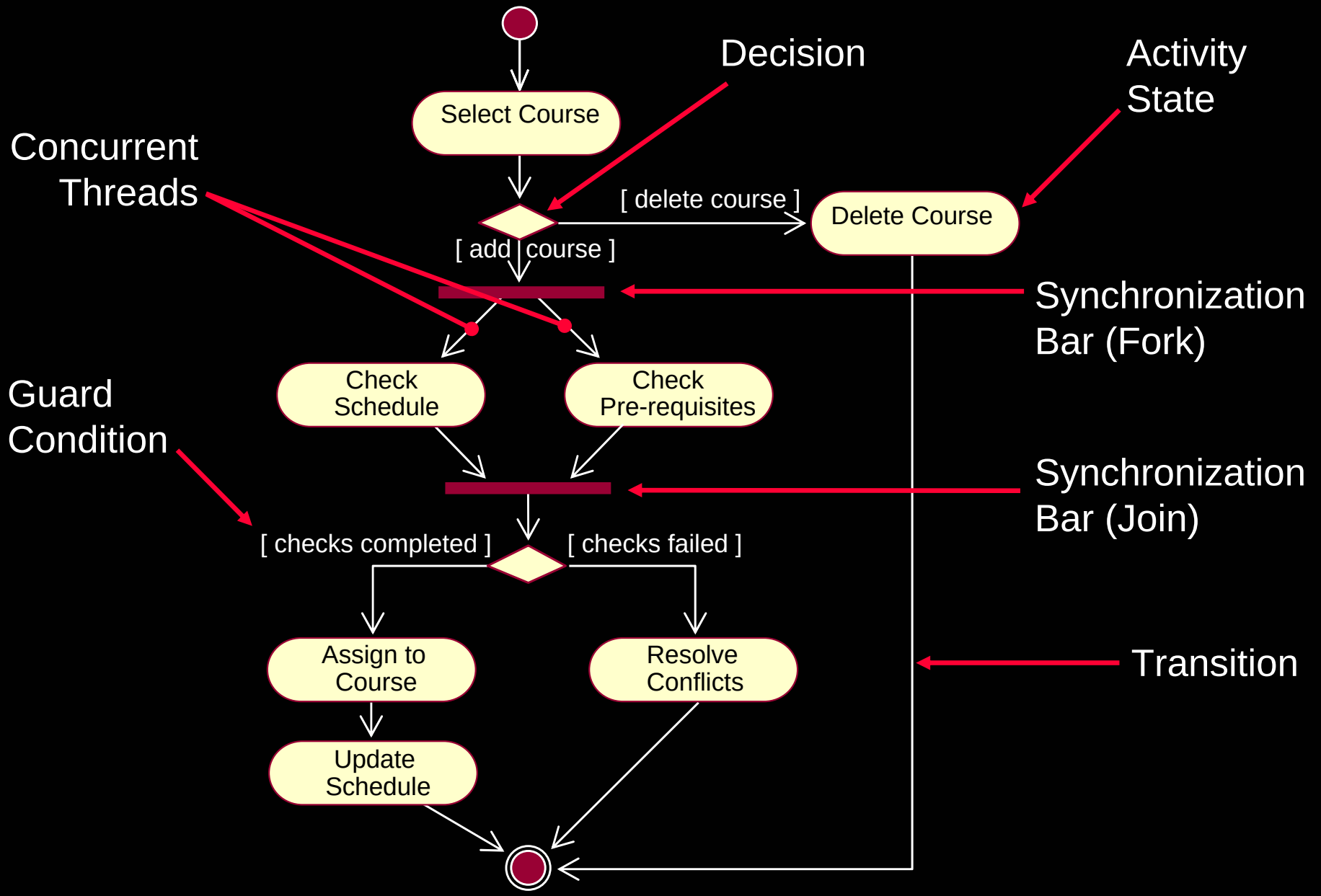
This use case starts when the Registrar requests that the system close registration.

1. The system checks to see if registration is in progress. If it is, then a message is displayed to the Registrar and the use case terminates. The Close Registration processing cannot be performed if registration is in progress.

2. For each course offering, the system checks if a professor has signed up to teach the course offering and at least three students have registered. If so, the system commits the course offering for each schedule that contains it.



Example: Activity Diagram

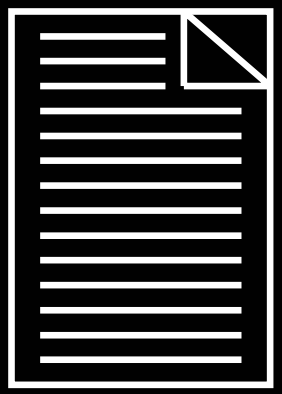


Requirements Overview Topics

- Introduction
- Key Concepts
- Use-Case Model
- ★ Glossary
- Supplementary Specifications
- Checkpoints

IBM

Glossary



Glossary

Course Registration System Glossary

1. Introduction

This document is used to define terminology specific to the problem domain, explaining terms, which may be unfamiliar to the reader of the use-case descriptions or other project documents. Often, this document can be used as an informal *data dictionary*, capturing data definitions so that use-case descriptions and other project documents can focus on what the system must do with the information.

2. Definitions

The glossary contains the working definitions for the key concepts in the Course Registration System.

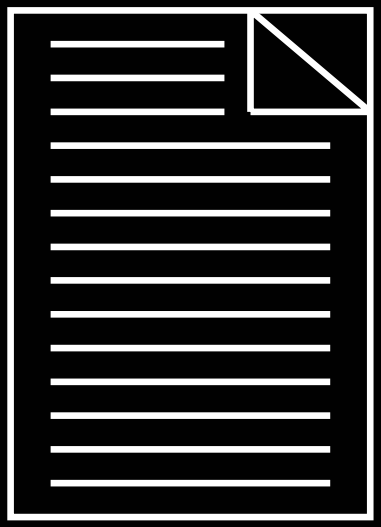
2.1 Course: A class offered by the university.

2.2 Course Offering: A specific delivery of the course for a specific semester – you could run the same course in parallel sessions in the semester. Includes the days of the week and times it is offered.

2.3 Course Catalog: The unabridged catalog of all courses offered by the university.

Case Study: Glossary

- Review the Glossary provided in the Course Registration Requirements Document



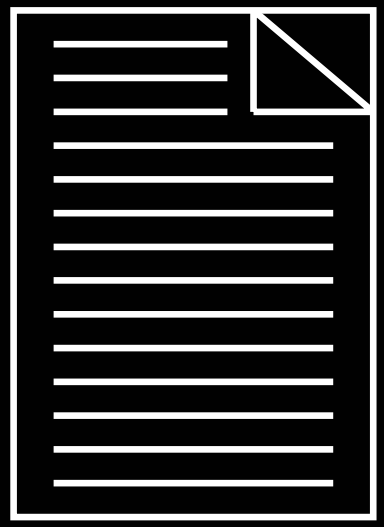
Glossary

Requirements Overview Topics

- Introduction
- Key Concepts
- Use-Case Model
- Glossary
- ★ Supplementary Specifications
- Checkpoints

Supplementary Specification

- Functionality
- Usability
- Reliability
- Performance
- Supportability
- Design constraints

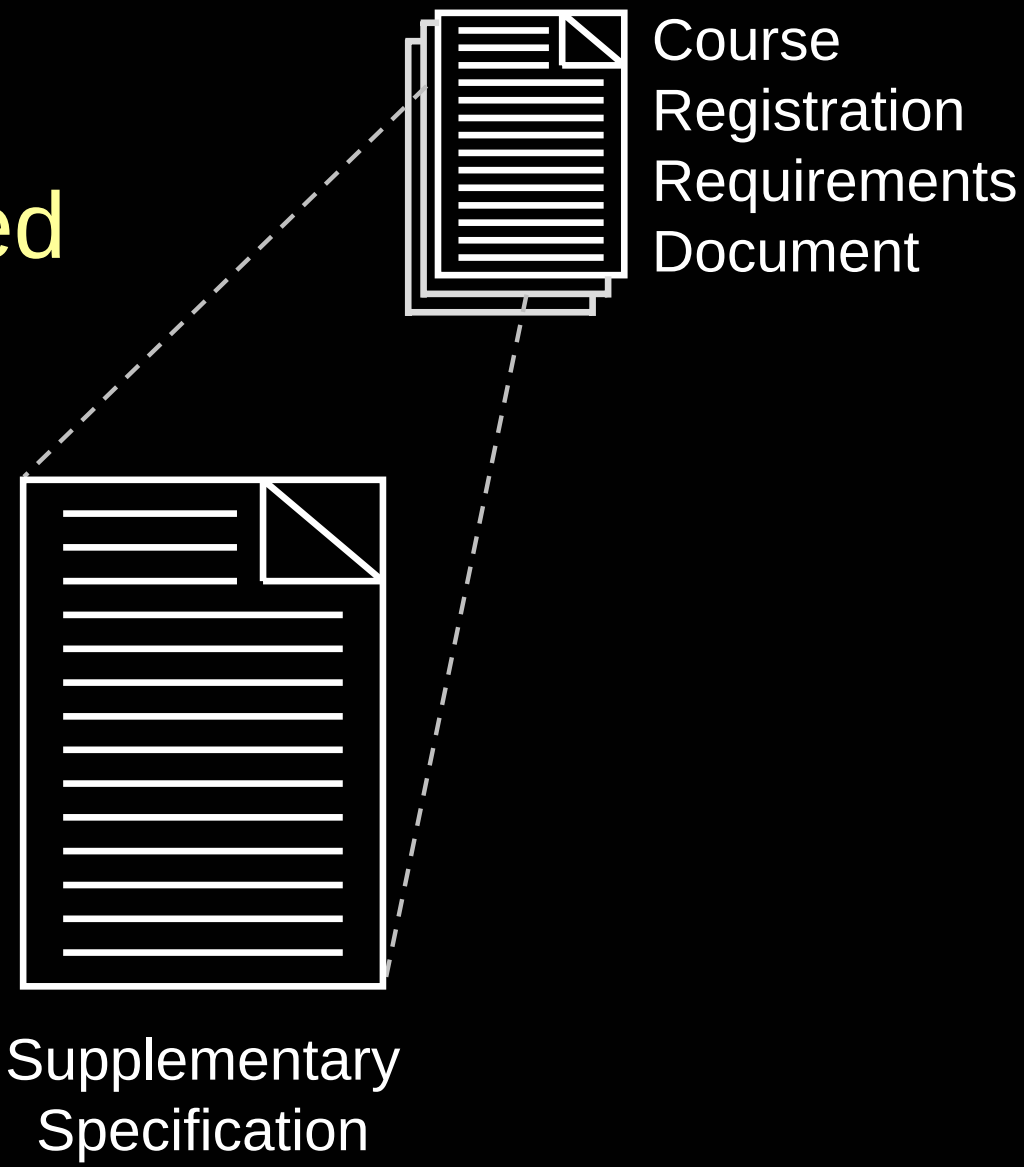


Supplementary Specification

IBM

Example: Supplementary Specification

- Review the Supplementary Specification provided in the Course Registration Requirements Document.



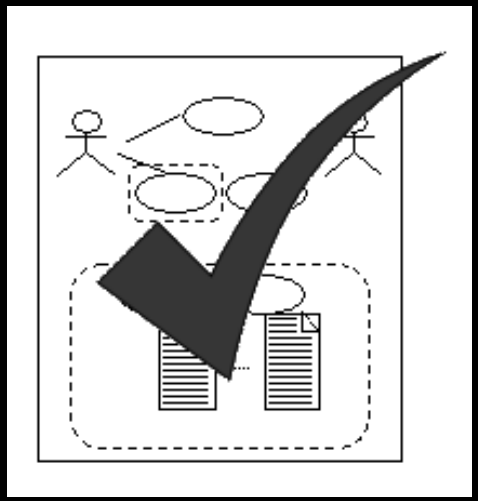
Requirements Overview Topics

- Introduction
- Key Concepts
- Use-Case Model
- Glossary
- Supplementary Specifications
- ★ Checkpoints

IBM

Checkpoints: Requirements: Use-Case Model

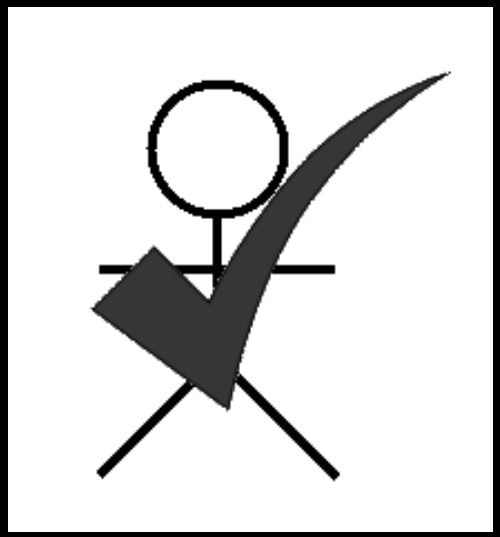
- Is the Use-Case Model understandable?
- By studying the Use-Case Model, can you form a clear idea of the system's functions and how they are related?
- Have all functional requirements been met?
- Does the Use-Case Model contain any superfluous behavior?
- Is the division of the model into use-case packages appropriate?



IBM

Checkpoints: Requirements: Actors

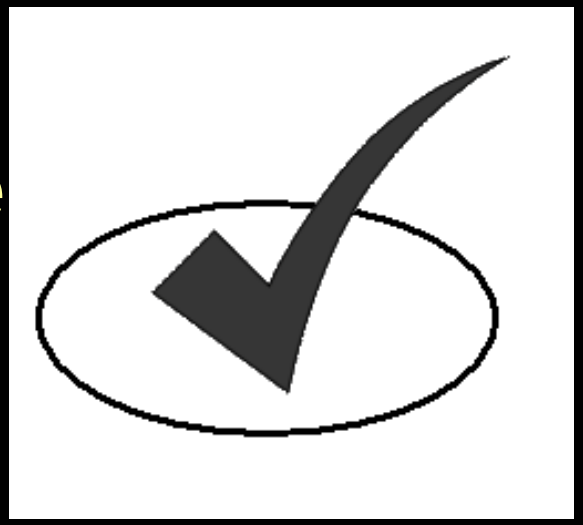
- Have all the actors been identified?
- Is each actor involved with at least one use case?
- Is each actor really a role? Should any be merged or split?
- Do two actors play the same role in relation to a use case?
- Do the actors have intuitive and descriptive names? Can both users and customers understand the names?



IBM

Checkpoints: Requirements: Use-Cases

- Is each use case involved with at least one actor?
- Is each use case independent of the others?
- Do any use cases have very similar behaviors or flows of events?
- Do the use cases have unique, intuitive, and explanatory names so that they cannot be mixed up at a later stage?
- Do customers and users alike understand the names and descriptions of the use cases?



IBM

Checkpoints: Requirements: Use-Case Specifications

- Is it clear who wants to perform a use case?
- Is the purpose of the use case also clear?
- Does the brief description give a true picture of the use case?
- Is it clear how and when the use case's flow of events starts and ends?
- Does the communication sequence between actor and use case conform to the user's expectations?
- Are the actor interactions and exchanged information clear?
- Are any use cases overly complex?

IBM

Checkpoints: Requirements: Glossary

- Does each term have a clear and concise definition?
- Is each glossary term included somewhere in the use-case descriptions?
- Are terms used consistently in the brief descriptions of actors and use cases?

IBM

Review: Requirements Overview

- What are the main artifacts of Requirements?
- What are the Requirements artifacts used for?
- What is a Use-Case Model?
- What is an actor?
- What is a use case? List examples of use case properties.
- What is the difference between a use case and a scenario?
- What is a Supplementary Specification and what does it include?
- What is a Glossary and what does it include?

IBM

Exercise: Requirements Overview

- Given the following Requirements artifacts:
  - Problem statement
  - Use-Case Model main diagram
  - Supplementary specification
  - Glossary
- Review the given Requirements artifacts, noting any questions, issues, inconsistencies.

IBM