IBM Software Group

Mastering OOAD: UML 1.x to 2.0 Migration
Module 2: Requirements Overview and Use-Case Analysis

Rational. software

Objectives: Requirements Overview and Use-Case Analysis

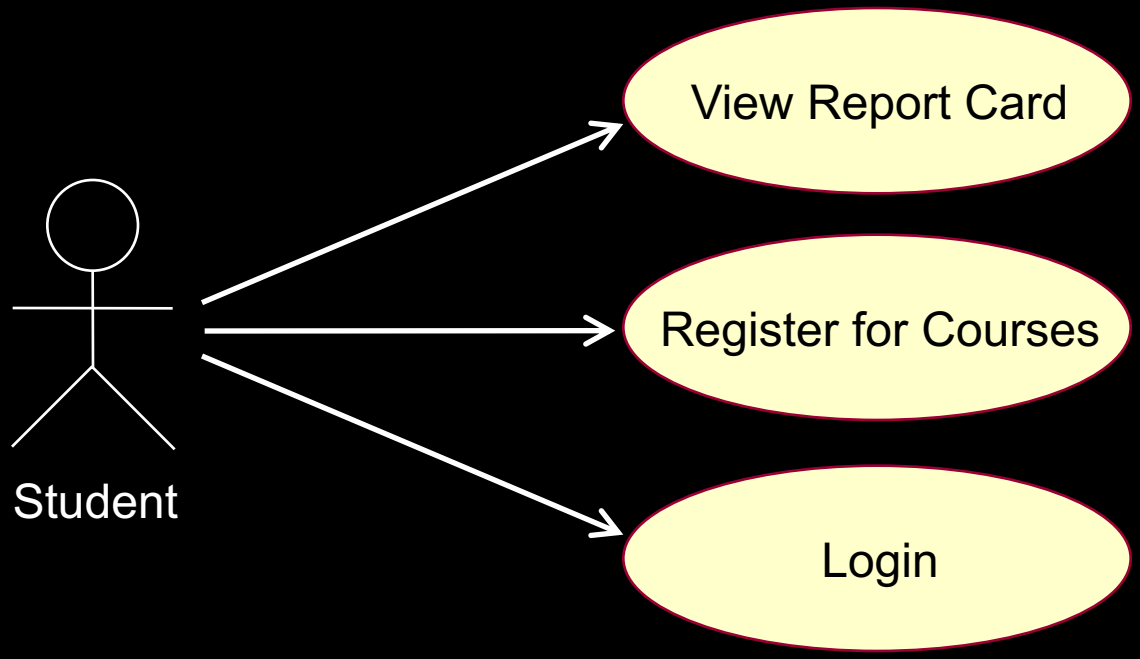
- Demonstrate how to read and interpret the Activity Diagram artifact of Requirements that is used as a starting point for Analysis and Design
- Distribute the use-case behavior to classes, identifying responsibilities of the classes
- Develop Use-Case Realizations that model the collaborations between instances of the identified classes

Requirements Overview Topics

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

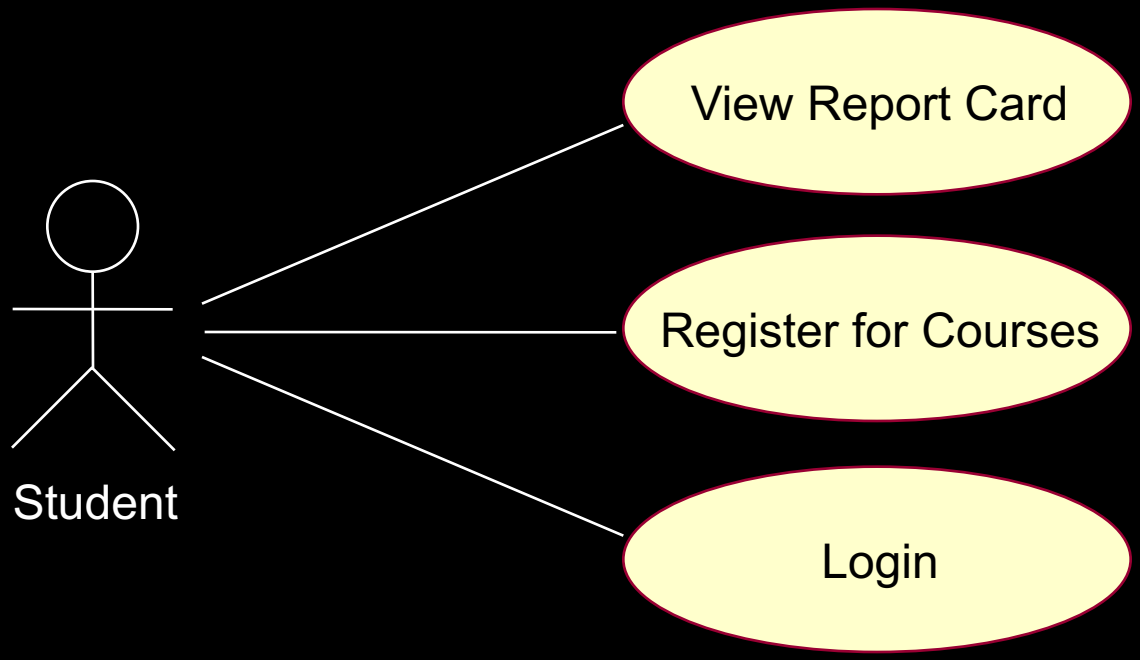
Review: What Is a Use-Case Model?

- 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)



Review: 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)



What Is an Activity Diagram?

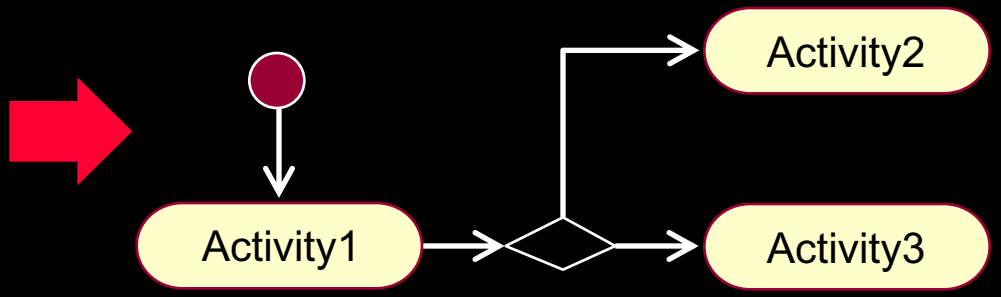
- An activity diagram in the Use-Case Model can be used to capture the activities in a use case.
- It is essentially a flow chart, showing flow of control from activity to activity.

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.



What Is an Activity Diagram?

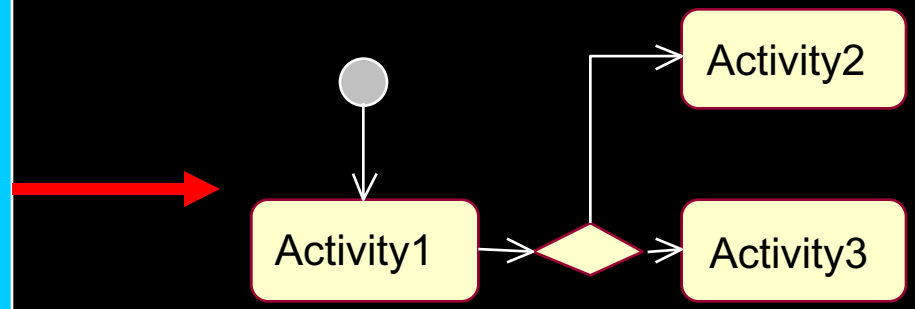
- An activity diagram in the Use-Case Model can be used to capture the activities in a use case.
- It is essentially a flow chart, showing flow of control from one activity or action to another.

Flow of Events

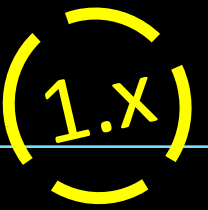
This use case starts when the Registrar requests that the system close registration.

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.

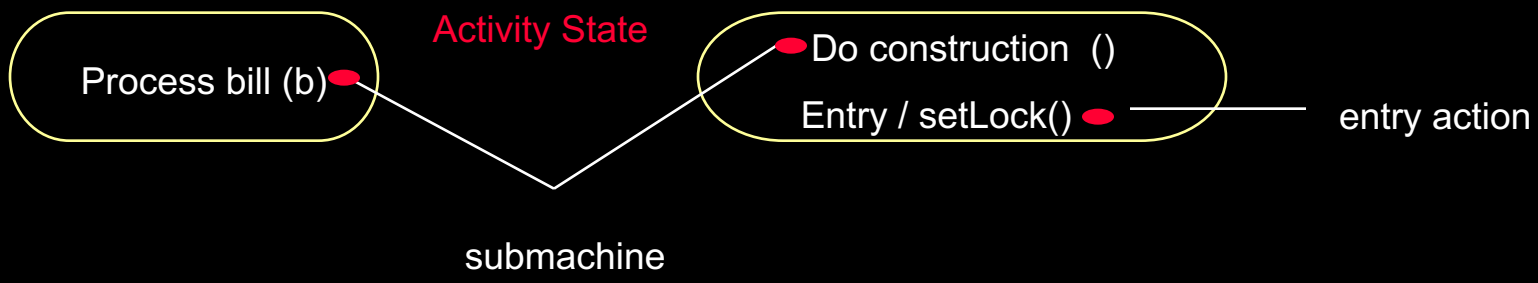
2. For each course offering, the system checks if a professor has signed up to teach the course offering and at least three students have registered. If so, the system commits the course offering for each schedule that contains it.



What Is an Activity State?

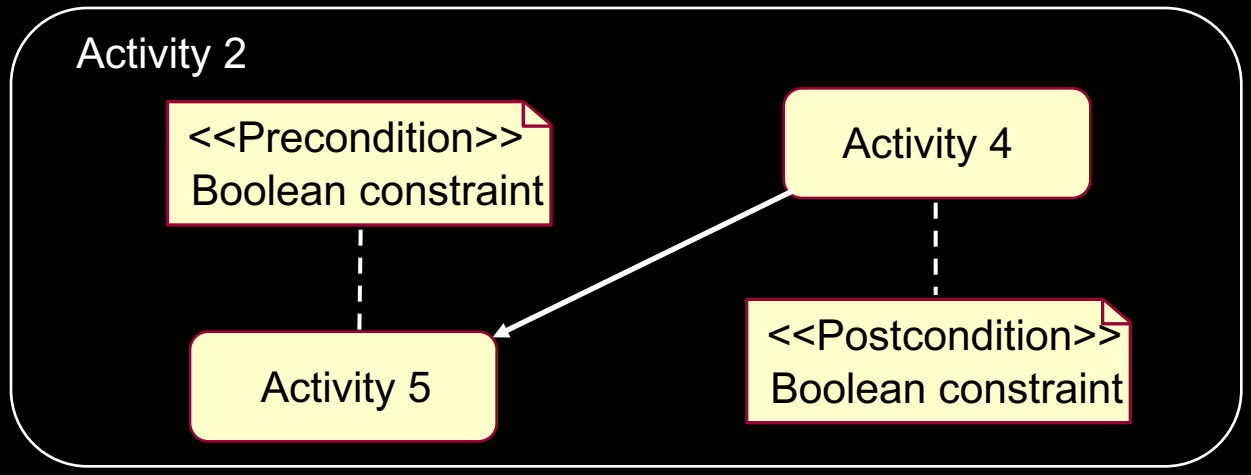


- The performance of an activity or step within the workflow.
- An activity is an operation that takes time to complete. It:
  - May have additional parts, such as entry and exit actions
  - Can have submachine specifications

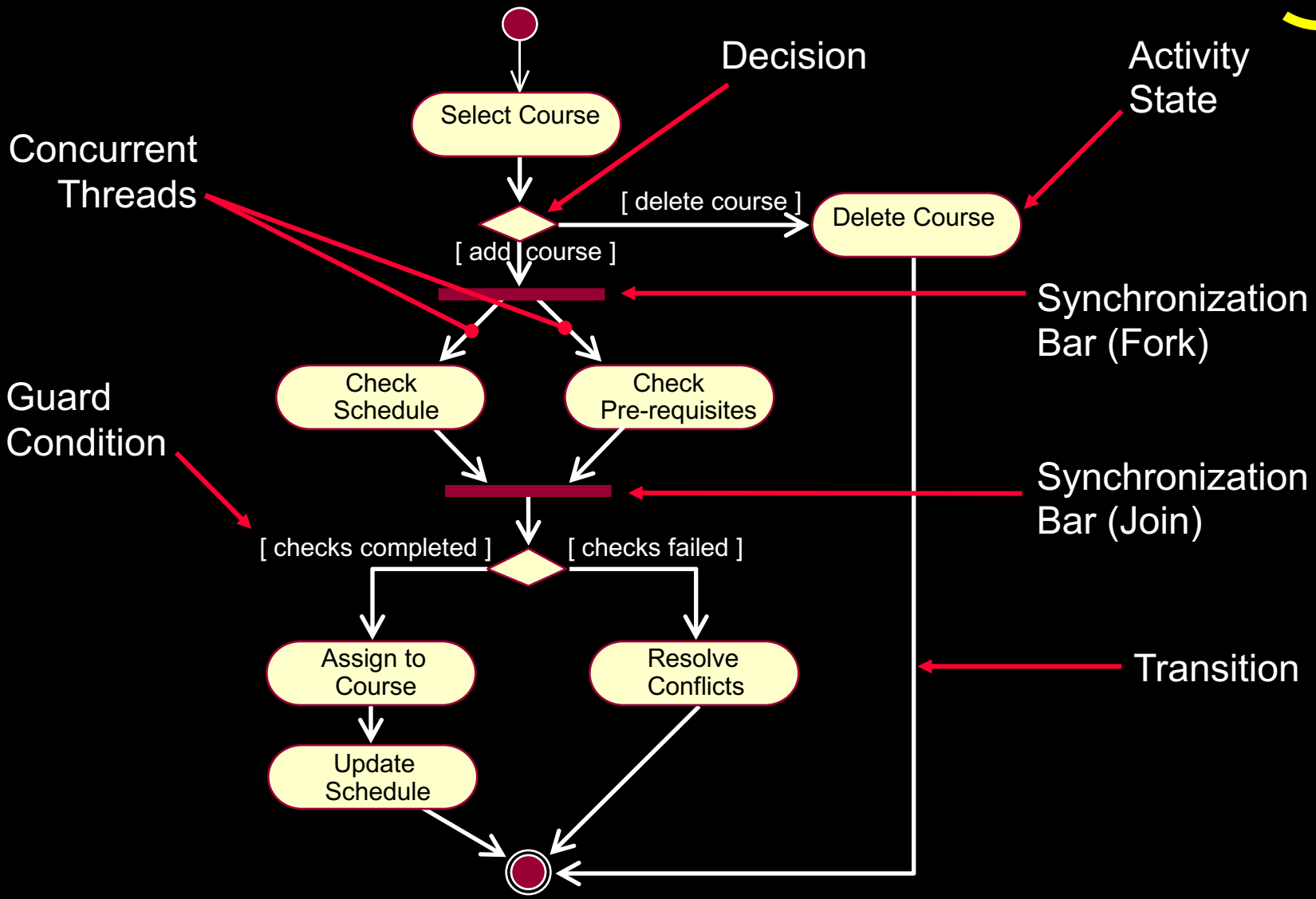


What Is an Activity?

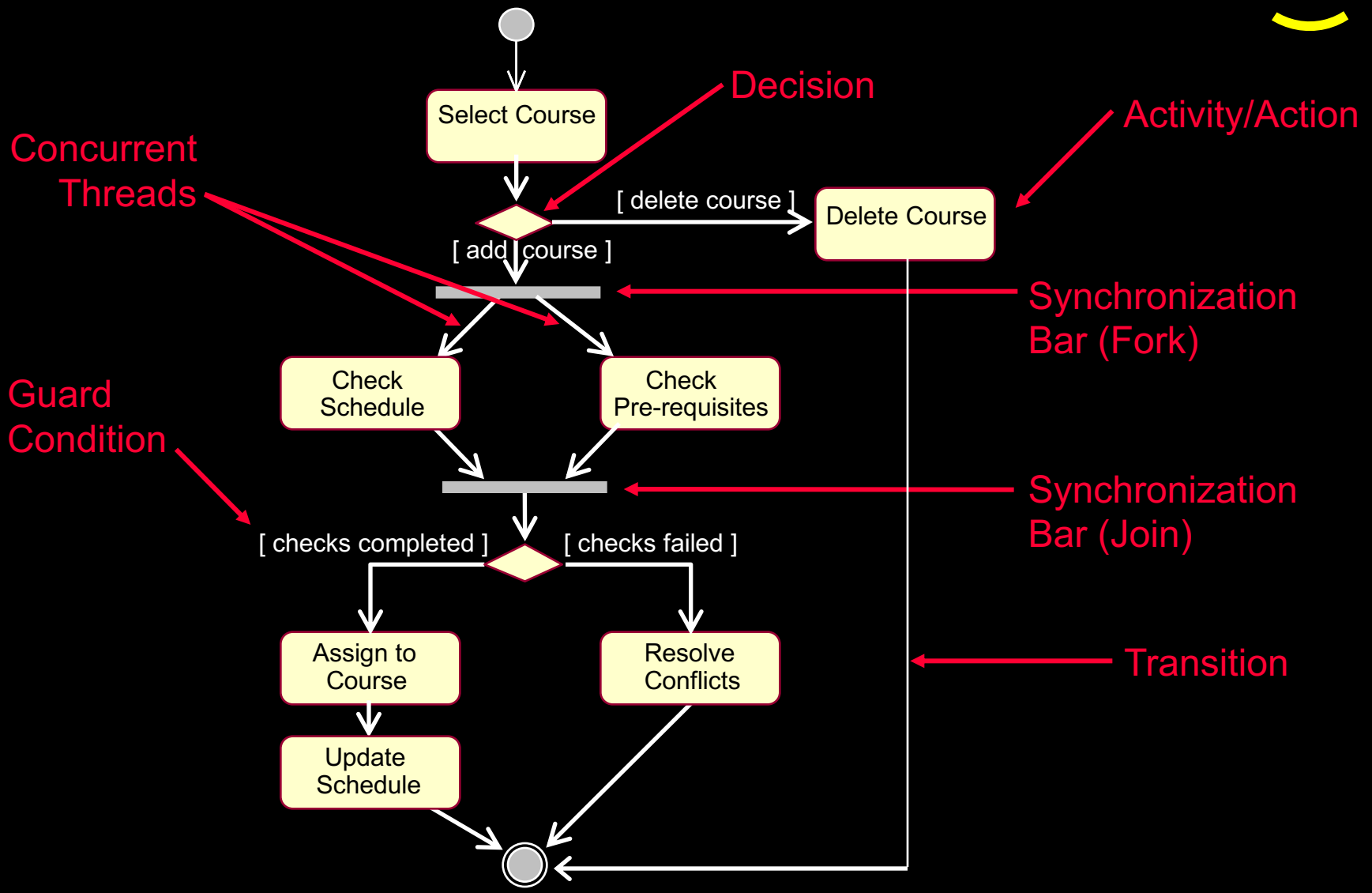
- A specification of behavior expressed as a flow of execution through sequencing of subordinate units.
  - Subordinate units include nested activities and ultimately individual actions.
- Activities may contain Boolean expression constraints when invoked or exited



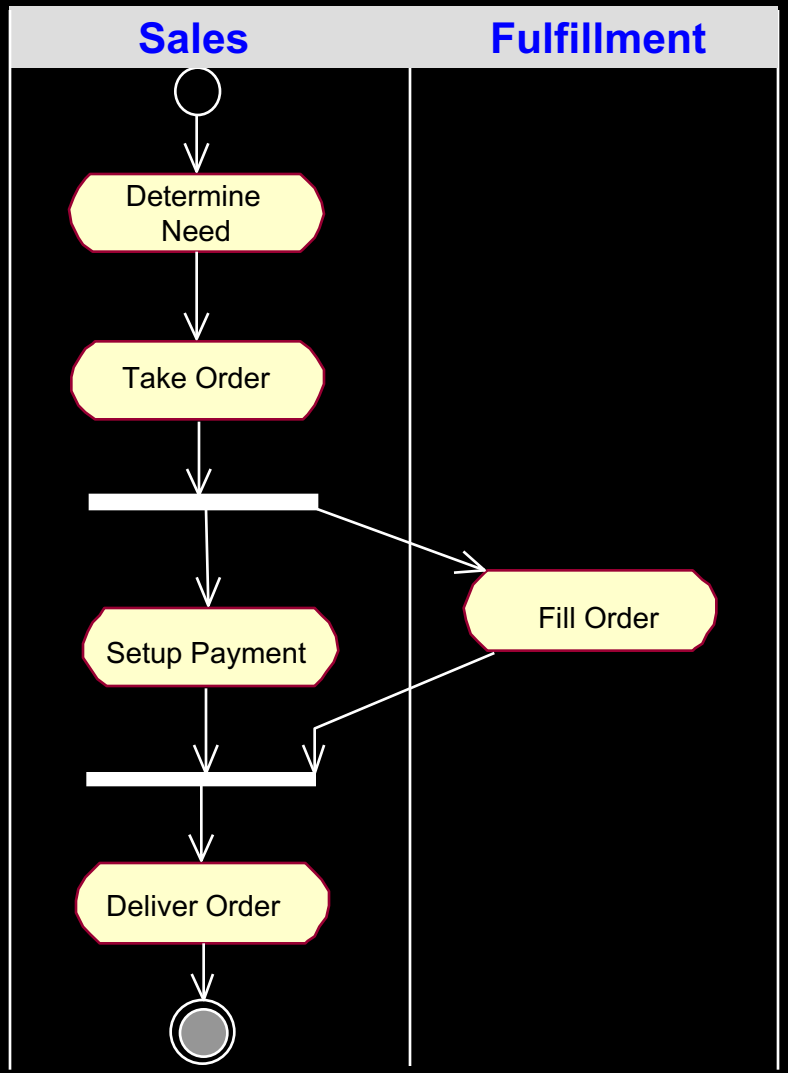
Example: Activity Diagram



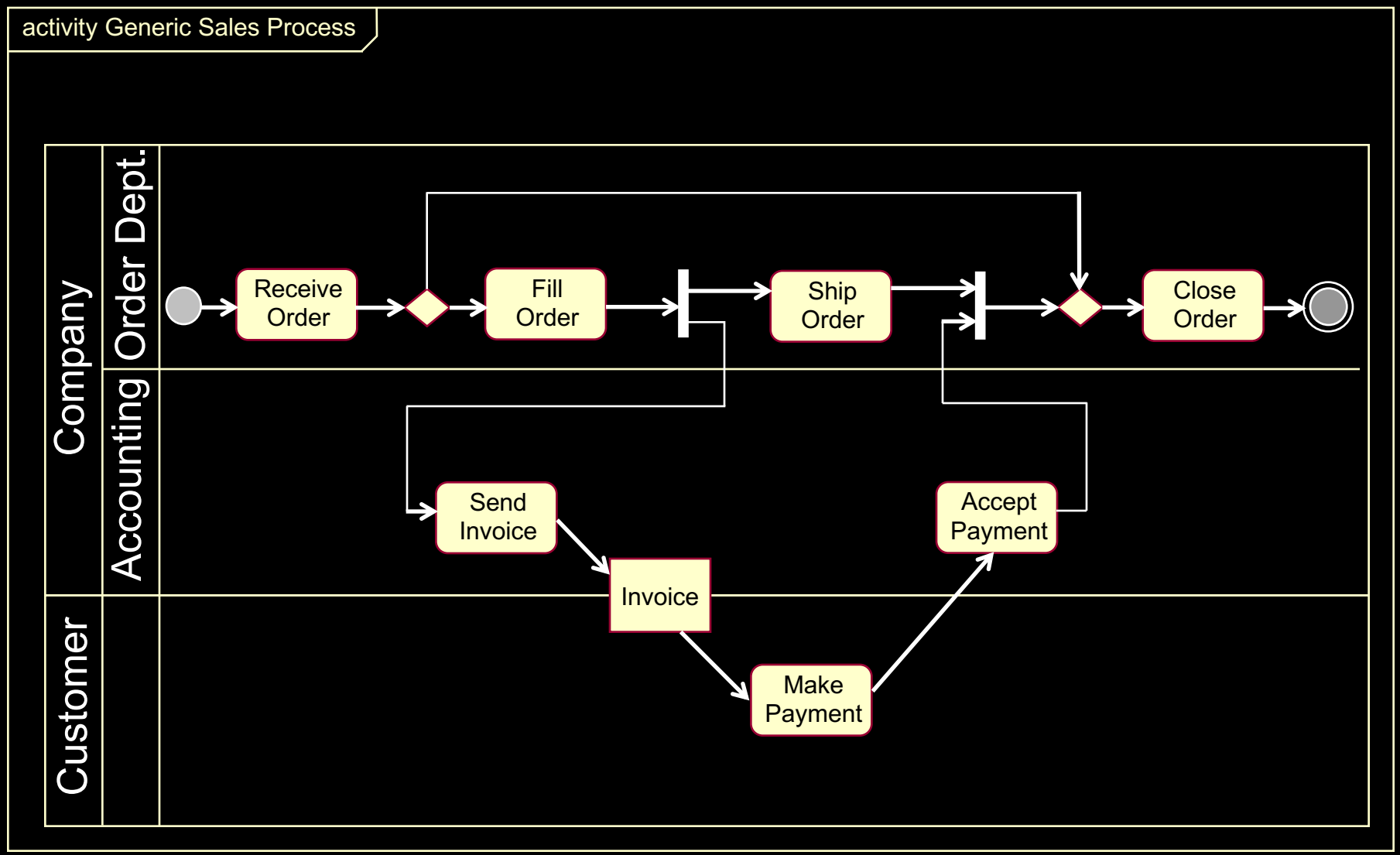
Example: Activity Diagram



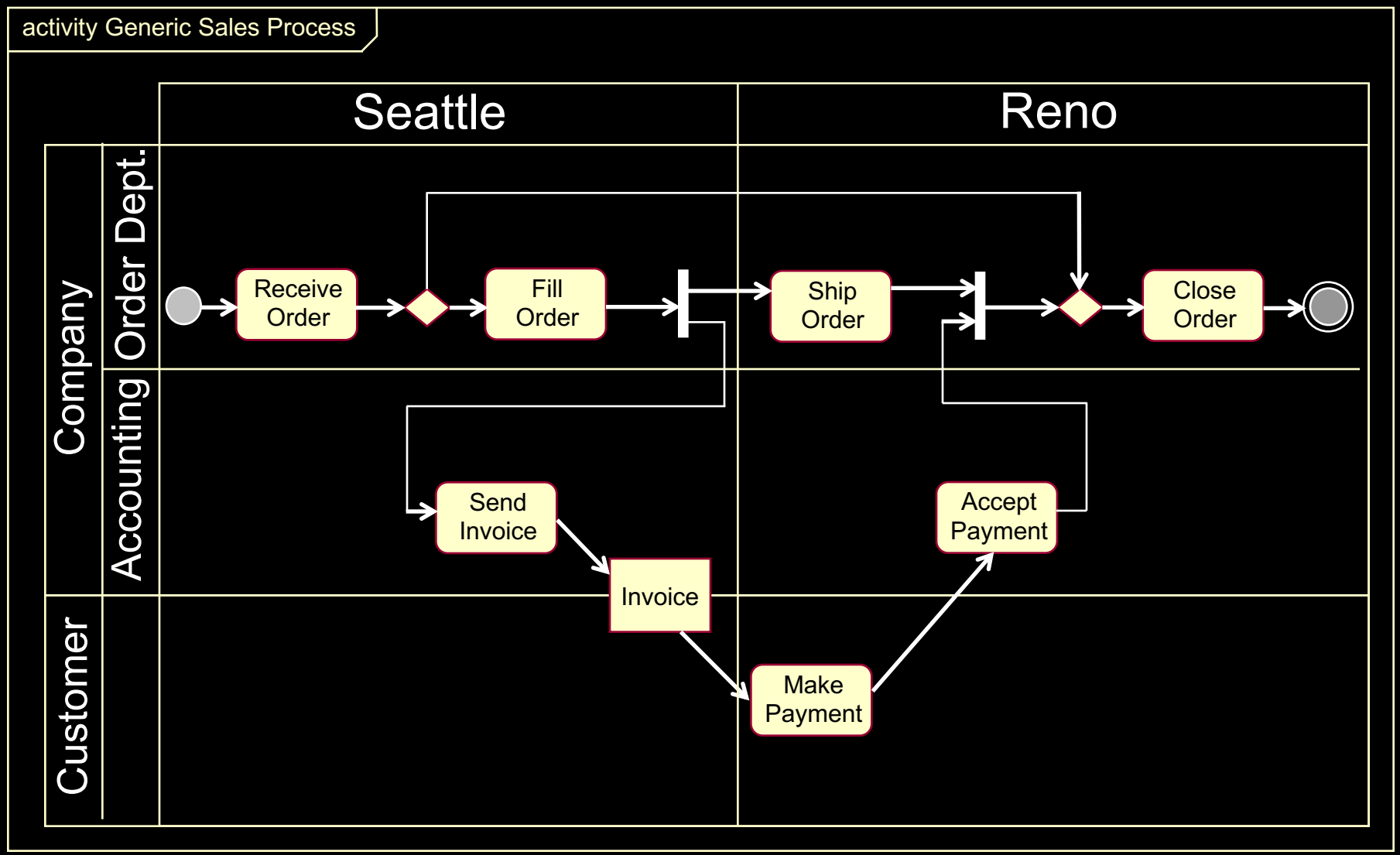
Swimlanes


Sales
Fulfillment
Determine Need
Take Order
Fill Order
Setup Payment
Deliver Order

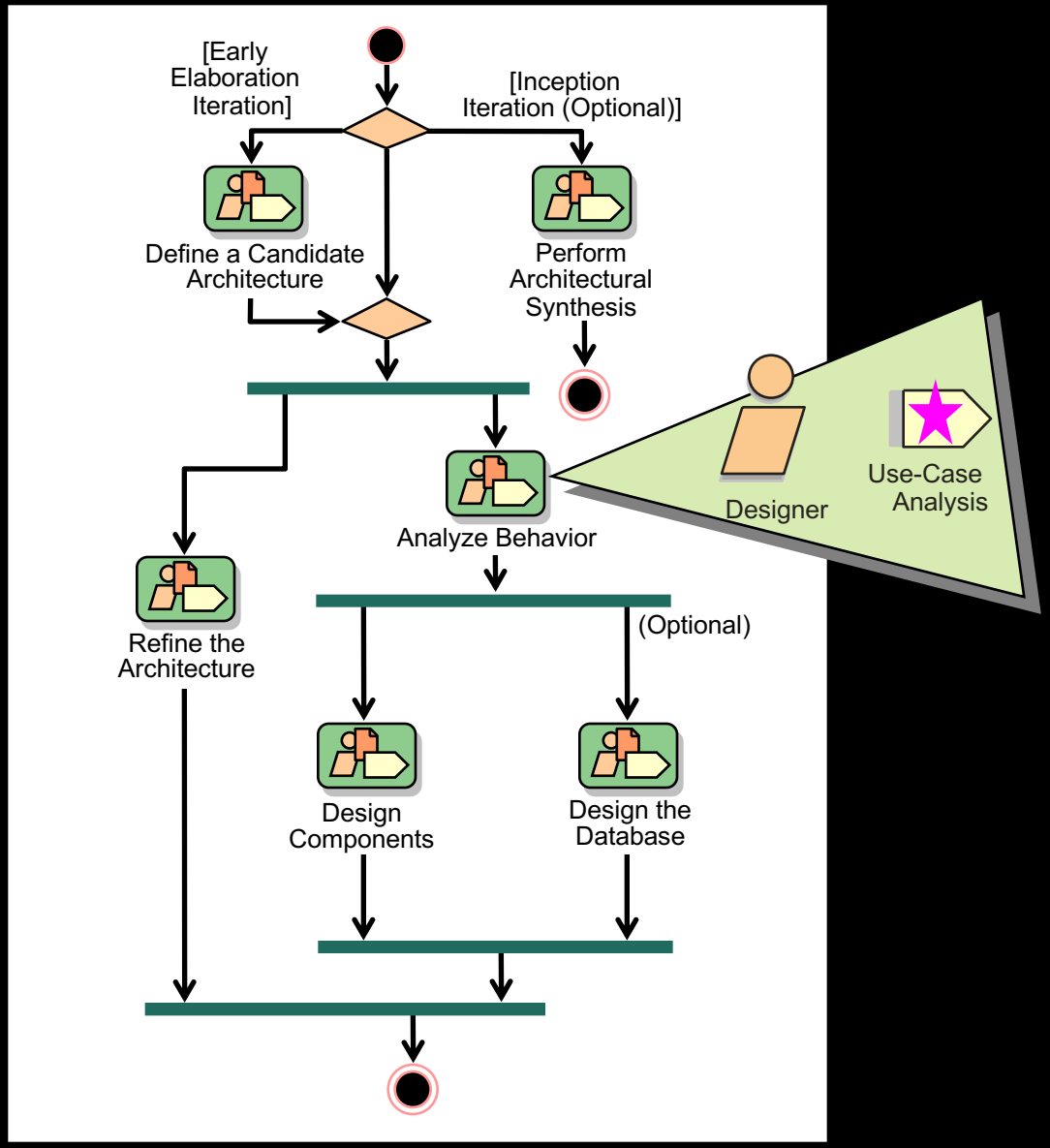
Partitions


activity Generic Sales Process
Company
Order Dept.
Accounting
Customer
Receive Order
Fill Order
Ship Order
Close Order
Send Invoice
Invoice
Make Payment
Accept Payment

Partitions: Example of Two-Dimensional



Use-Case Analysis in Context

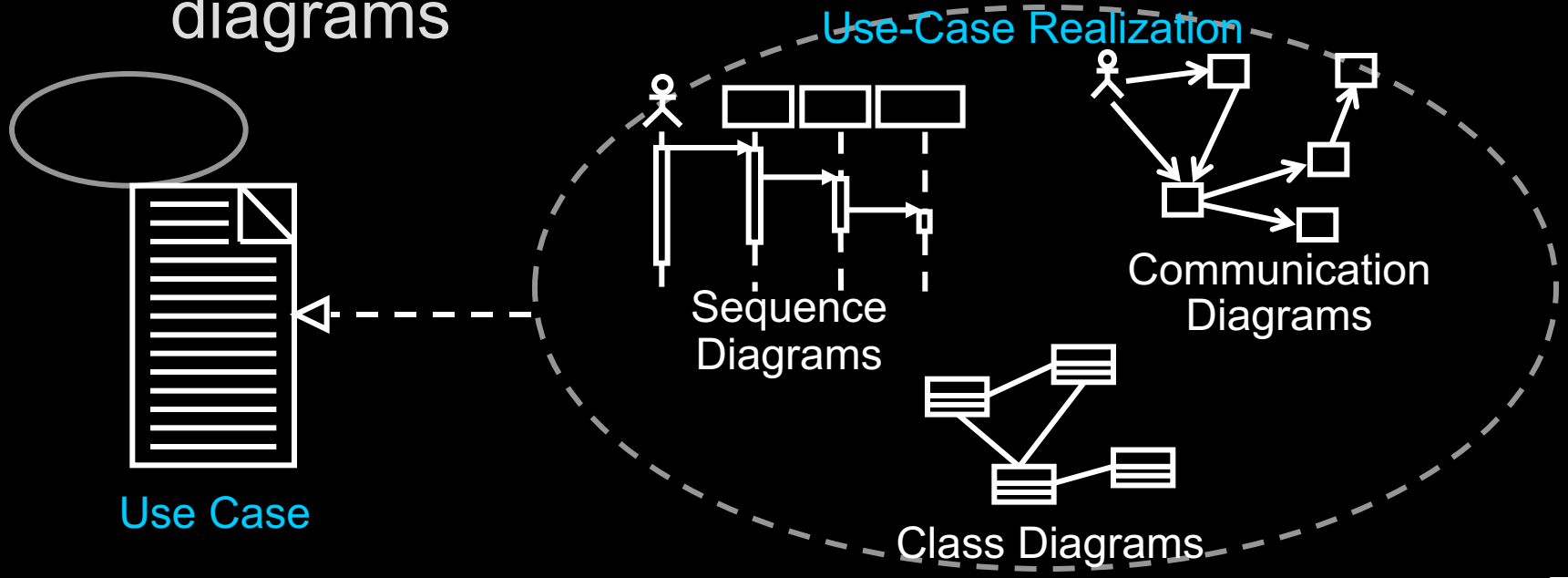


Use-Case Analysis Steps

- Supplement the Use-Case Descriptions
- For each Use-Case Realization
  - Find Classes from Use-Case Behavior
  - ★ Distribute Use-Case Behavior to Classes
- For each resulting analysis class
  - Describe Responsibilities
  - Describe Attributes and Associations
  - Qualify Analysis Mechanisms
- Unify Analysis Classes
- Checkpoints

Distribute Use-Case Behavior to Classes

- For each use-case flow of events:
  - Identify analysis classes
  - Allocate use-case responsibilities to analysis classes
  - Model analysis class interactions in Interaction diagrams



Guidelines: Allocating Responsibilities to Classes

- Use analysis class stereotypes as a guide
  - Boundary Classes - Behavior that involves communication with an actor
  - Entity Classes - Behavior that involves the data encapsulated within the abstraction
  - Control Classes - Behavior specific to a use case or part of a very important flow of events

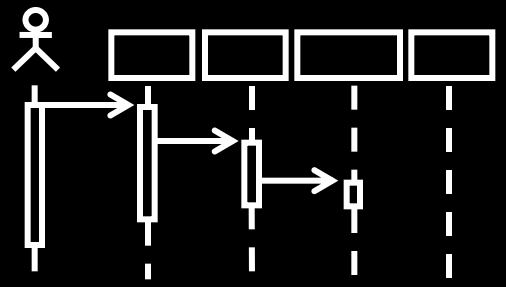
Guidelines: Allocating Responsibilities to Classes (cont.)

- Who has the data needed to perform the responsibility?
  - If one class has the data, put the responsibility with the data
  - If multiple classes have the data:
    - Put the responsibility with one class and add a relationship to the other
    - Create a new class, put the responsibility in the new class, and add relationships to classes needed to perform the responsibility
    - Put the responsibility in the control class, and add relationships to classes needed to perform the responsibility

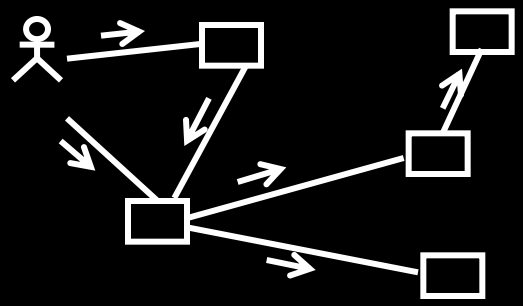
What Is an Interaction Diagram?

1.x

- An interaction diagram shows an interaction that consists of a set of objects and their relationships, including the messages that may be dispatched among them.
- It models the dynamic aspects of a system.



Sequence Diagrams

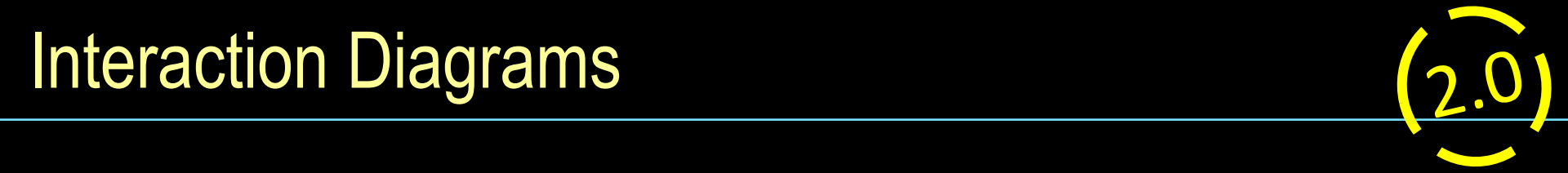


Collaboration Diagrams

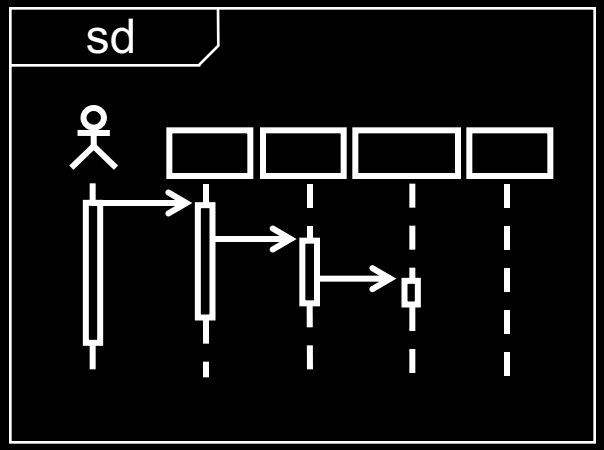
What is an Interaction Diagram?

- Generic term that applies to several types of diagrams that emphasize object interactions, including the messages that may be dispatched among them.
  - Sequence Diagram
  - Communication Diagram
  - Timing Diagram
  - Interaction Overview Diagram

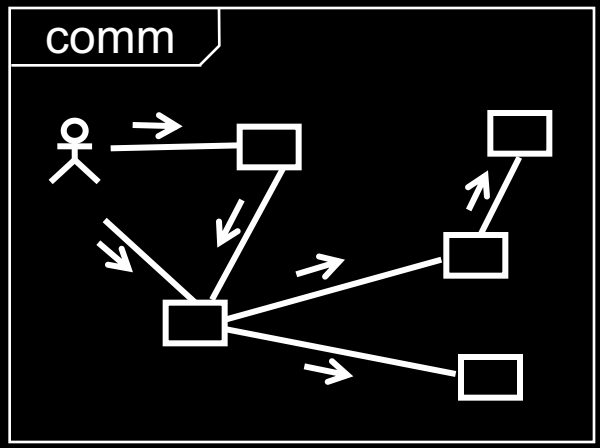
Interaction Diagrams



- Sequence Diagram
  - Time oriented view of interactions



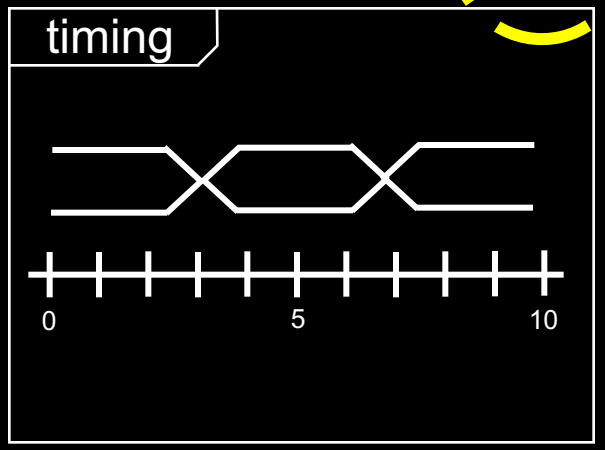
- Communication Diagram
  - Structural view of messaging roles or parts



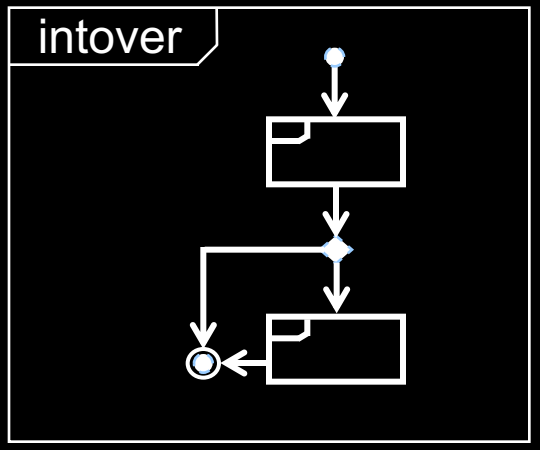
Interaction Diagrams (continued)

2.0

- Timing Diagram
  - Time constraint view of messages involved in an interaction



- Interaction Overview Diagram
  - High level view of interaction sets combined into logic sequence



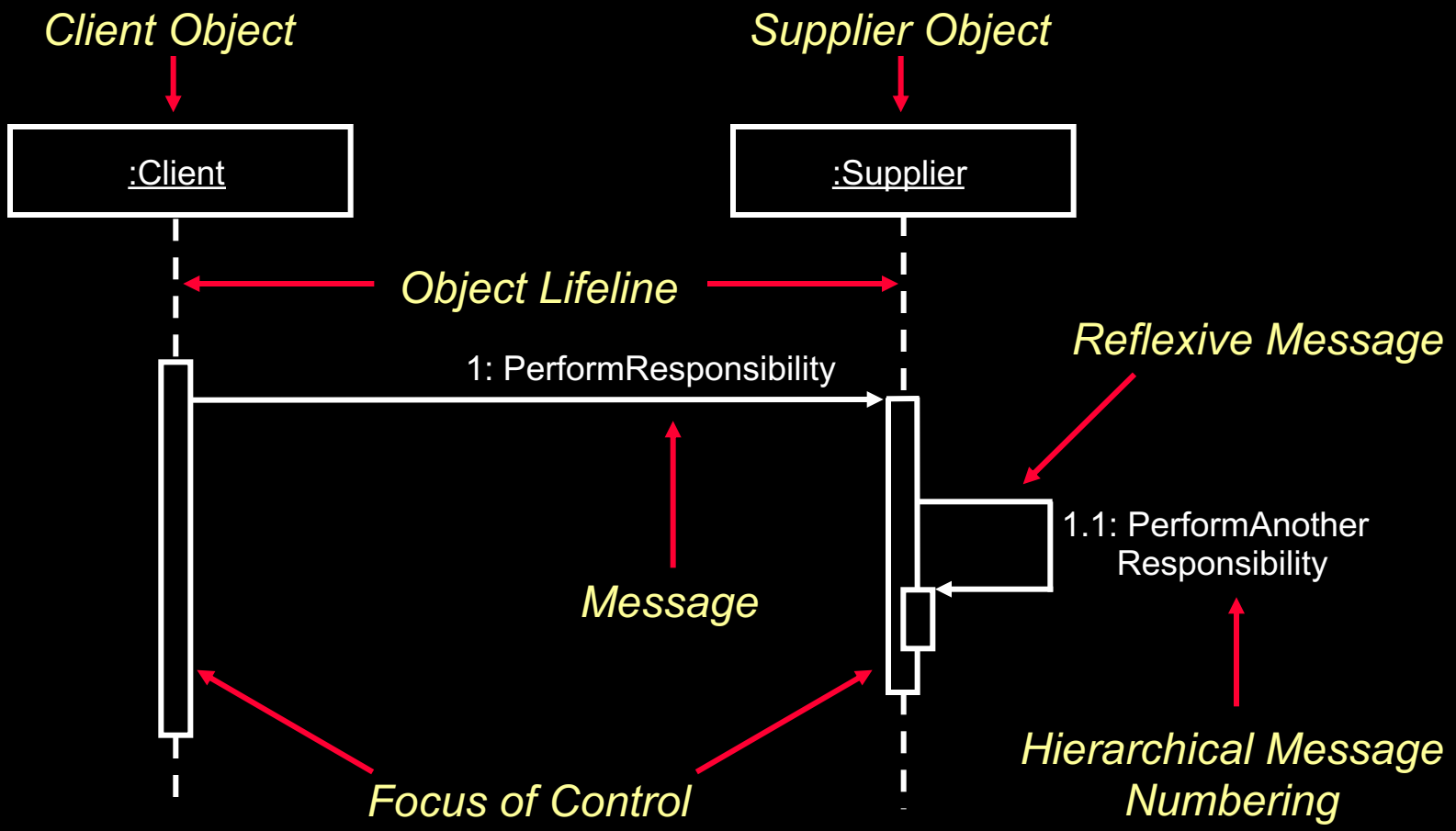
IBM

Sequence Diagrams vs. Communication Diagrams

Sequence Diagrams	Communication Diagrams
▪ Show the explicit sequence of messages ▪ Better for visualizing overall flow ▪ Better for real-time specifications and for complex scenarios	▪ Show relationships in addition to interactions ▪ Better for visualizing patterns of collaboration ▪ Better for visualizing all of the effects on a given role or part ▪ Easier to use for brainstorming sessions

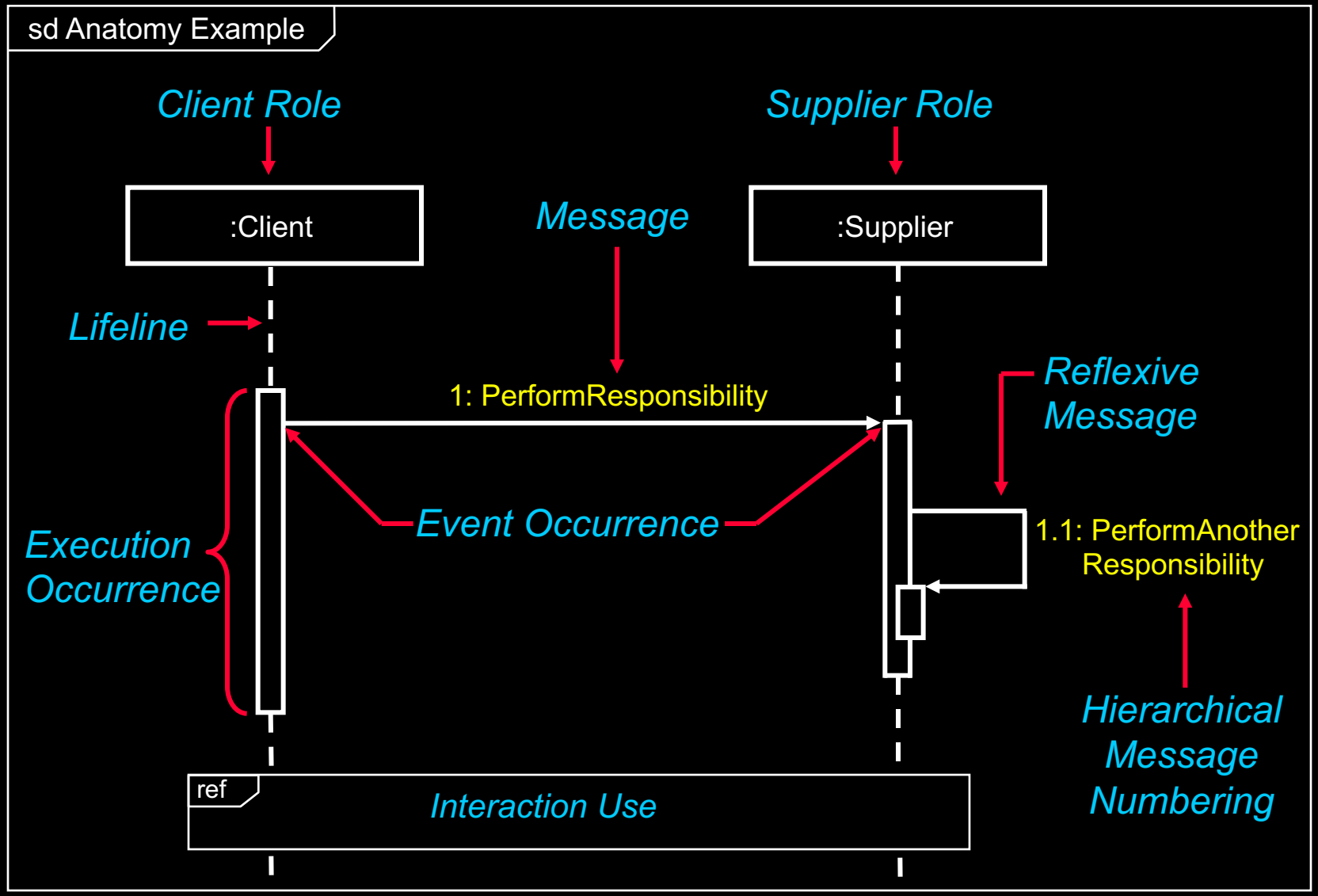
The Anatomy of Sequence Diagrams

This is a sample script.

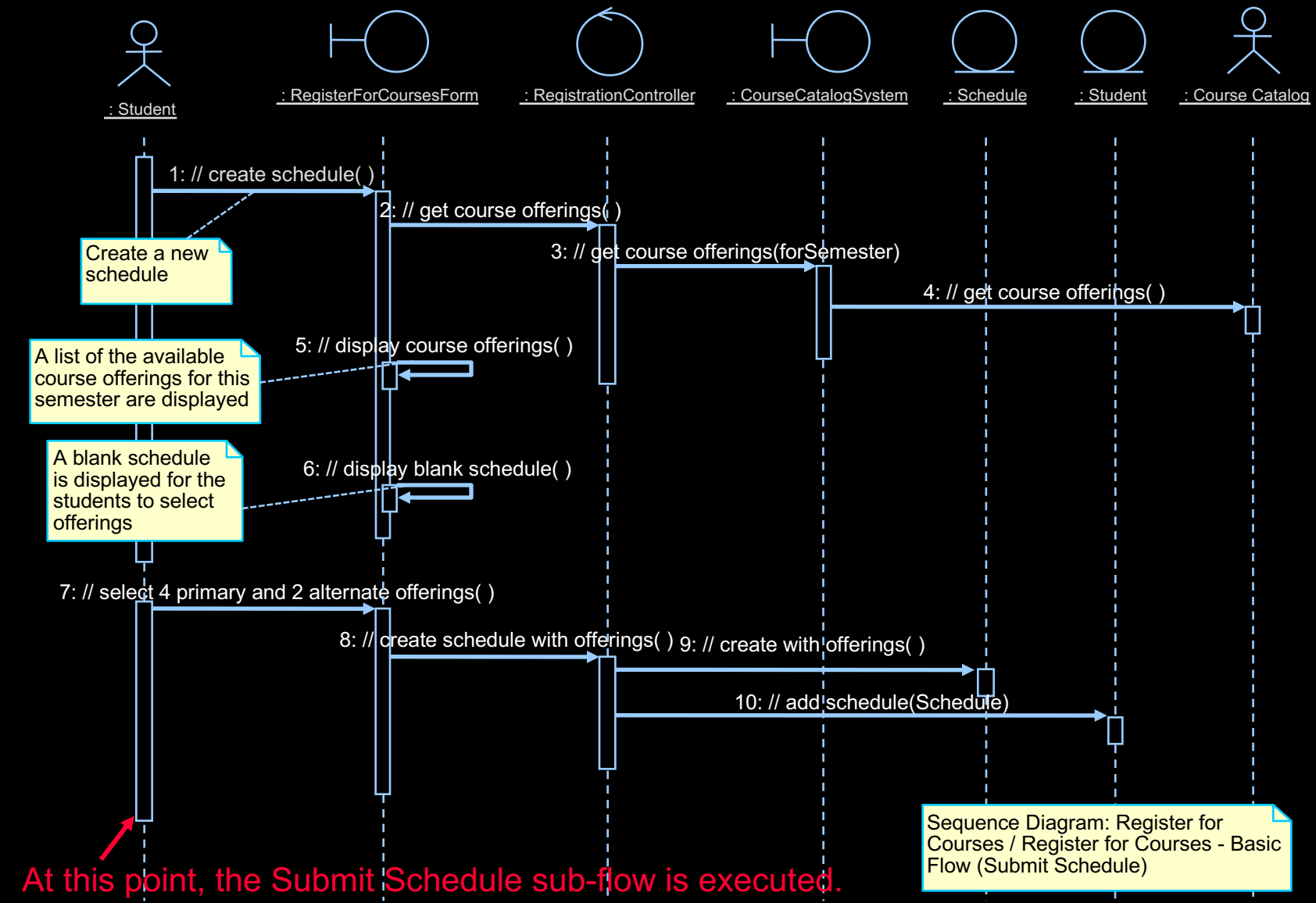


IBM

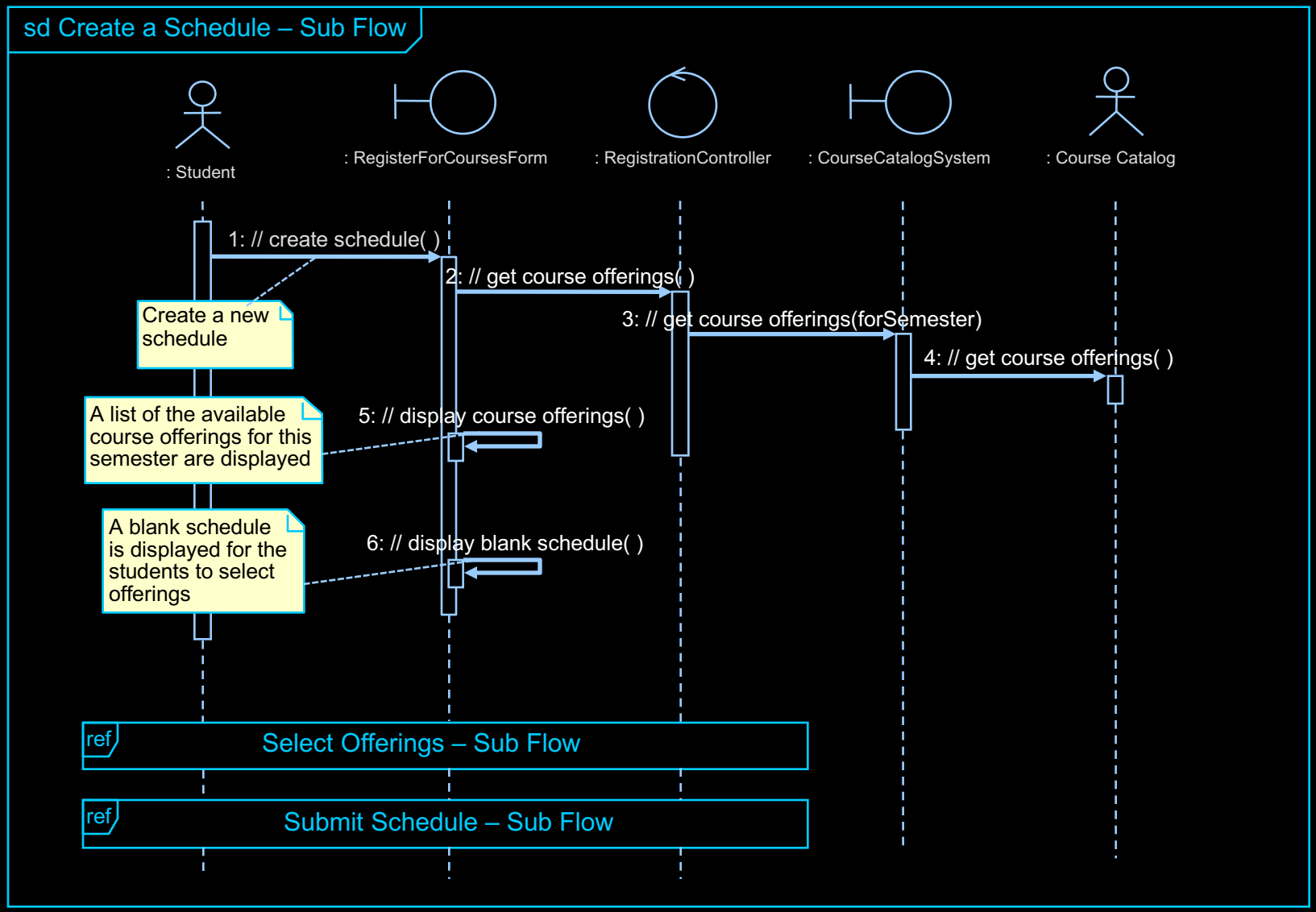
The Anatomy of Sequence Diagrams



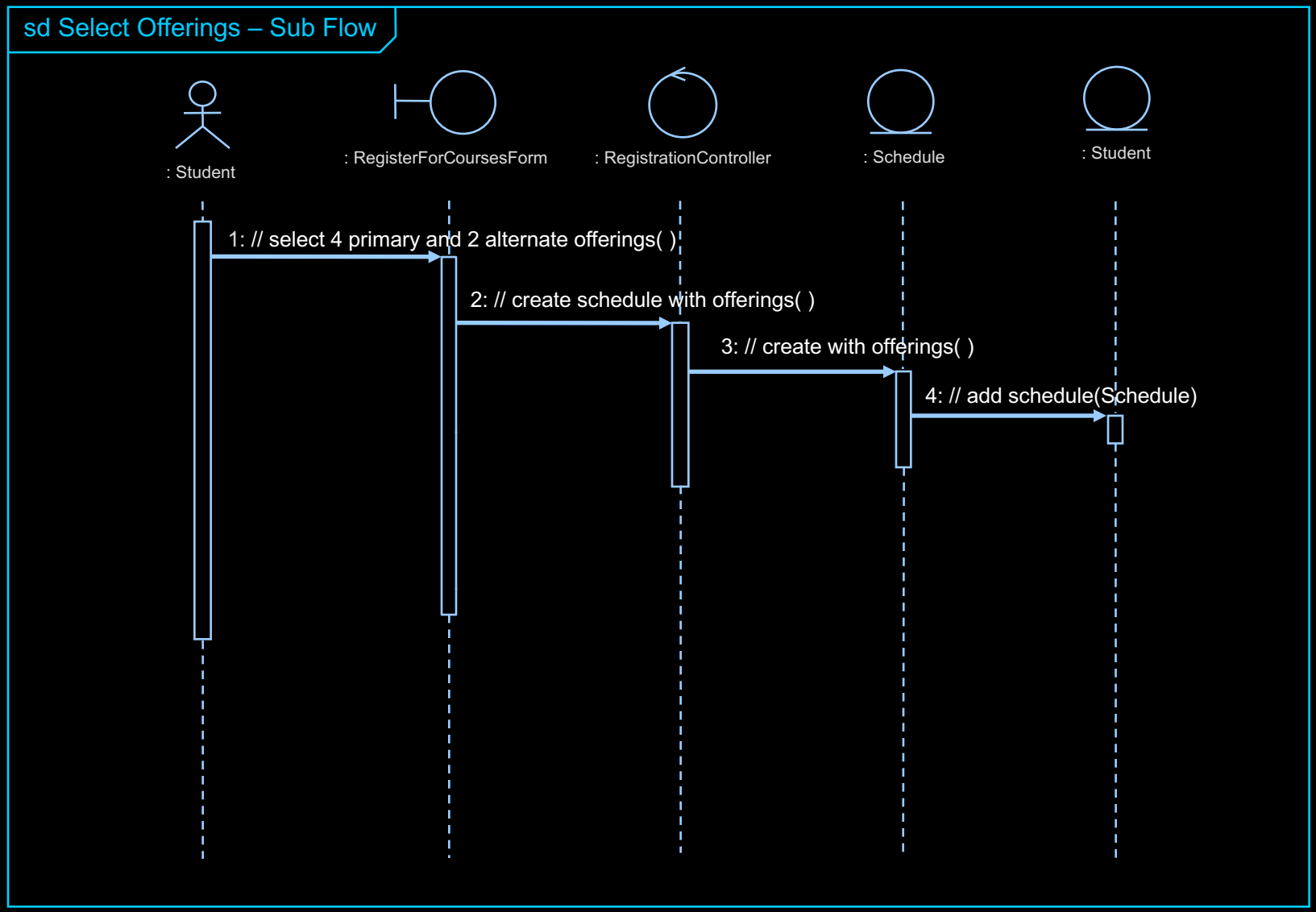
Example: Sequence Diagram



Example: Sequence Diagram

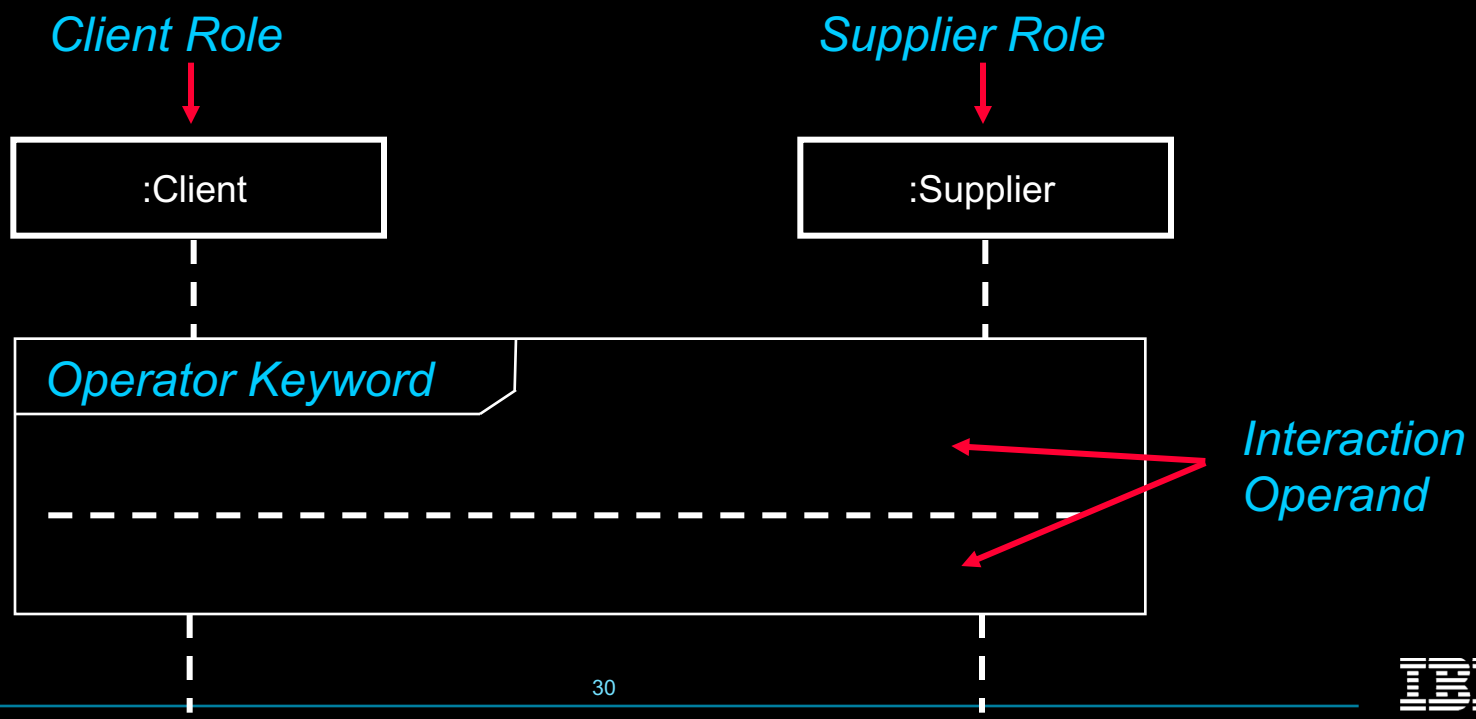


Example: Sequence Diagram (continued)



What Is a Combined Fragment?

- A grouping of behavior that comprises an operator keyword and one or more interaction operands, each of which is a fragment of an interaction.
  - It is shown as a nested region within a sequence diagram.



What is an Interaction Operand?

- Each fragment comprises one or more interaction operands, each a subfragment of the interaction.
  - The number of operands depends on the type of combined fragment. For example, a loop has one operand (the loop body) and an alternative has one or more operands (the branches of the conditional).
  - An operand is a nested fragment of an interaction. Each operand covers the lifelines covered by the combined fragment or a subset of them.

Combined Fragment Types

- Alternatives (**alt**) - Dynamic choice of behaviors where at most one subfragment will execute
- Assertion (**assert**) - Subfragment must be executed if reached
- Break (**break**) - Represents an alternative that is executed instead of the remainder of the fragment (like a break in a loop)

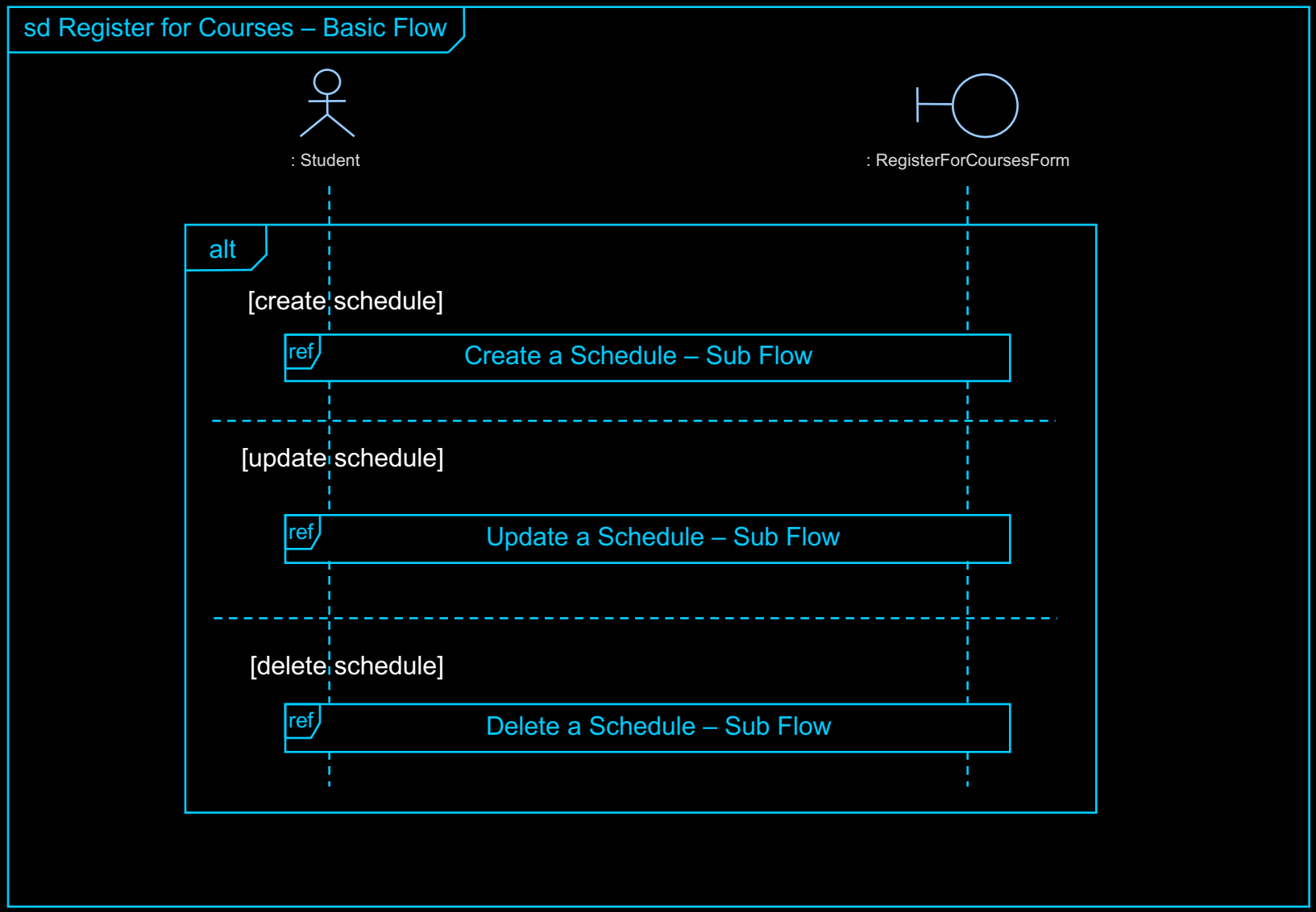
Combined Fragment Types (continued)

- Consider (**consider**)
  - List of message types that are represented for consideration within the subfragment
- Critical Region (**critical**)
  - Events on a single lifeline cannot be interleaved with events in other regions
- Ignore (**ignore**)
  - List of message types that are represented within the subfragment to be ignored
- Loop (**loop**)
  - Will be repeated a number of times

Combined Fragment Types (continued)

- Negative (**neg**) - Identifies sequences that must not occur
- Option (**opt**) - Executed if the guard equates to true
- Parallel (**par**) - Concurrent (interleaved) subfragments
- Weak Sequencing (**seq**) - Events on same lifeline are ordered
- Strict Sequencing (**strict**) - Events on different lifelines are ordered

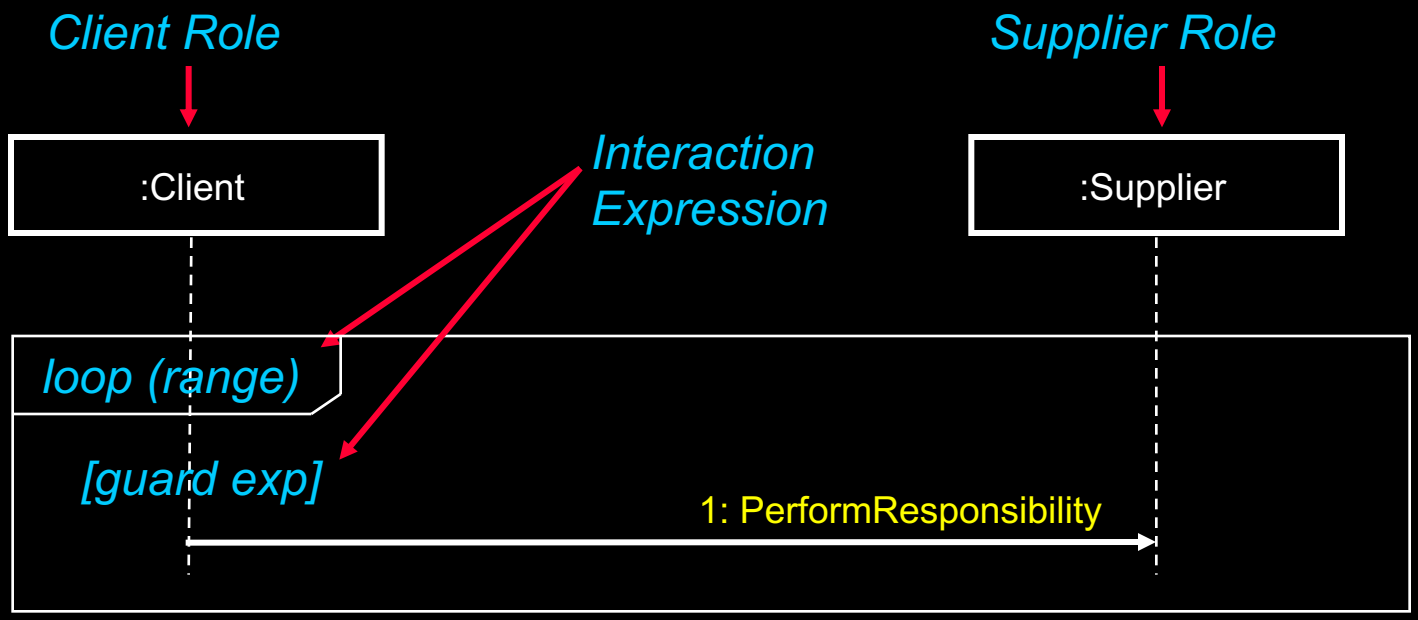
Example: Alternative Combined Fragment



What is an Interaction Expression?

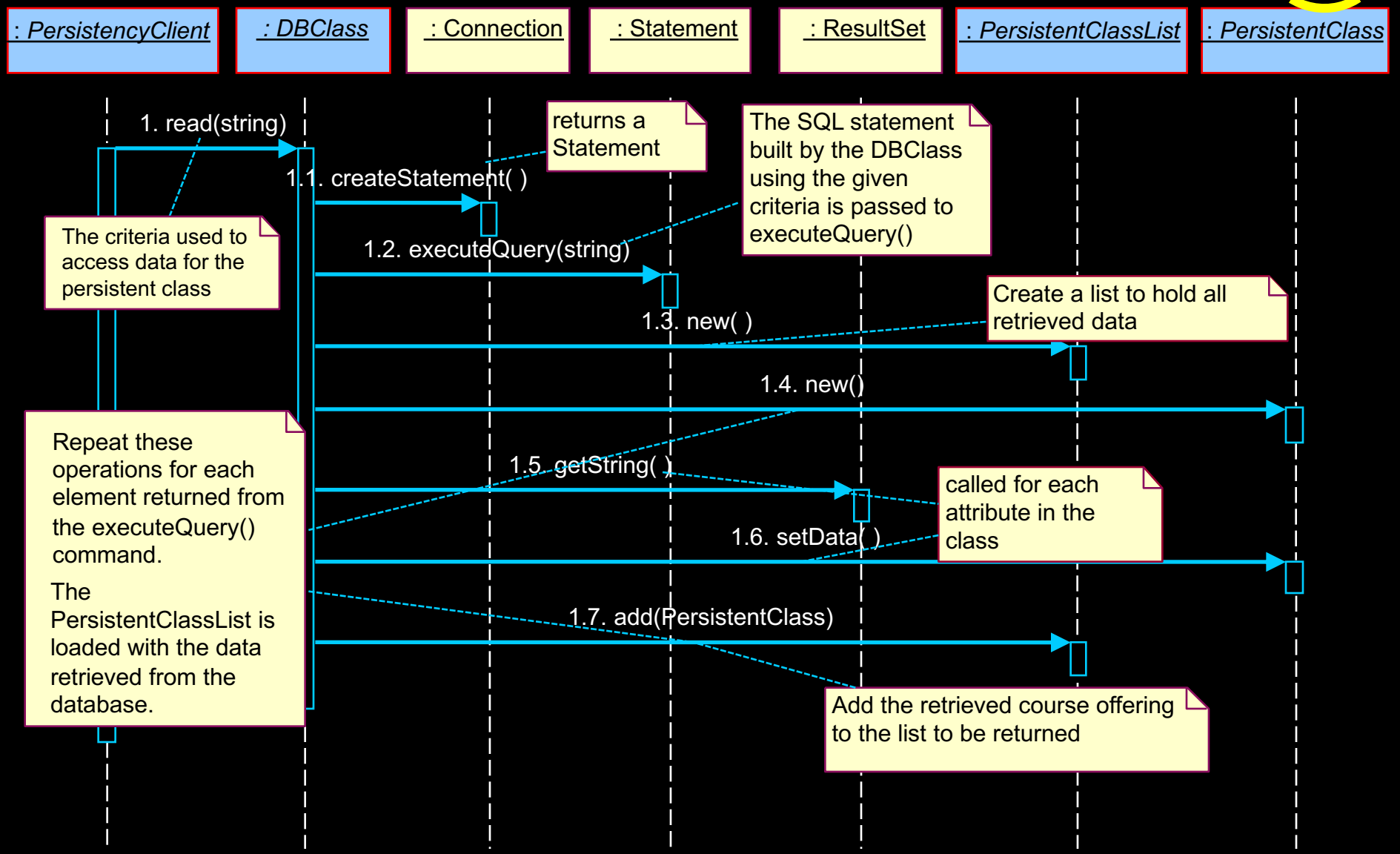
2.0

- A specification of the range of number of iterations of a loop.
  - The range can be specified with minimum and maximum values.
  - A guard condition, enclosed in square brackets, can be included on a lifeline.

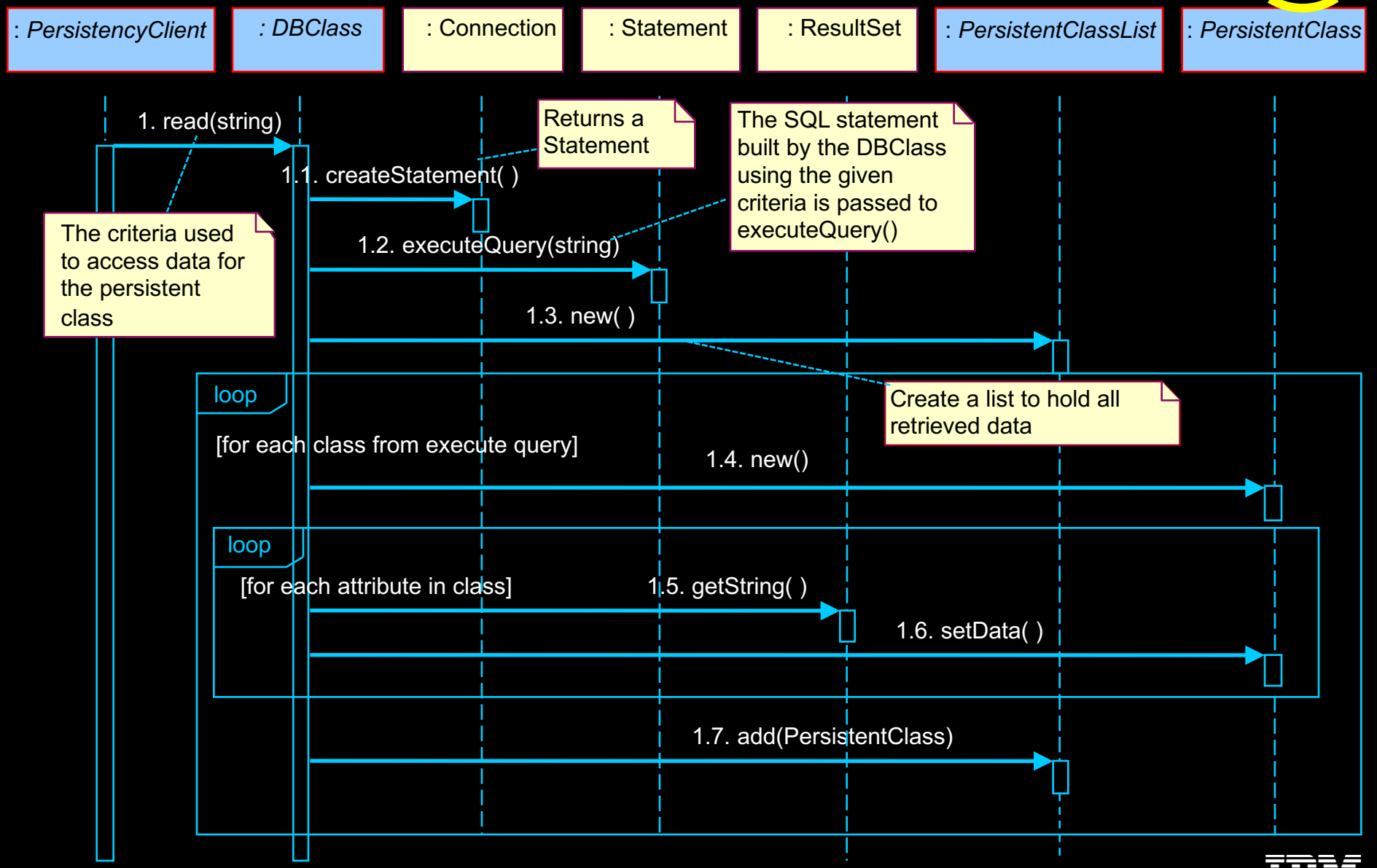


IBM

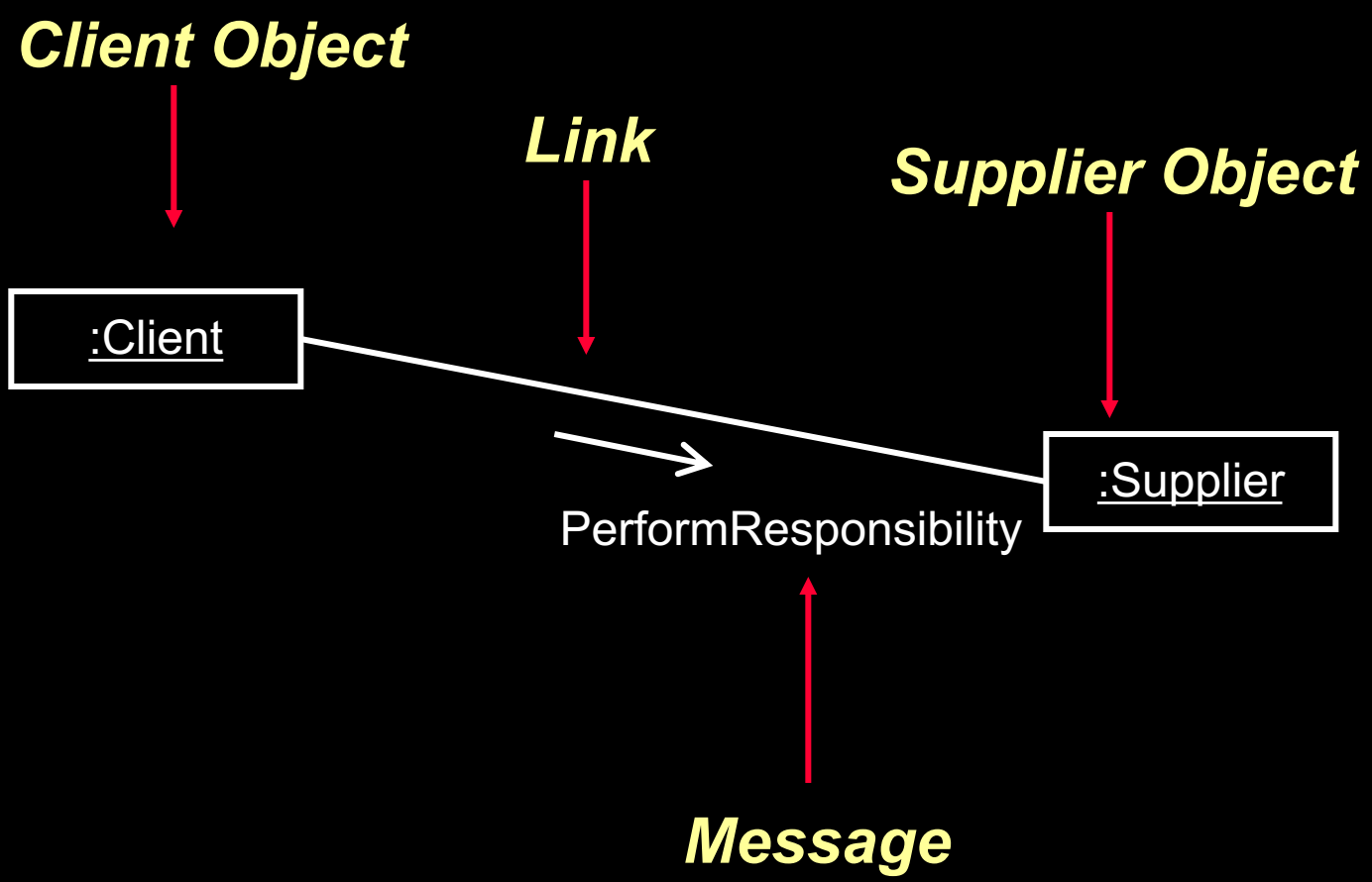
Example: Persistency: RDBMS: JDBC: Read



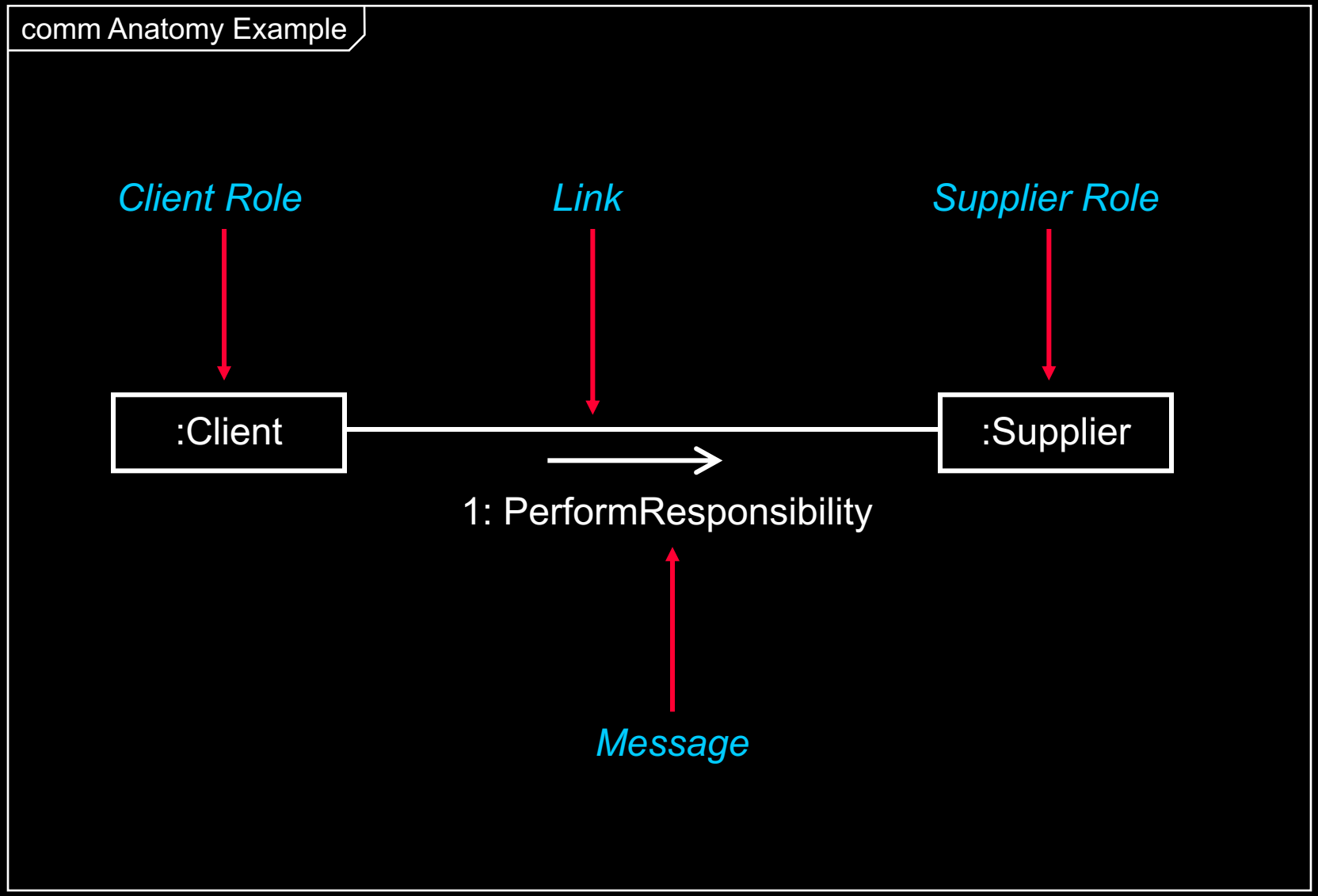
Example: Persistency: RDBMS: JDBC: Read



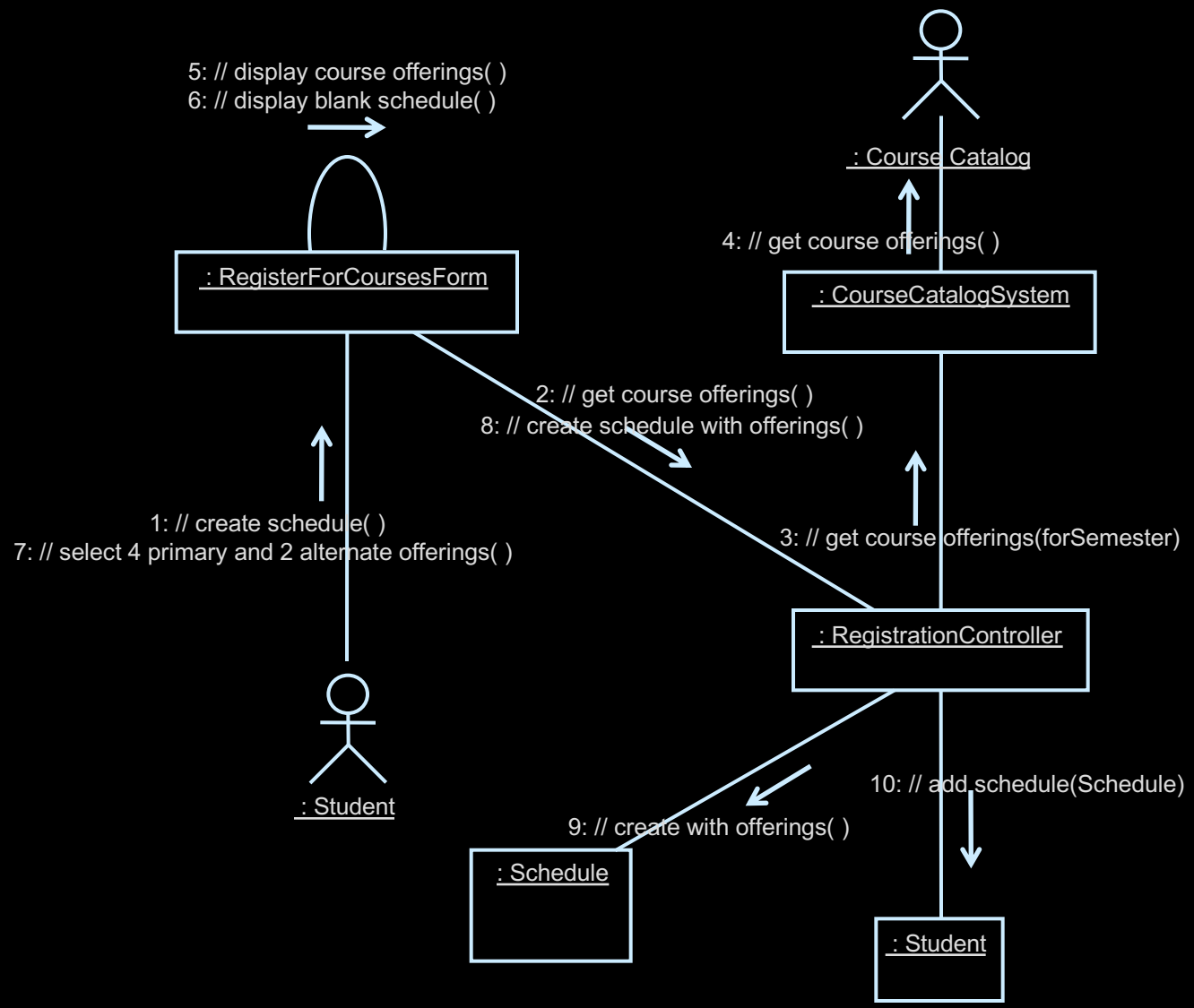
The Anatomy of Collaboration Diagrams



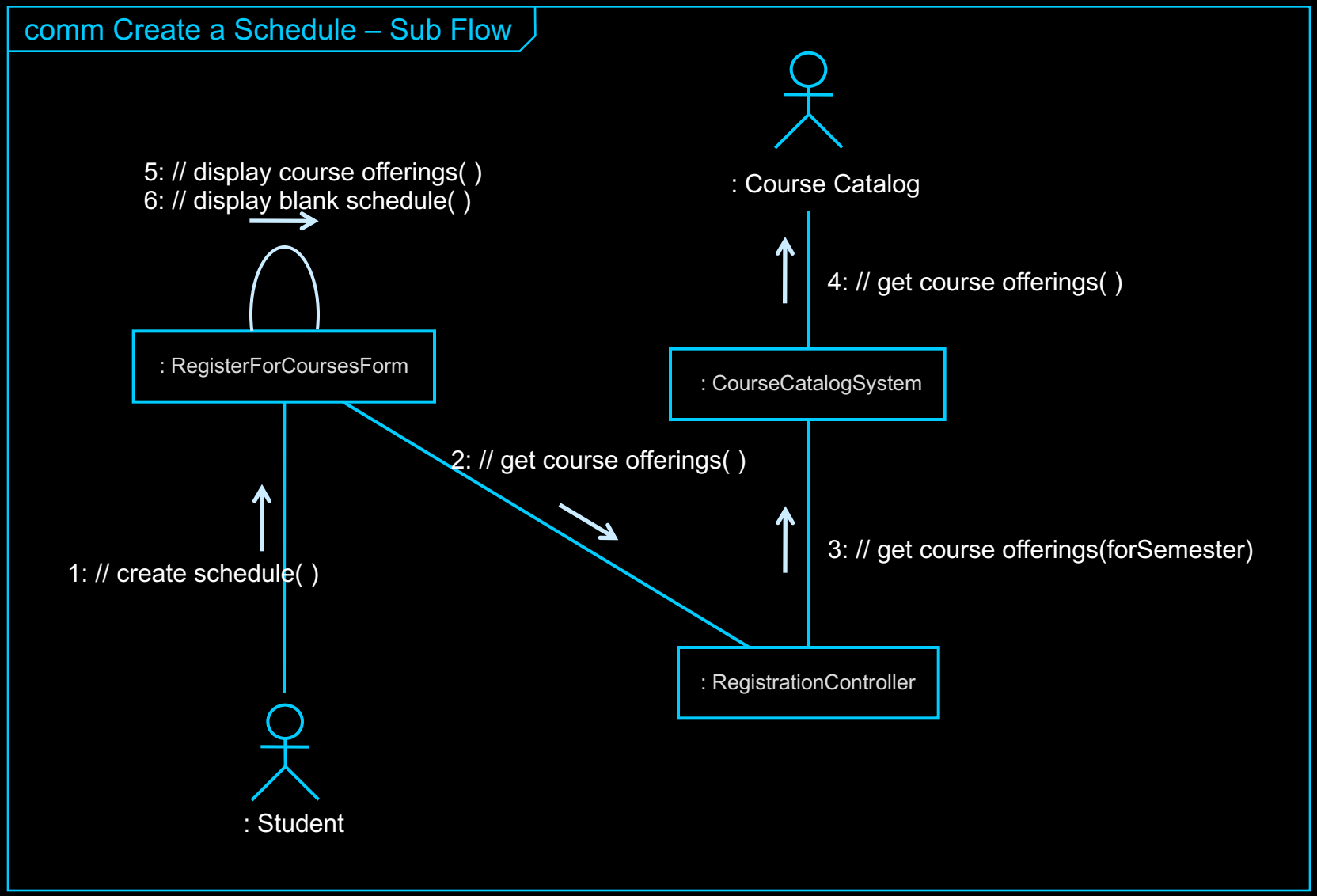
The Anatomy of Communication Diagrams



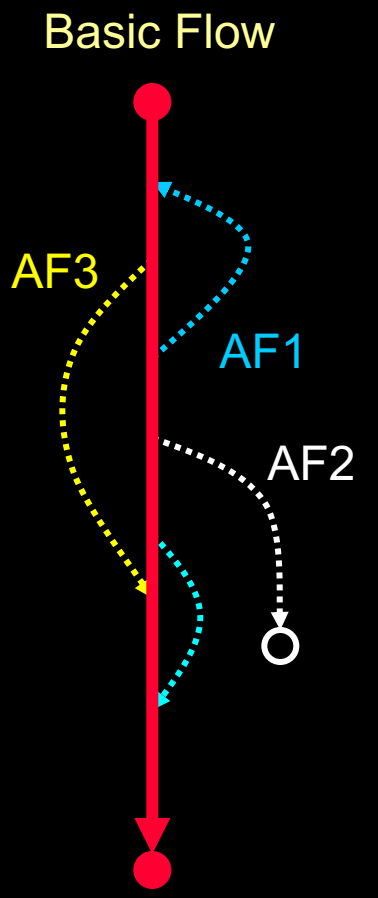
Example: Collaboration Diagram



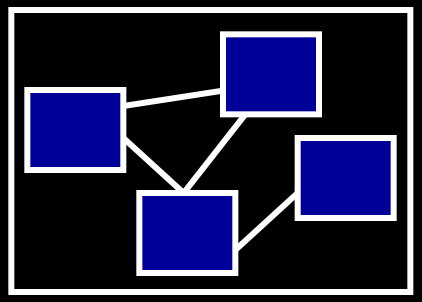
Example: Communication Diagram



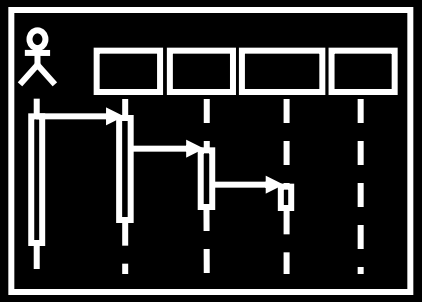
One Interaction Diagram Is Not Good Enough



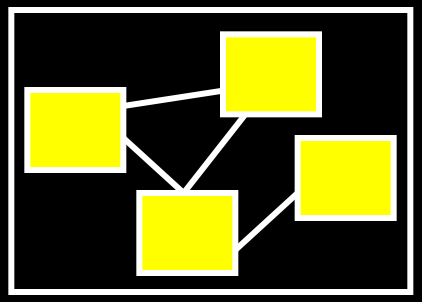
Alternate Flow 1



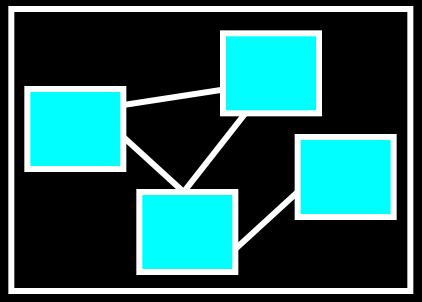
Alternate Flow 2



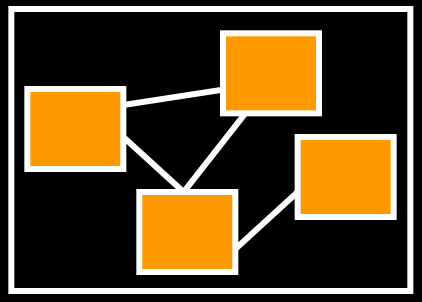
Alternate Flow 3



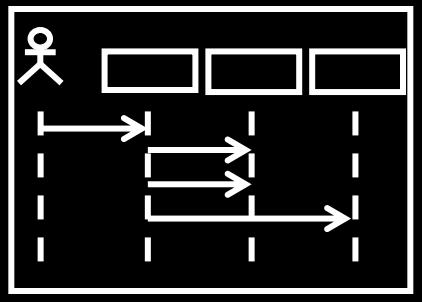
Alternate Flow 4



Alternate Flow 5



Alternate Flow n



Review: Requirements Overview and Use-Case Analysis

- What is an activity diagram and why would you use one?
- What is the difference between an activity and an action?
- What is a partition?
- What are the different types of interaction diagrams?
- What is a combined fragment?
- What are some examples of interaction operators?

Exercise: Use-Case Analysis

- Given the following:
  - Use-Case Model, especially the use-case flows of events
    - Exercise Workbook: Payroll Requirements, Use-Case Model section
  - Key abstractions/classes
    - Payroll Exercise Solution: Architectural Analysis section
  - The Supplementary Specification
    - Exercise Workbook: Payroll Requirements, Supplementary Specification section

Exercise: Use-Case Analysis (continued)

- Identify the following for a particular use case:
  - The collaborations needed to implement the use case
- Produce the following for a particular use case:
  - Use-Case Realization Interaction diagram for at least one of the use-case flows of events