

University of Science, VNU-HCM
Faculty of Information Technology

Dynamic Analysis

Assoc. Prof. TRAN Minh Triet
Department of Software Engineering



Software Analysis and Design



Reference

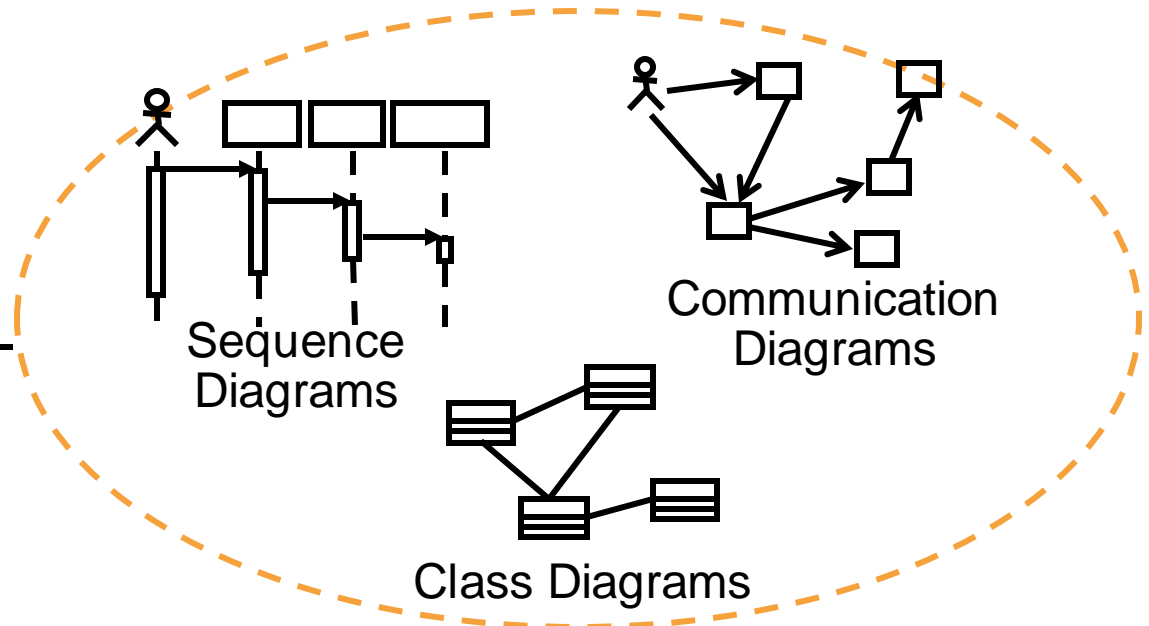
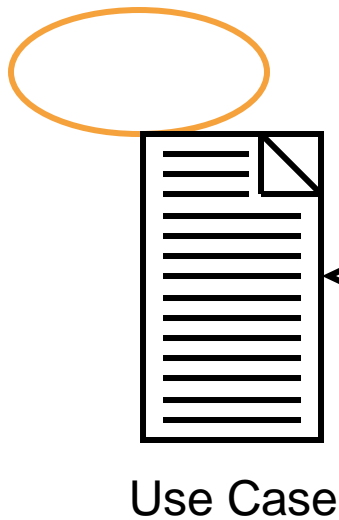
“Mastering Object-Oriented Analysis and Design with UML 2.0”

IBM Software Group

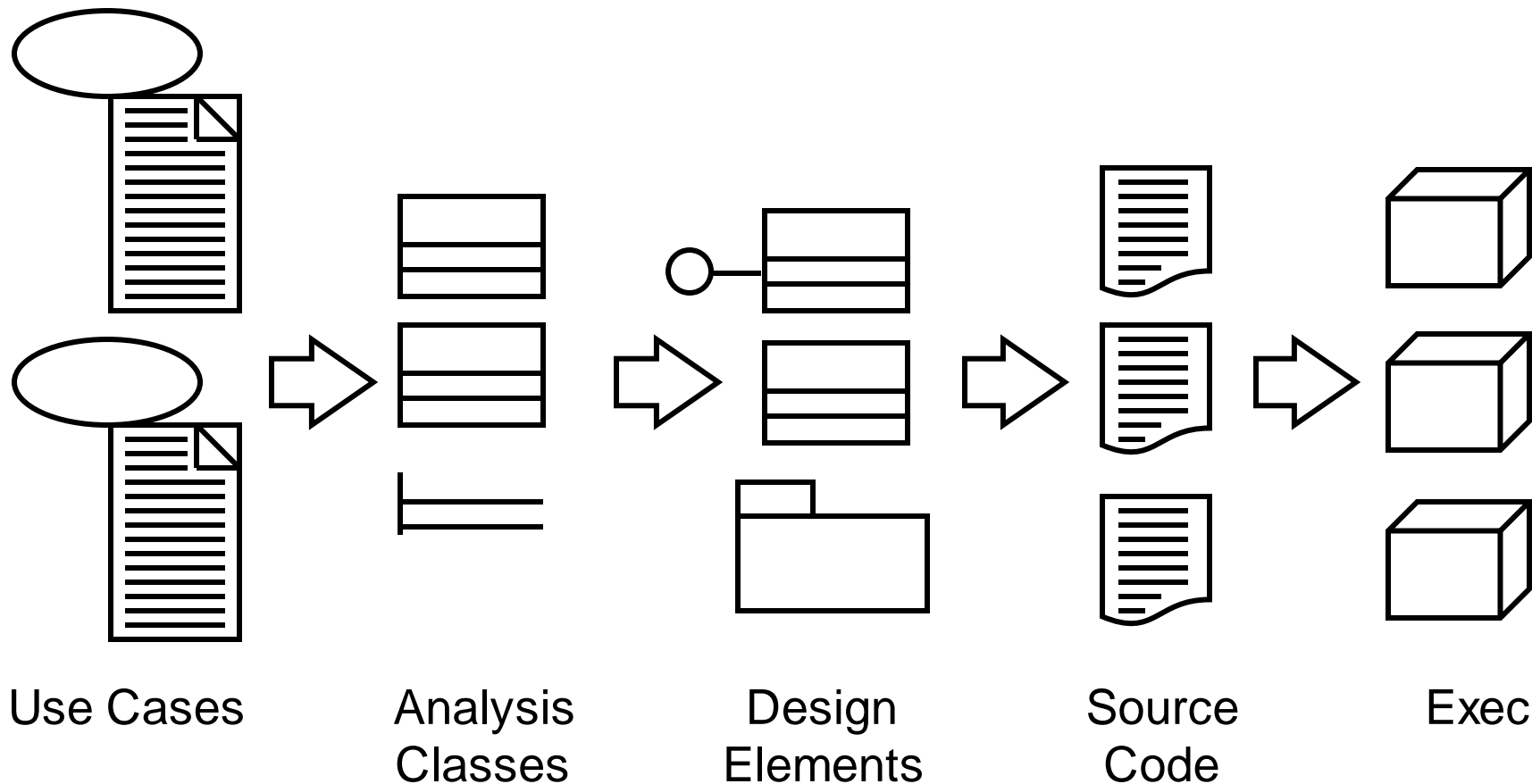
Use-Case Realization

Use-Case Model

Design Model



Analysis Classes: A First Step Toward Executables



Use Cases

Analysis
Classes

Design
Elements

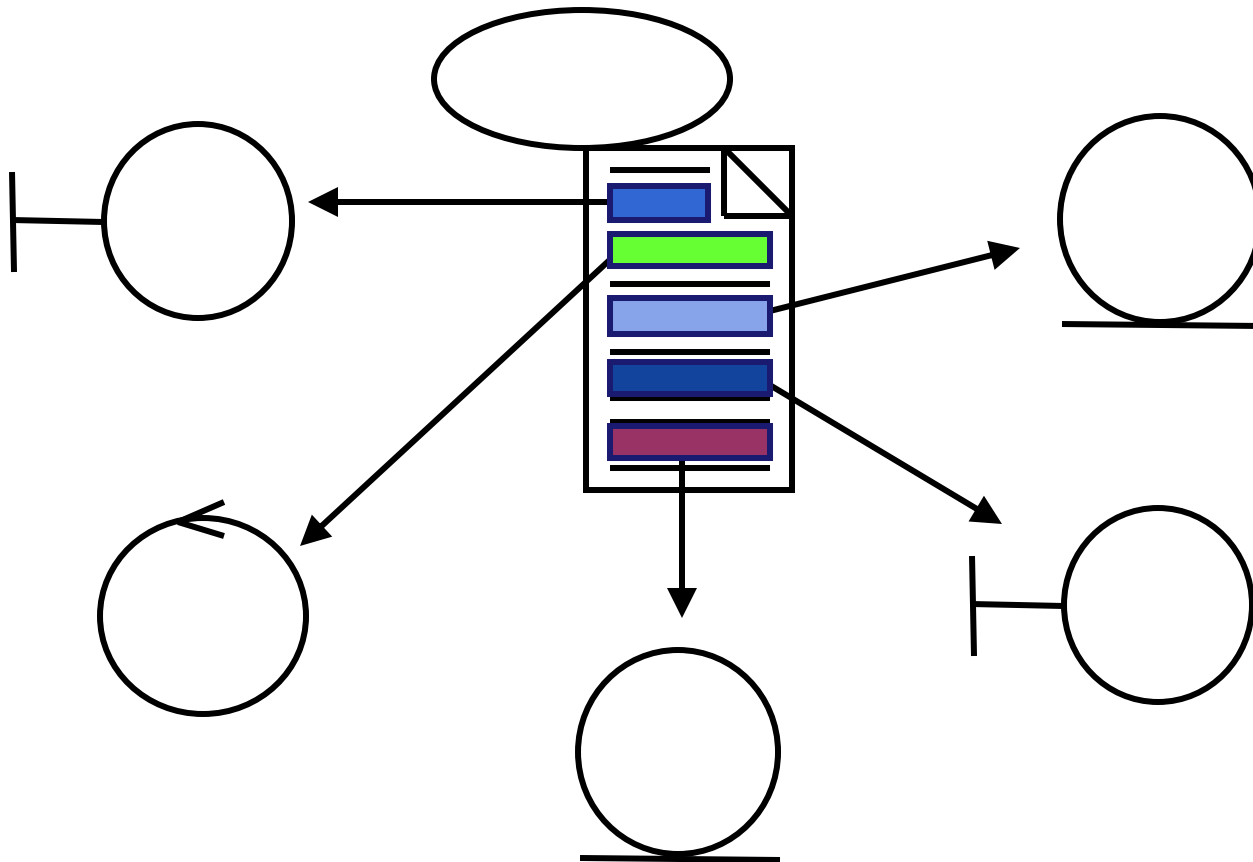
Source
Code

Exec

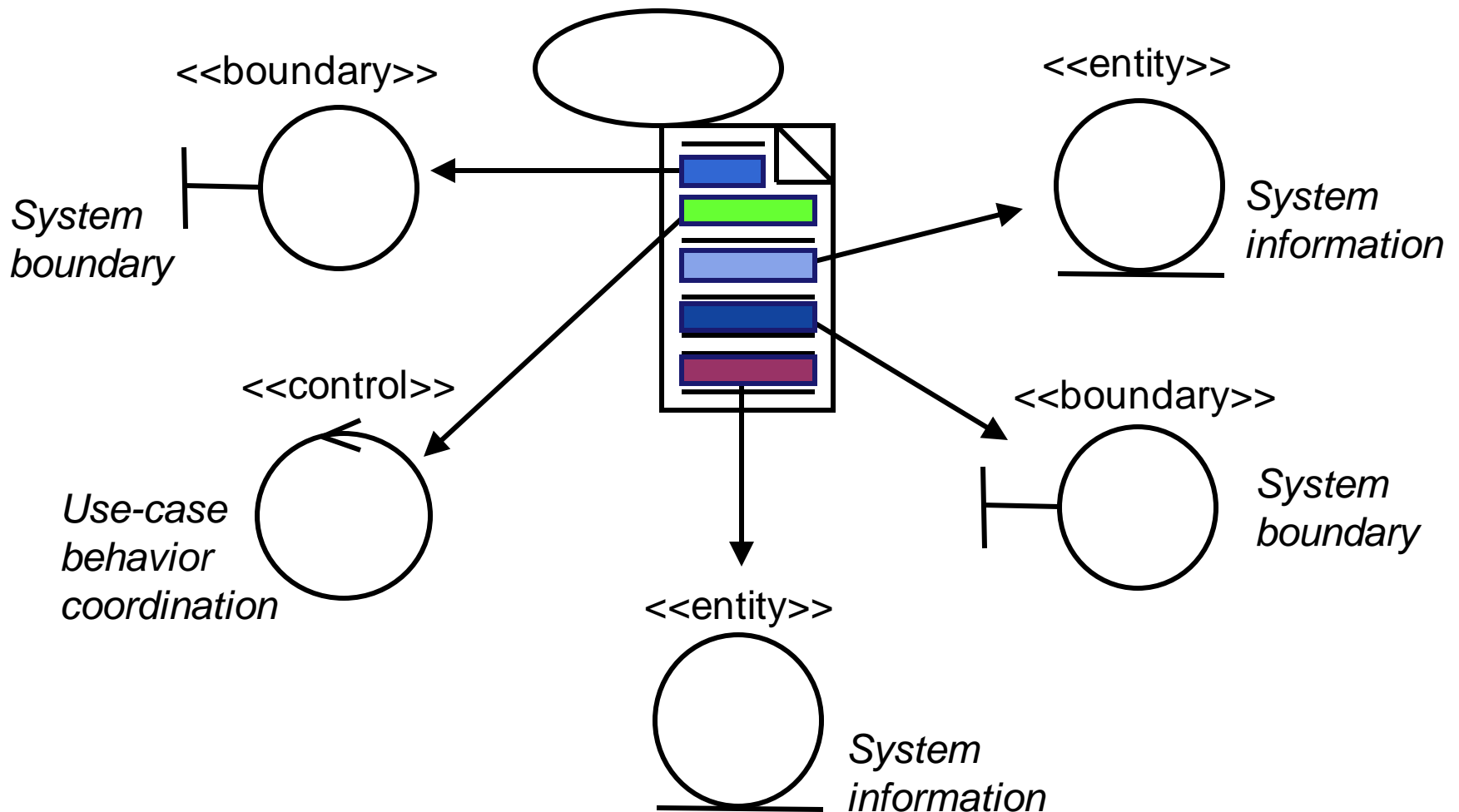
Use-Case Analysis

Find Classes from Use-Case Behavior

- ❖ The complete behavior of a use case has to be distributed to analysis classes

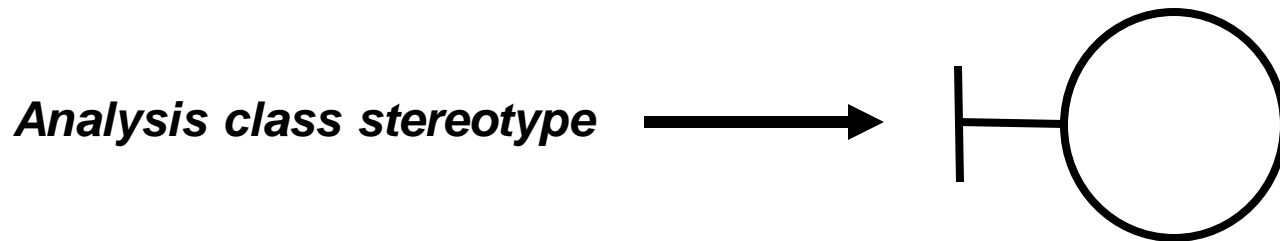


What Is an Analysis Class?



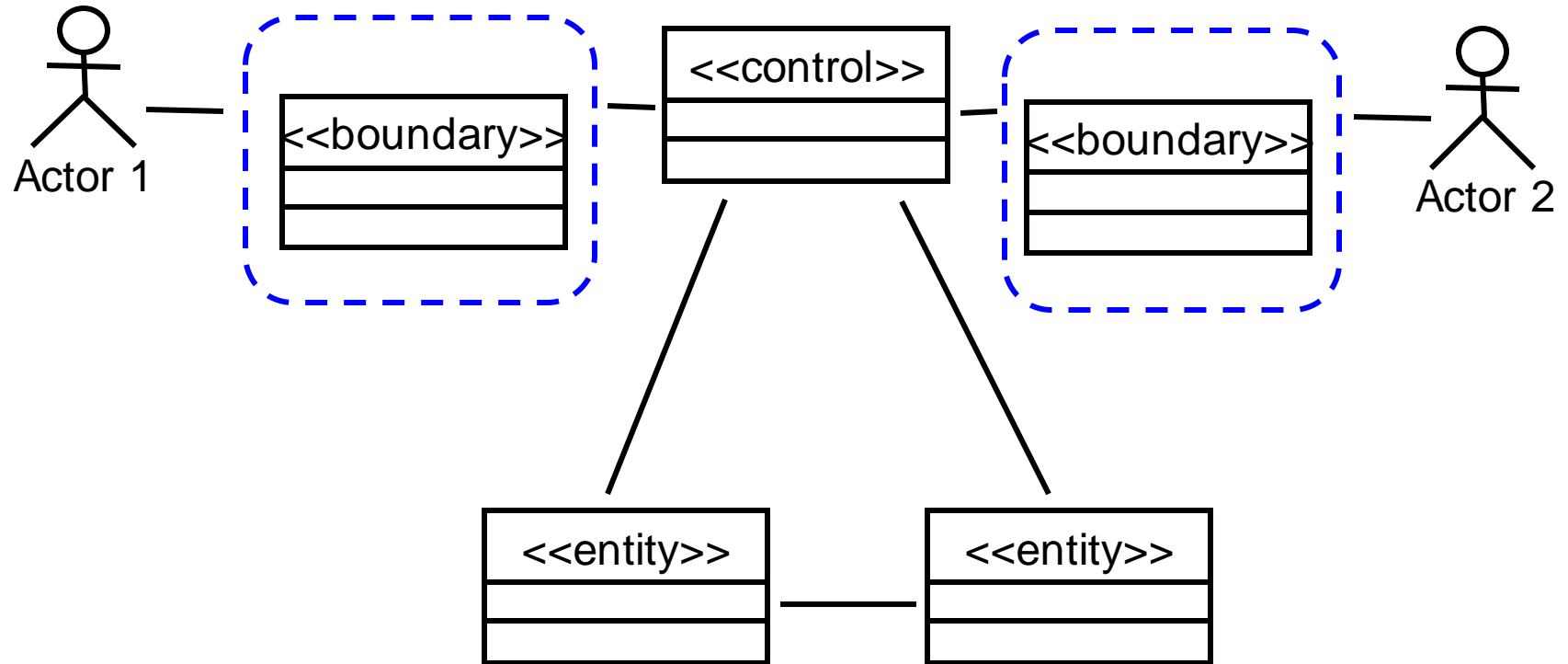
What Is a Boundary Class?

- ❖ Intermediates between the interface and something outside the system
- ❖ Several Types
 - User interface classes
 - System interface classes
 - Device interface classes
- ❖ *One boundary class per actor/use-case pair*



Environment dependent.

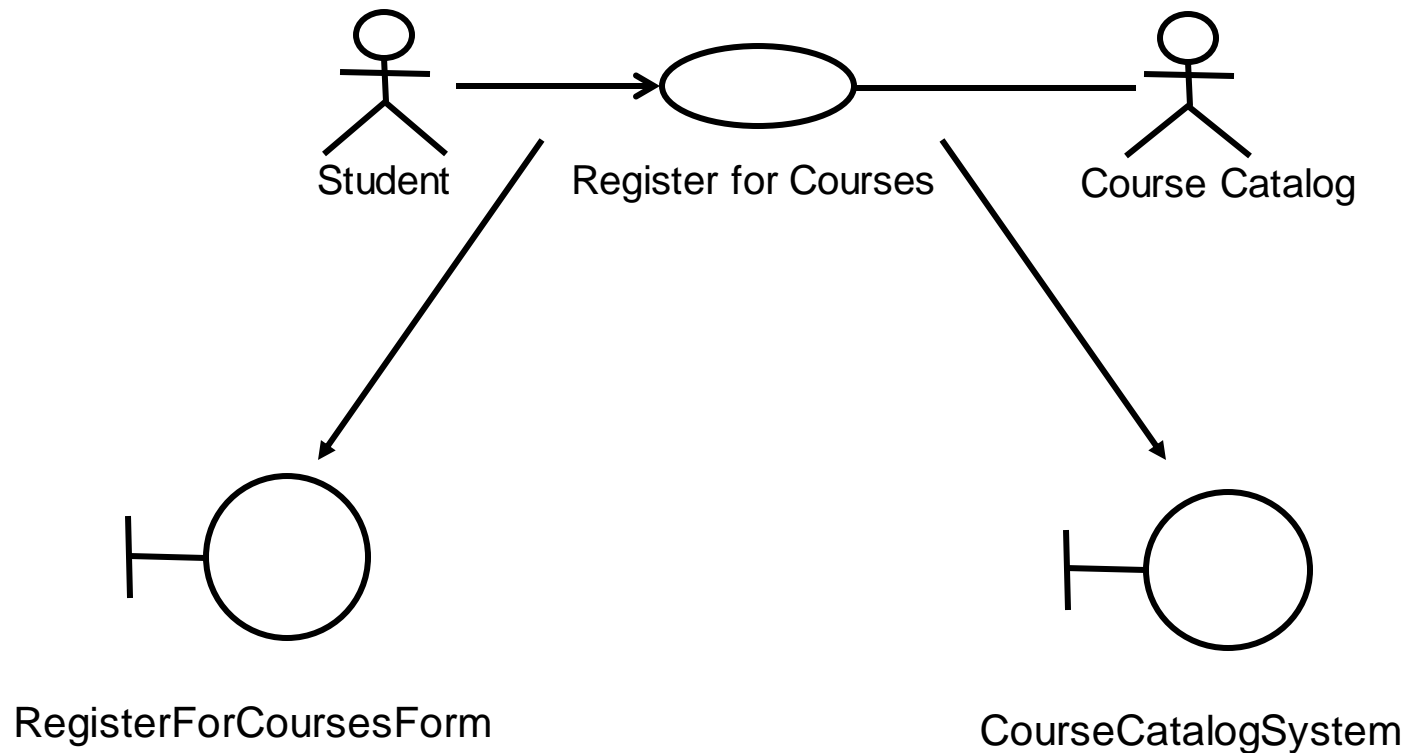
The Role of a Boundary Class



Model interaction between the system and its environment.

Example: Finding Boundary Classes

- ❖ One boundary class per actor/use case pair





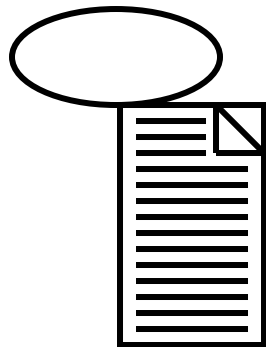
Guidelines: Boundary Class

- ❖ User Interface Classes
 - Concentrate on what information is presented to the user
 - Do NOT concentrate on the UI details
- ❖ System and Device Interface Classes
 - Concentrate on what protocols must be defined
 - Do NOT concentrate on how the protocols will be implemented

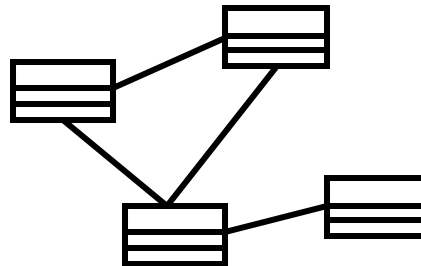
Concentrate on the responsibilities, not the details!

What Is an Entity Class?

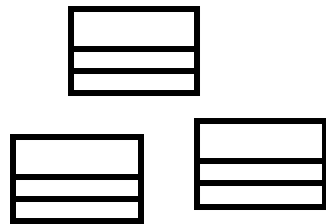
❖ Key abstractions of the system



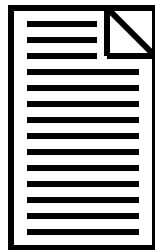
Use Case



Business-Domain
Model



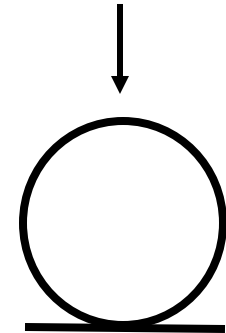
Architectural Analysis
Abstractions



Glossary

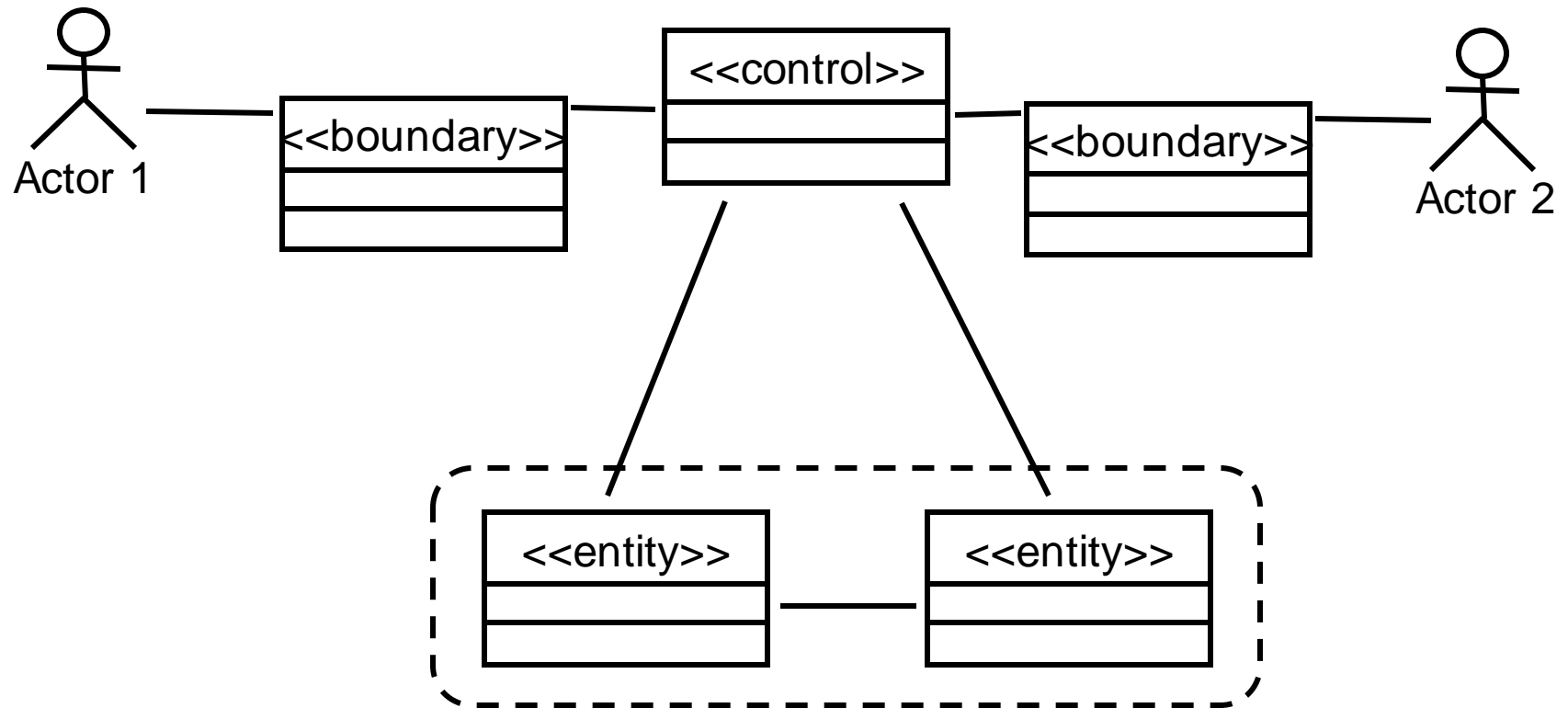


*Analysis class
stereotype*



Environment independent.

The Role of an Entity Class



Store and manage information in the system.



Example: Finding Entity Classes

- ❖ Use use-case flow of events as input
- ❖ Key abstractions of the use case
- ❖ Traditional, filtering nouns approach
 - Underline noun clauses in the use-case flow of events
 - Remove redundant candidates
 - Remove vague candidates
 - Remove actors (out of scope)
 - Remove implementation constructs
 - Remove attributes (save for later)
 - Remove operations

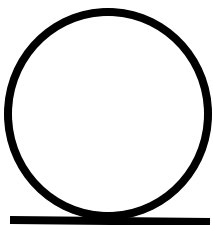
Example: Candidate Entity Classes

❖ Register for Courses (Create Schedule)



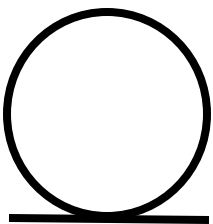
CourseOffering

The diagram shows a circle representing an entity class, with a horizontal line segment positioned directly beneath its bottom edge.



Schedule

The diagram shows a circle representing an entity class, with a horizontal line segment positioned directly beneath its bottom edge.

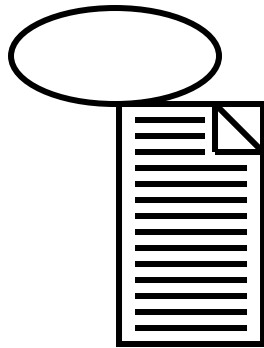


Student

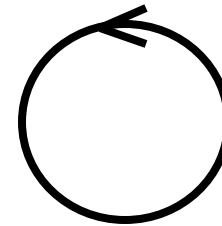
The diagram shows a circle representing an entity class, with a horizontal line segment positioned directly beneath its bottom edge.

What Is a Control Class?

- ❖ Use-case behavior coordinator
 - More complex use cases generally require one or more control cases



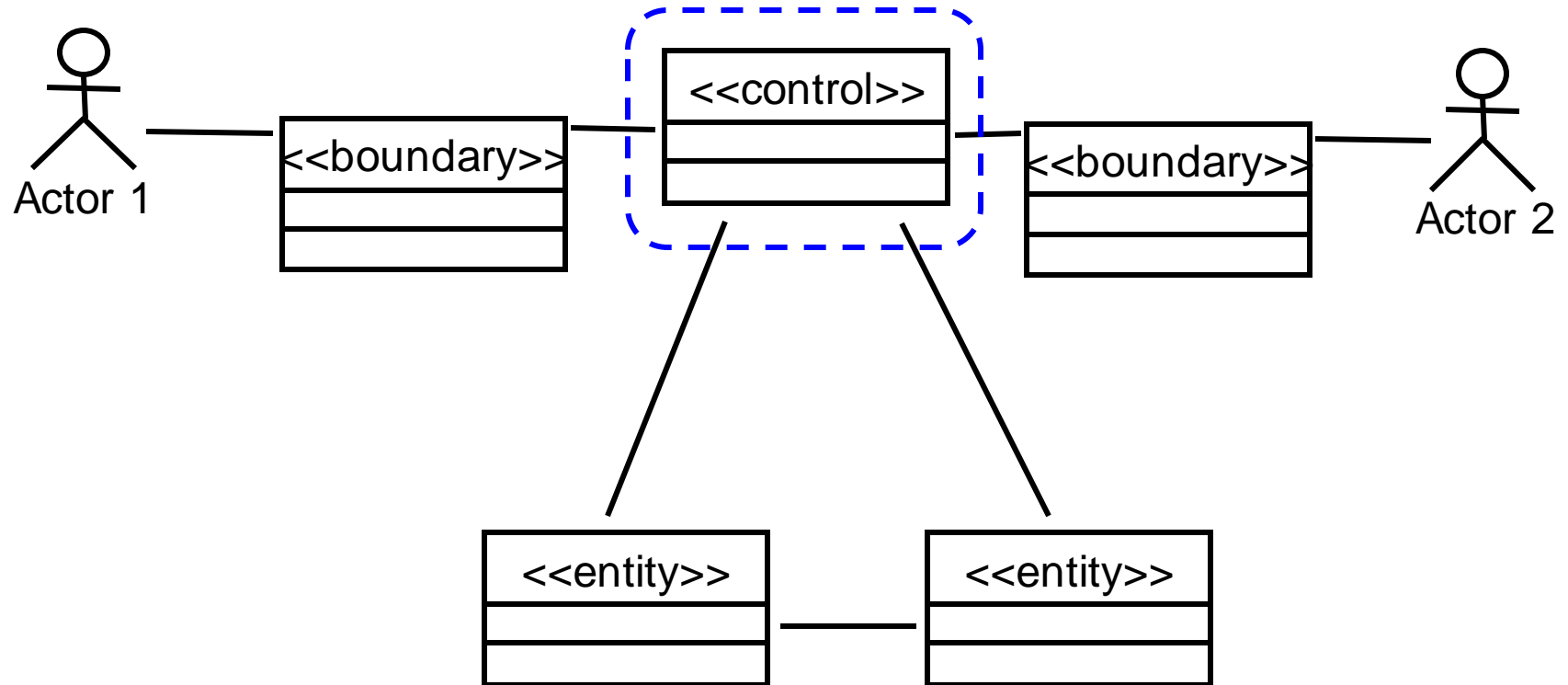
Use Case



*Analysis class
stereotype*

Use-case dependent. Environment independent.

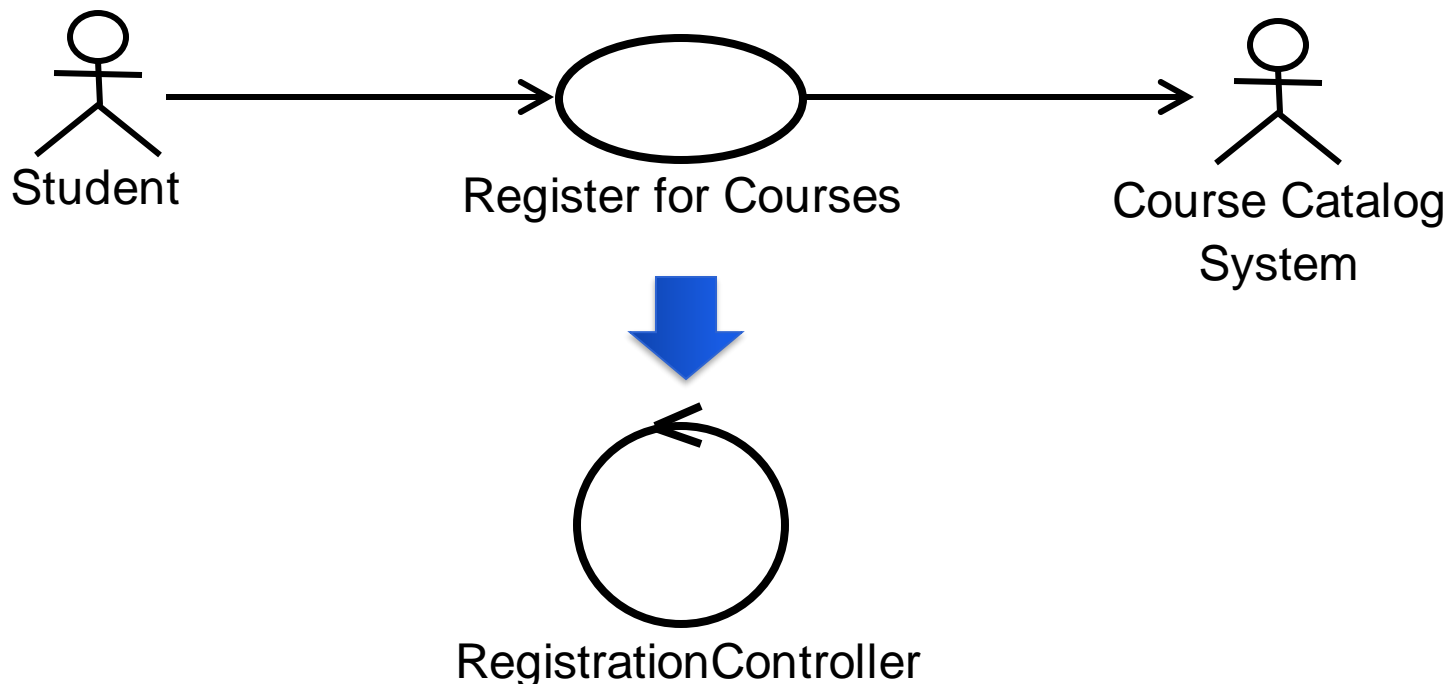
The Role of a Control Class



Coordinate the use-case behavior.

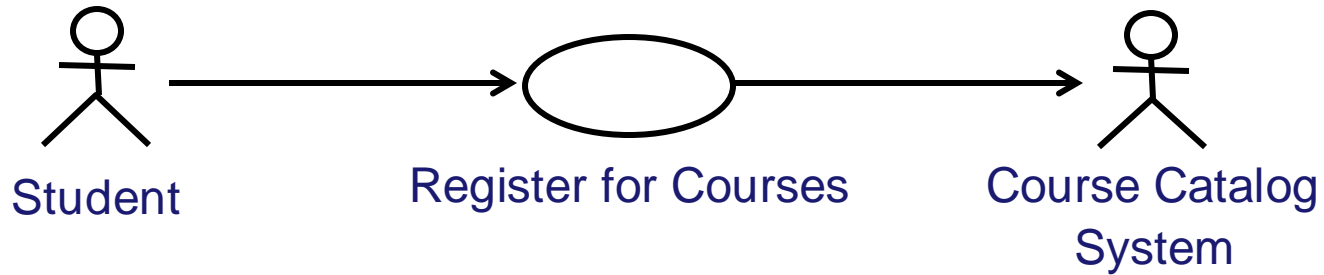
Example: Finding Control Classes

- ❖ In general, identify one control class per use case.
 - As analysis continues, a complex use case's control class may evolve into more than one class



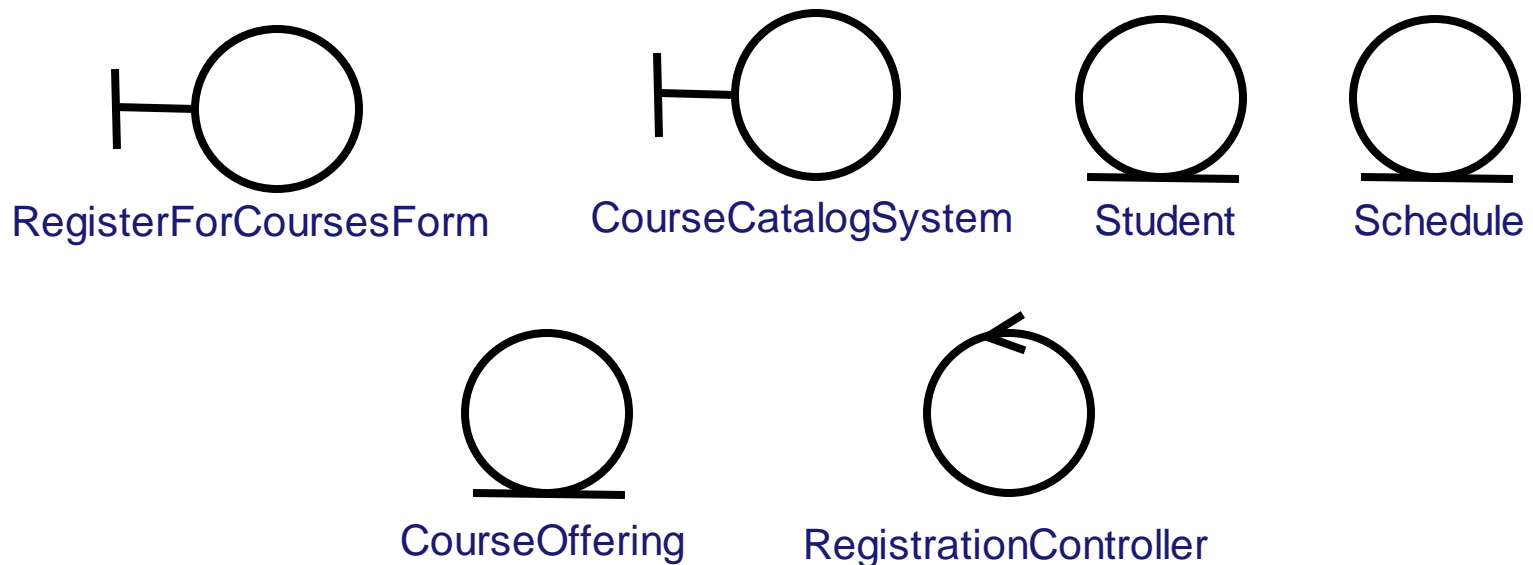


Example: Summary: Analysis Classes



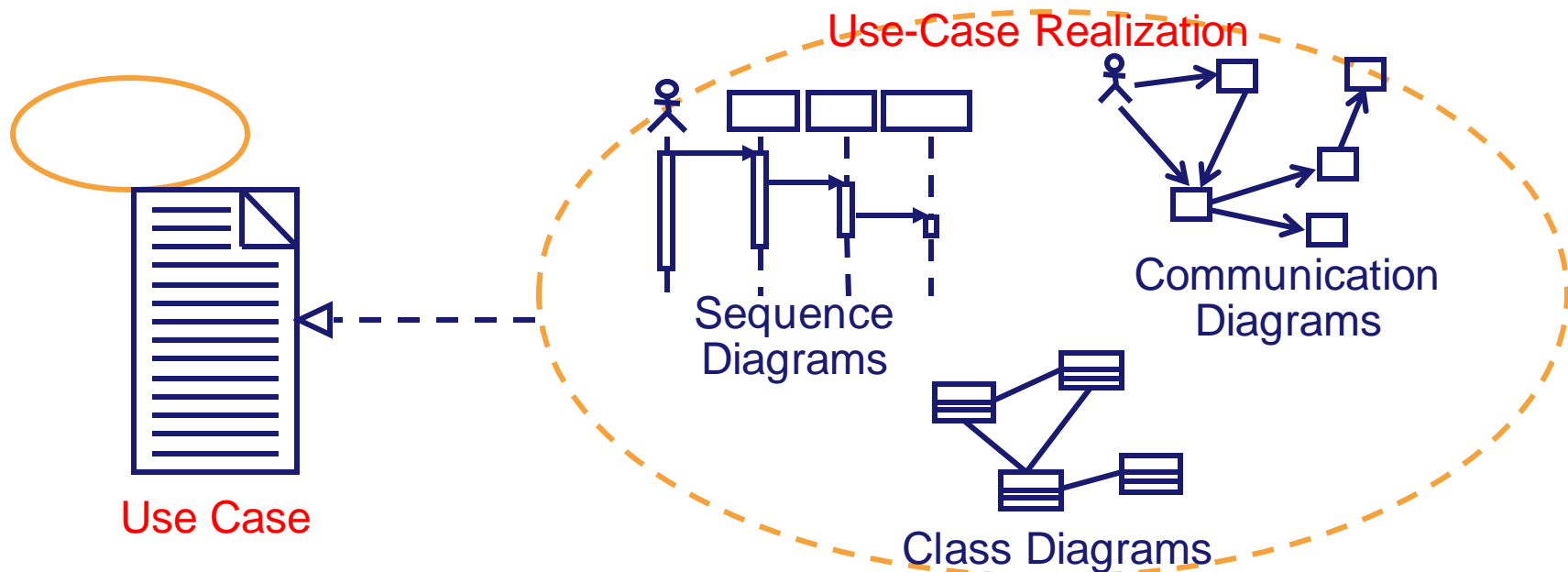
Use-Case Model

Design Model



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

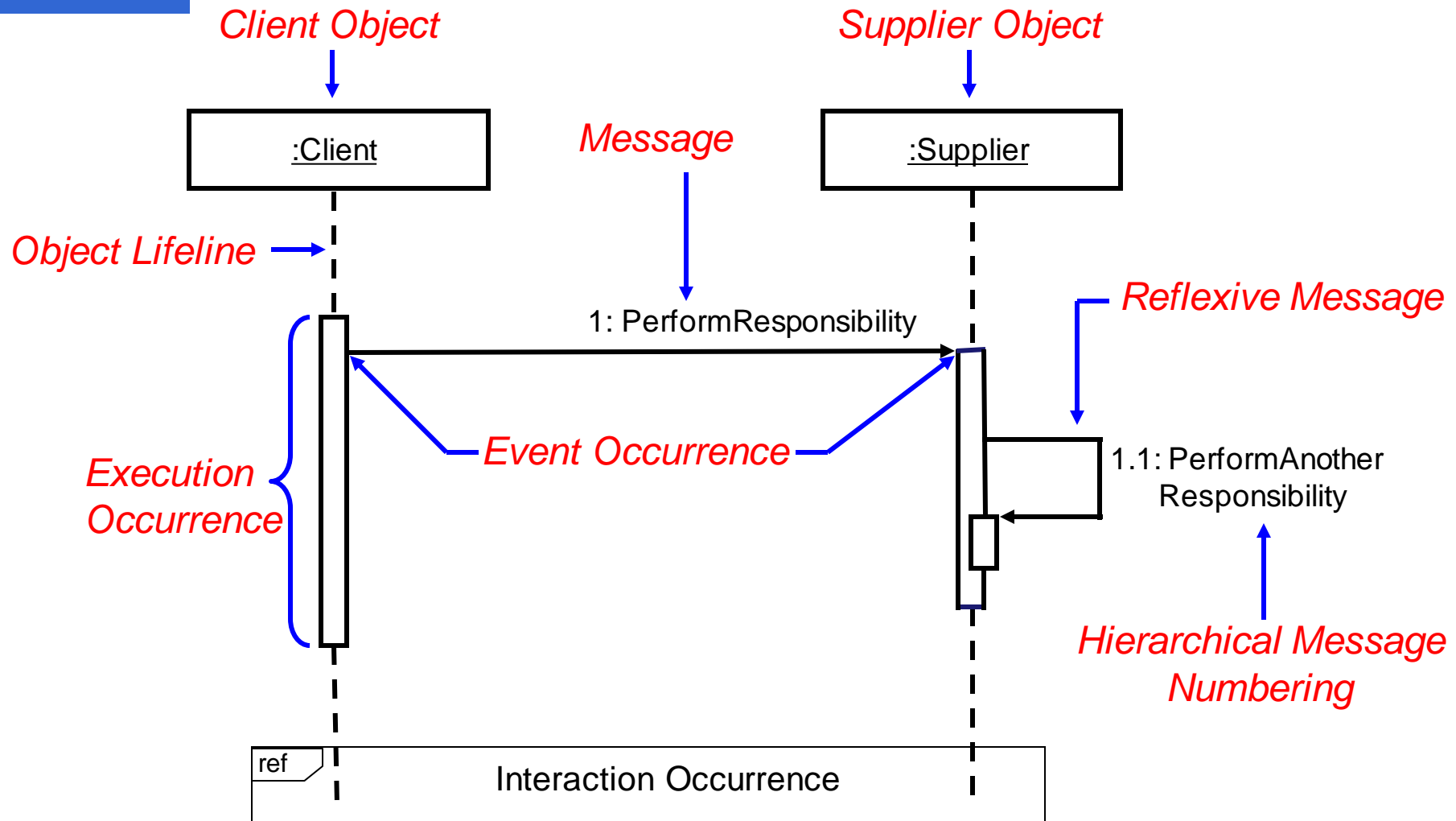
- ❖ Use analysis class stereotypes as a guide
 - Boundary Classes
 - Behavior that involves communication with an actor
 - Entity Classes
 - Behavior that involves the data encapsulated within the abstraction
 - Control Classes
 - Behavior specific to a use case or part of a very important flow of events



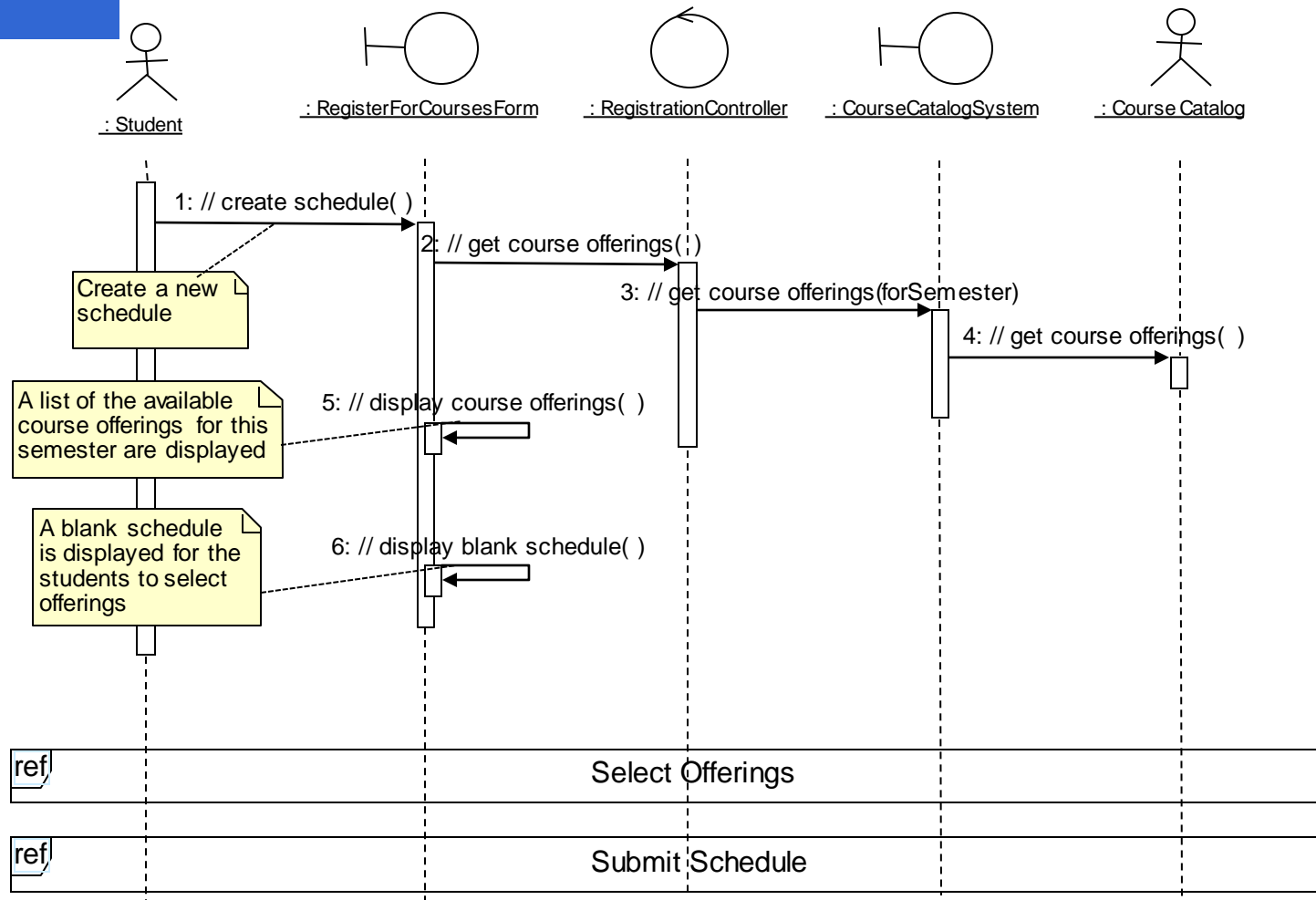
Guidelines: Allocating Responsibilities to Classes

- ❖ 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

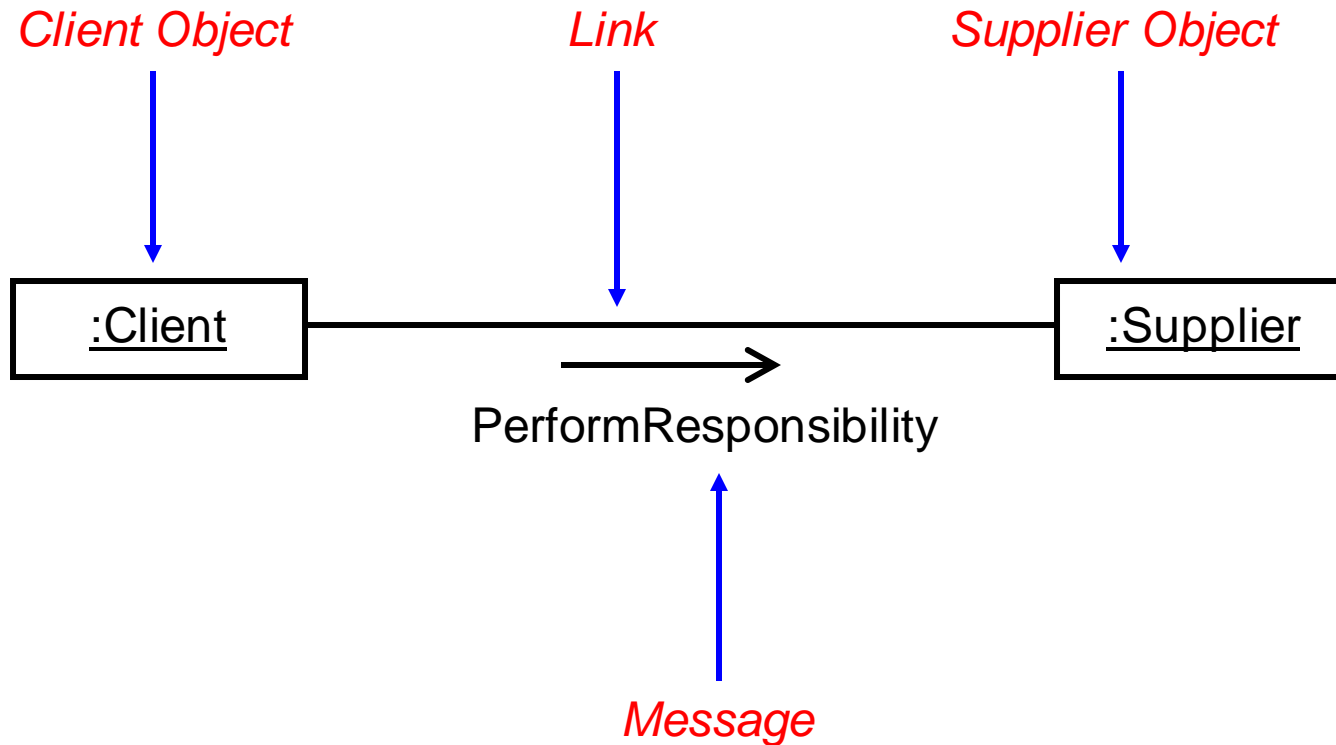
Anatomy of Sequence Diagrams



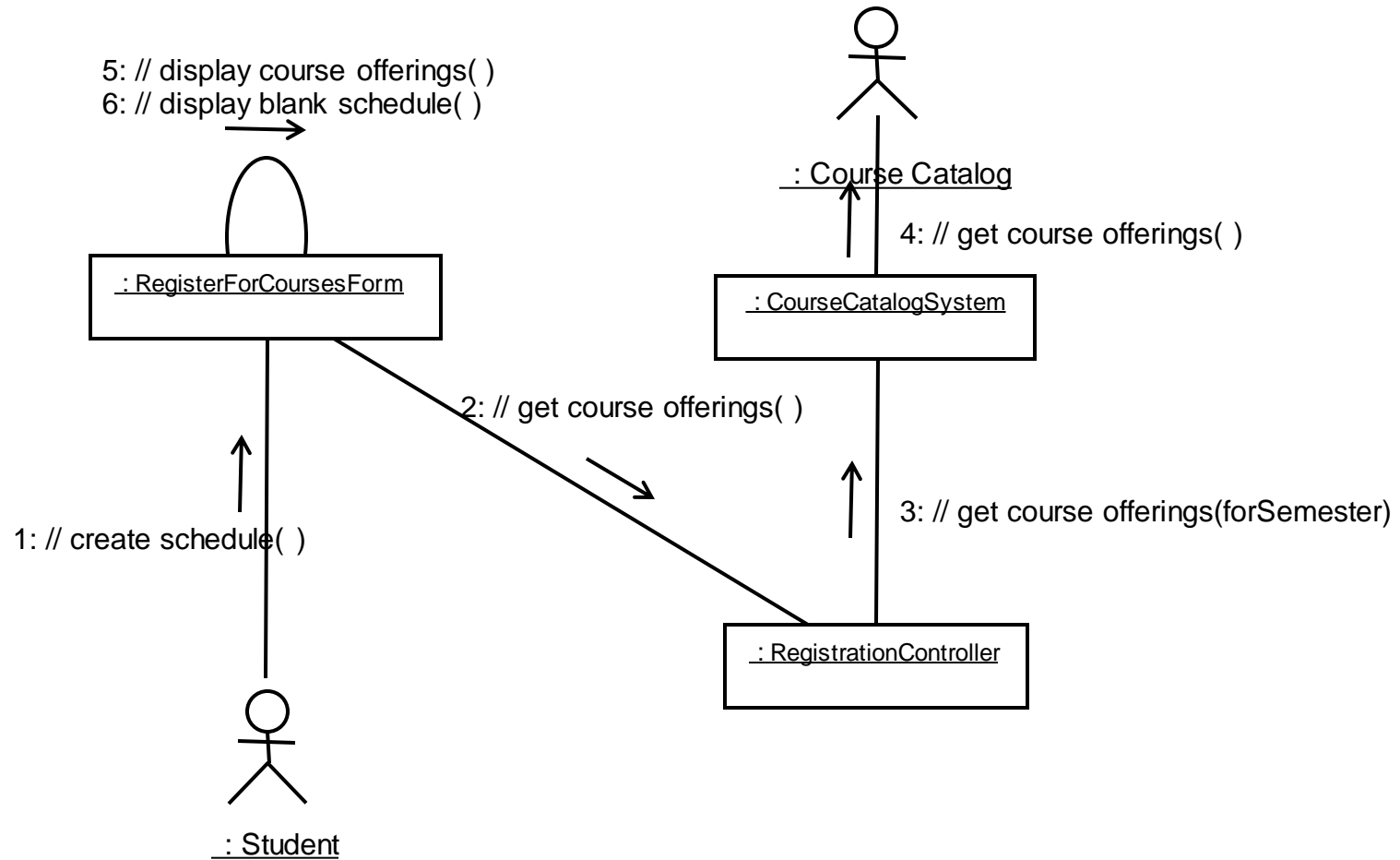
Example: Sequence Diagram



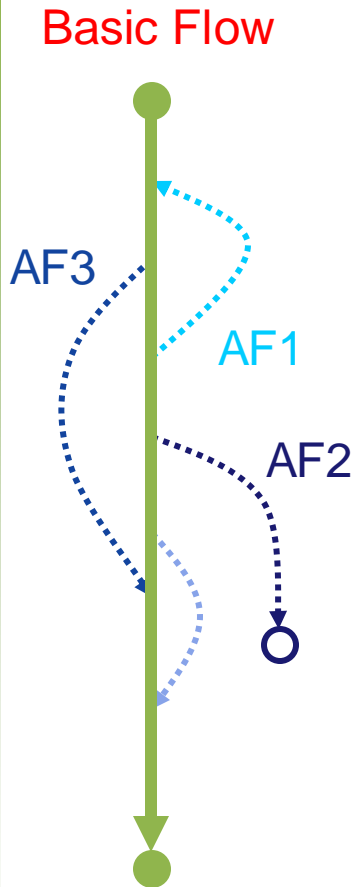
Anatomy of Communication Diagrams



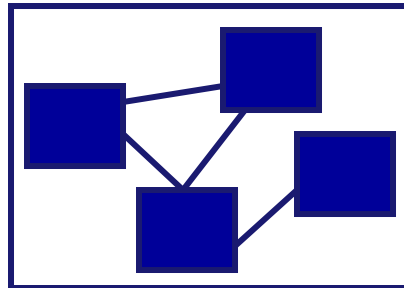
Example: Communication Diagram



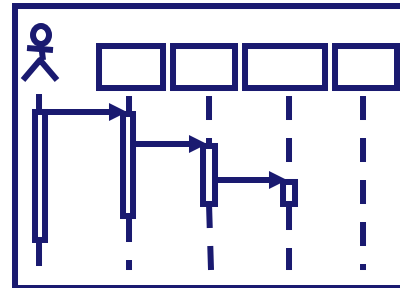
One Interaction Diagram: Not Good Enough



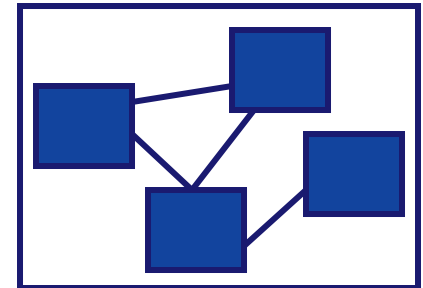
Alternate Flow 1



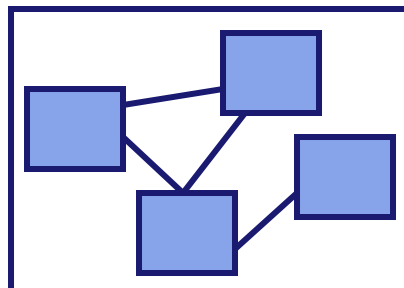
Alternate Flow 2



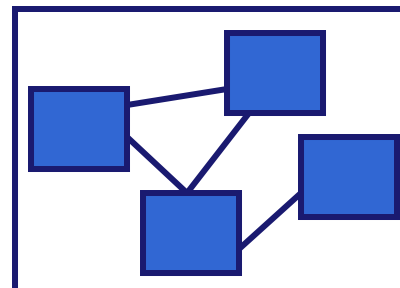
Alternate Flow 3



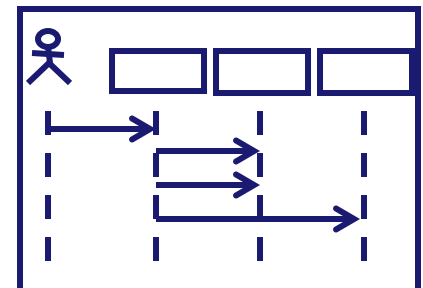
Alternate Flow 4



Alternate Flow 5



Alternate Flow n





Communication Diagrams vs. Sequence Diagrams

Communication Diagrams	Sequence Diagrams
<ul style="list-style-type: none">■ Show relationships in addition to interactions■ Better for visualizing patterns of collaboration■ Better for visualizing all of the effects on a given object■ Easier to use for brainstorming sessions	<ul style="list-style-type: none">■ Show the explicit sequence of messages■ Better for visualizing overall flow■ Better for real-time specifications and for complex scenarios