

Unified Modeling Language (UML) Part 1

Dr. Krishnendu Guha

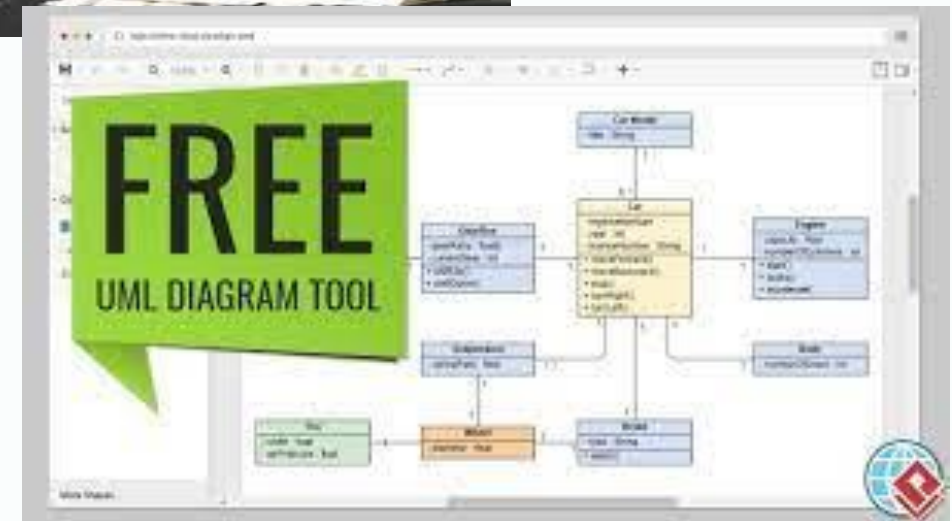
Assistant Professor
(On Contract)

National Institute of Technology
(NIT), Jamshedpur

Email: krishnendu.ca@nitjsr.ac.in

Introduction

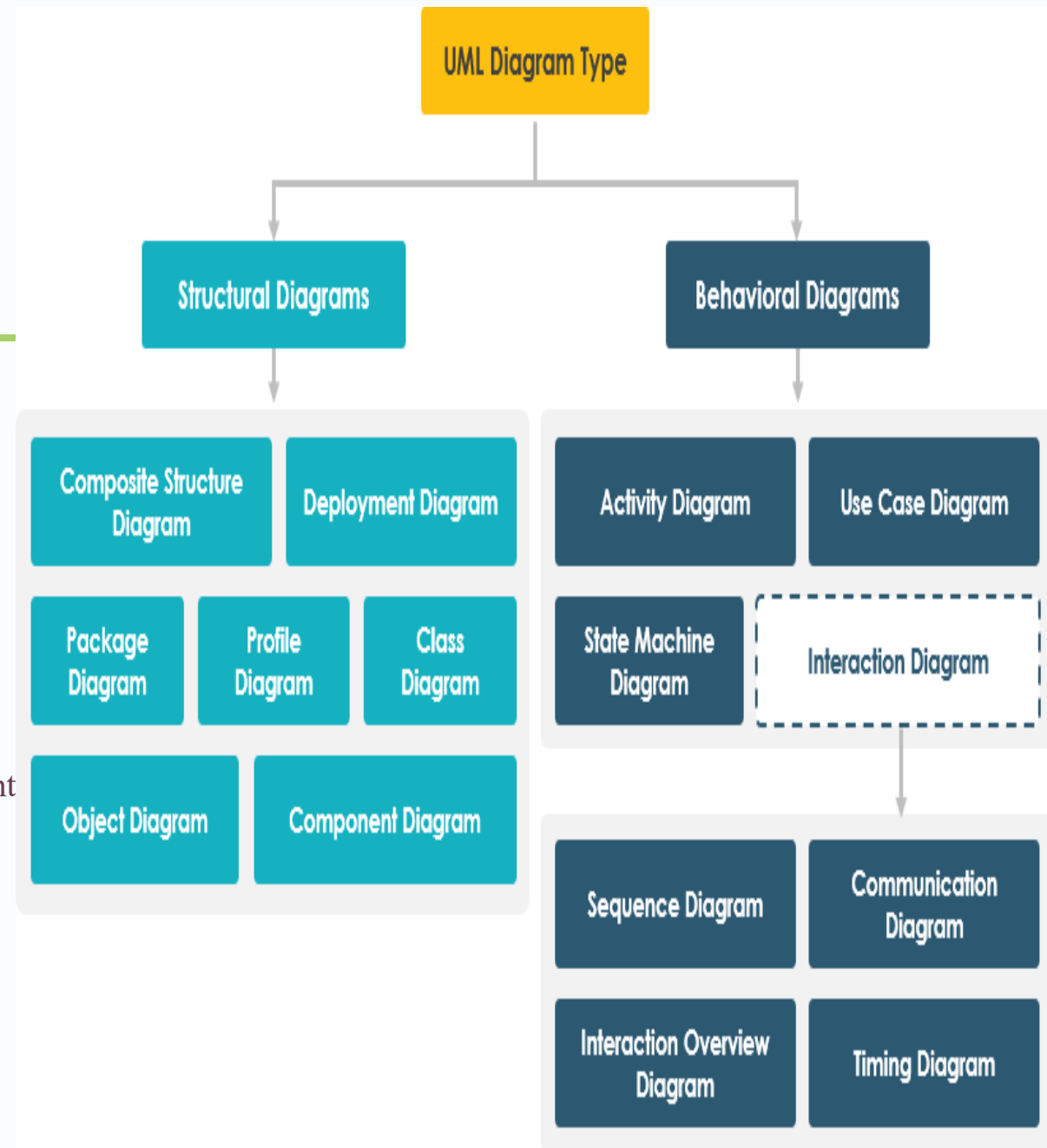
- UML provides a standardized set of tools to document the analysis and design of a software system.
- The UML toolset includes diagrams that allow people to visualize the construction of an object-oriented system
- This is similar to the way a set of blueprints allows people to visualize the construction of a building.
- Whether you are working independently or with a large systems development team, the documentation that you create with UML provides an effective means of communication between the development team and the business team on a project.



- UML consists of things, relationships, and diagrams
- The first components, or primary elements, of UML are called things or object, but in UML they are called things
- Structural things are most common.
- Structural things are classes, interfaces, use cases, and many other elements that provide a way to create models.
- Structural things allow the user to describe relationships.
- Behavioral things describe how things work.
- Examples of behavioral things are interactions and state machines.
- Group things are used to define boundaries. An example of a group thing is a package.
- Finally, we have annotational things, so that we can add notes to the diagrams.

UML Category	UML Elements	Specific UML Details
Things	Structural Things	Classes Interfaces Collaborations Use Cases Active Classes Components Nodes
	Behavioral Things	Interactions State Machines
	Grouping Things	Packages
	Annotational Things	Notes
Relationships	Structural Relationships	Dependencies Aggregations Associations Generalizations
	Behavioral Relationships	Communicates Includes Extends Generalizes
Diagrams	Structural Diagrams	Class Diagrams Component Diagrams Deployment Diagrams
	Behavioral Diagrams	Use Case Diagrams Sequence Diagrams Communication Diagrams Statechart Diagrams Activity Diagrams

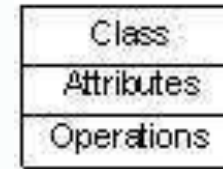
- Relationships are the glue that holds the things together.
 - It is useful to think of relationships in two ways.
 - Structural relationships are used to tie the things together in the structural diagrams.
 - Structural relationships include dependencies, aggregations, associations, and generalizations.
 - Structural relationships show inheritance, for example.
 - Behavioral relationships are used in the behavioral diagrams.
 - The four basic types of behavioral relationships are communicates, includes, extends, and generalizes.
-
- There are two main types of diagrams in UML: structural diagrams and behavioral diagrams.
 - Structural diagrams are used, for example, to describe the relationships between classes. They include class diagrams, object diagrams, component diagrams, and deployment diagrams.
 - Behavioral diagrams, on the other hand, can be used to describe the interaction between people (called actors in UML) and the thing we refer to as a use case, or how the actors use the system.
 - Behavioral diagrams include use case diagrams, sequence diagrams, communication diagrams, state chart diagrams, and activity diagrams.



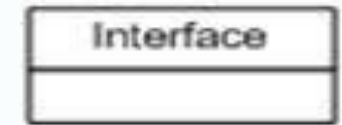
Structural Things

Structural things define the static part of the model. They represent the physical and conceptual elements. Following are the brief descriptions of the structural things.

Class – Class represents a set of objects having similar responsibilities.



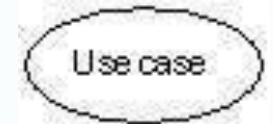
Interface – Interface defines a set of operations, which specify the responsibility of a class.



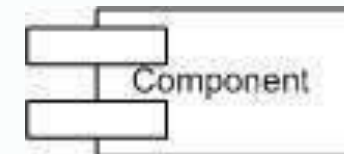
Collaboration – Collaboration defines an interaction between elements.



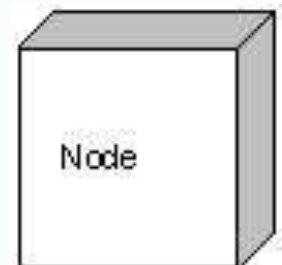
Use case – Use case represents a set of actions performed by a system for a specific goal.



Component – Component describes the physical part of a system.



Node – A node can be defined as a physical element that exists at run time.



Behavioral Things

A **behavioral thing** consists of the dynamic parts of UML models. Following are the behavioral things –

Interaction – Interaction is defined as a behavior that consists of a group of messages exchanged among elements to accomplish a specific task.

State machine – State machine is useful when the state of an object in its life cycle is important. It defines the sequence of states an object goes through in response to events. Events are external factors responsible for state change

Grouping Things

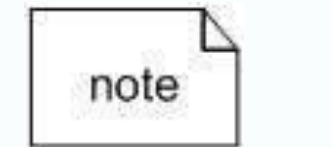
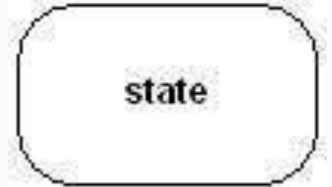
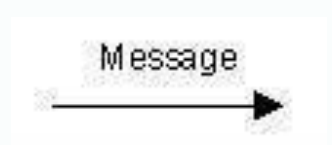
Grouping things can be defined as a mechanism to group elements of a UML model together.

There is only one grouping thing available –

Package – Package is the only one grouping thing available for gathering structural and behavioral things.

Annotational Things

Annotational things can be defined as a mechanism to capture remarks, descriptions, and comments of UML model elements. **Note** - It is the only one Annotational thing available. A note is used to render comments, constraints, etc. of an UML element.

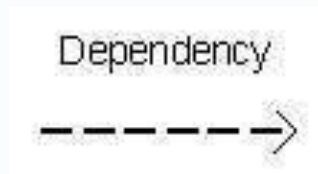


Relationship

Relationship is another most important building block of UML. It shows how the elements are associated with each other and this association describes the functionality of an application. There are four kinds of relationships available.

Dependency

Dependency is a relationship between two things in which change in one element also affects the other.



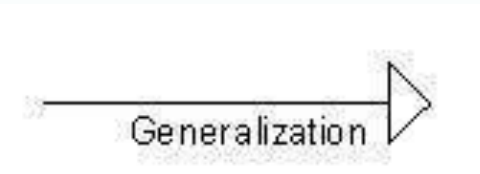
Association

Association is basically a set of links that connects the elements of a UML model. It also describes how many objects are taking part in that relationship.



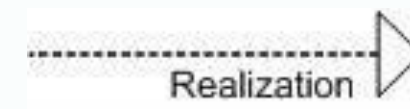
Generalization

Generalization can be defined as a relationship which connects a specialized element with a generalized element. It basically describes the inheritance relationship in the world of objects.

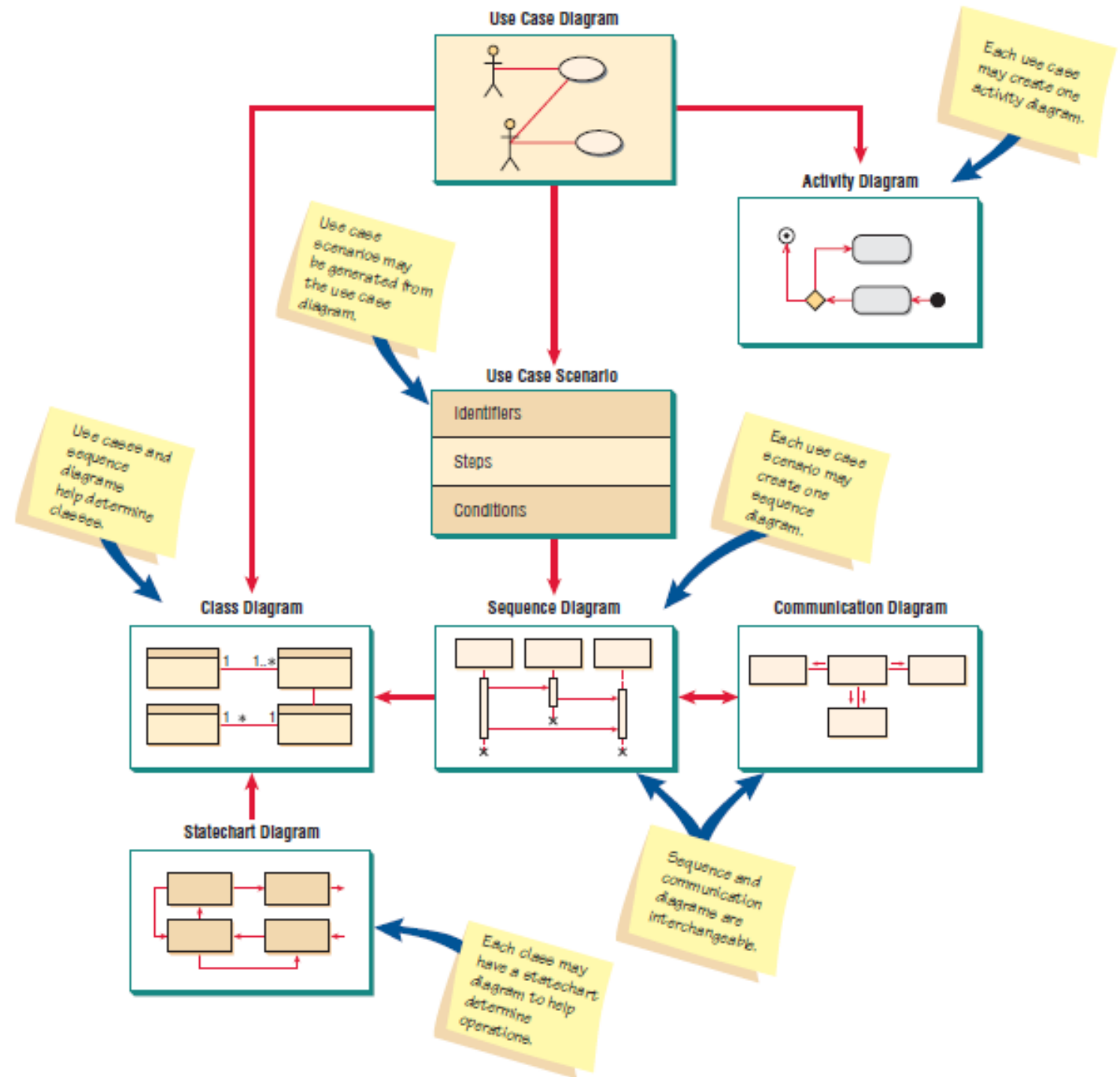


Realization

Realization can be defined as a relationship in which two elements are connected. One element describes some responsibility, which is not implemented and the other one implements them. This relationship exists in case of interfaces.

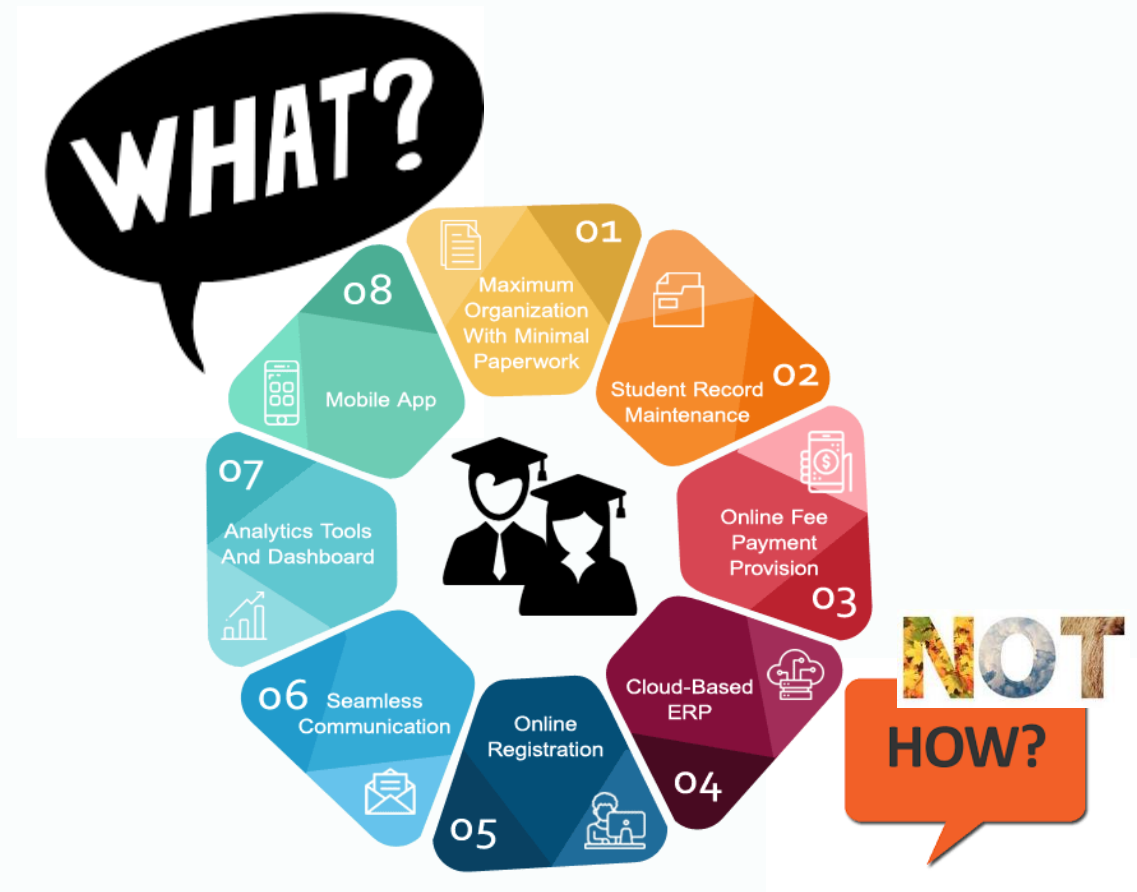


- The six most commonly used UML diagrams are:
- **1.** A use case diagram, describing how the system is used. Analysts start with a use case diagram.
- **2.** A use case scenario (although technically it is not a diagram). This scenario is a verbal articulation of exceptions to the main behavior described by the primary use case.
- **3.** An activity diagram, illustrating the overall flow of activities. Each use case may create one activity diagram.
- **4.** Sequence diagrams, showing the sequence of activities and class relationships. Each use case may create one or more sequence diagrams. An alternative to a sequence diagram is a communication diagram, which contains the same information but emphasizes communication instead of timing.
- **5.** Class diagrams, showing the classes and relationships. Sequence diagrams are used to determine classes. An offshoot of a class diagram is a gen/spec diagram (which stands for generalization/specialization).
- **6.** Statechart diagrams, showing the state transitions. Each class may create a statechart diagram, which is useful for determining class methods.

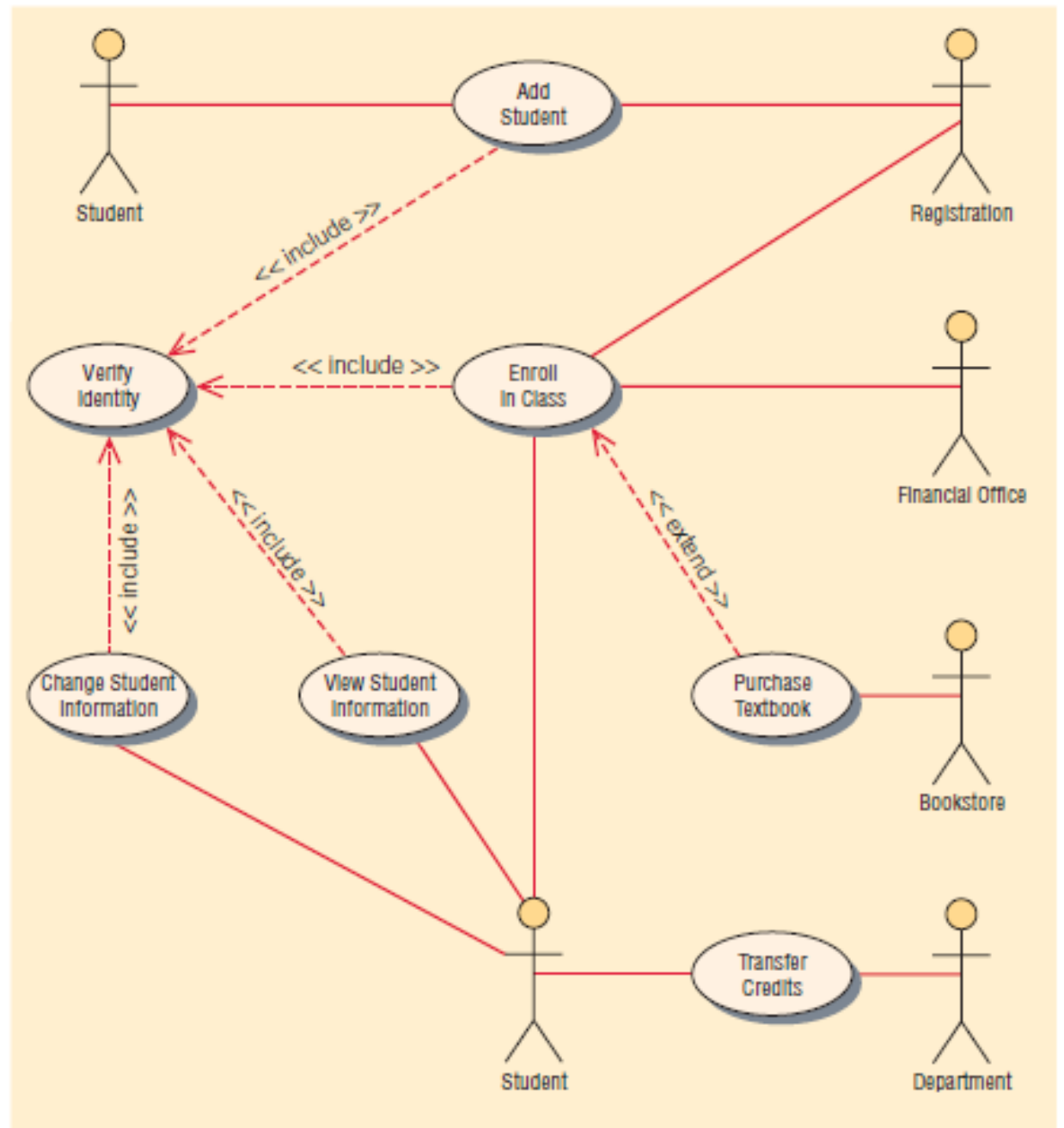


Use Case Modeling

- UML is fundamentally based on an object-oriented analysis technique known as use case modeling
- A use case model shows a view of the system from the user perspective, thus describing *what* a system does without describing *how* the system does it.
- UML can be used to analyze the use case model, and to derive system objects and their interactions with each other and with the users of the system.
- Using UML techniques, you further analyze the objects and their interactions to derive object behavior, attributes, and relationships.
- A use case provides developers with a view of what the users want.
- It is free of technical or implementation details.
- We can think of a use case as a sequence of transactions in a system. The use case model is based on the interactions and relationships of individual use cases.



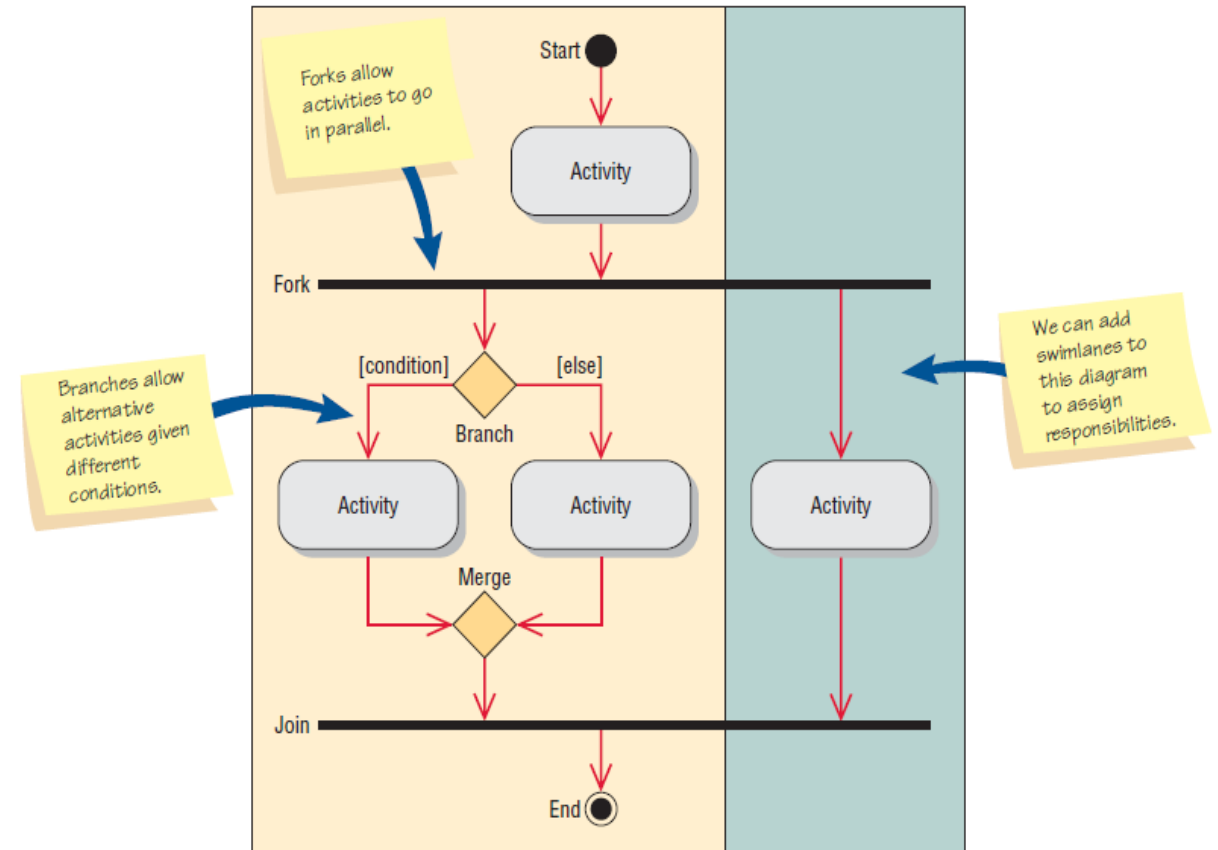
- A use case always describes three things:
 - an actor that initiates an event;
 - the event that triggers a use case; and
 - the use case that performs the actions triggered by the event.
-
- In a use case, an actor using the system initiates an event that begins a related series of interactions in the system.
 - Use cases are used to document a single transaction or event.
 - An event is an input to the system that happens at a specific time and place and causes the system to do something.
 - The three main areas are:
 1. A header area containing case identifiers and initiators.
 2. Steps performed.
 3. A footer area containing preconditions, assumptions, questions, and other information



use case example of student enrollment at a university

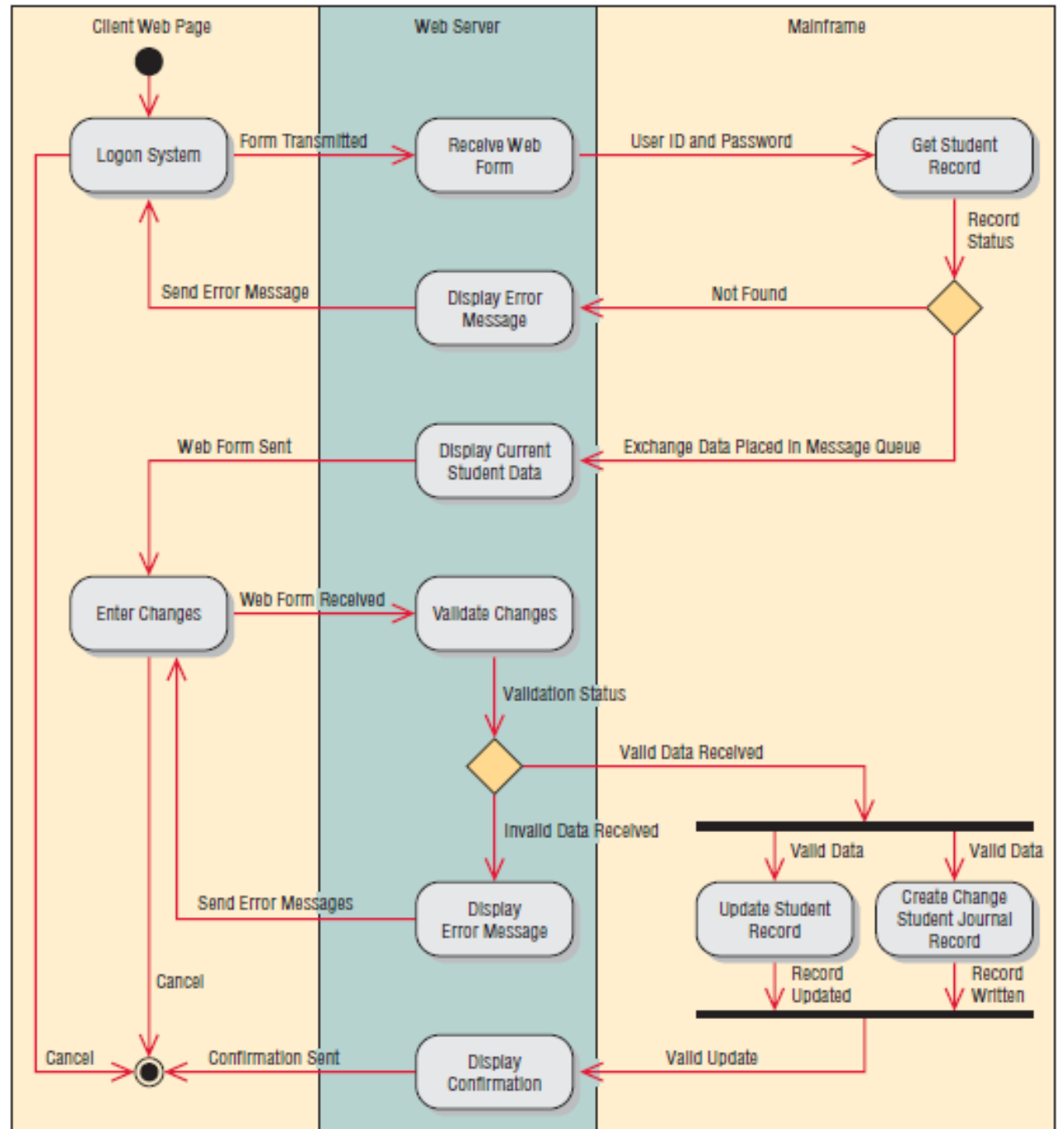
Activity Diagrams

- Activity diagrams show the sequence of activities in a process, including sequential and parallel activities, and decisions that are made.
- An activity diagram is usually created for one use case and may show the different possible scenarios.
- The symbols on an activity diagram are
 - A rectangle with rounded ends represents an activity, either a manual one, such as signing a legal document, or an automated one, such as a method or program.
 - An arrow represents an event. Events represent things that happen at a certain time and place.
 - A diamond represents either a decision (also called a branch) or a merge.



- A long, flat rectangle represents a synchronization bar. These are used to show parallel activities, and may have one event going into the synchronization bar and several events going out of it, called a fork.
- A synchronization in which several events merge into one event is called a join.
- There are two symbols that show the start and end of the diagram.
- The initial state is shown as a filled-in circle.
- The final state is shown as a black circle surrounded by a white circle.
- Rectangles surrounding other symbols, called swimlanes, indicate partitioning and are used to show which activities are done on which platform, such as a browser, server, or mainframe computer; or to show activities done by different user groups.
- Swimlanes are zones that can depict logic as well as the responsibility of a class

This activity diagram shows three swimlanes: Client Web Page, Web Server, and Mainframe.



Following diagram is drawn with the four main activities –

- Send order by the customer
- Receipt of the order
- Confirm the order
- Dispatch the order

After receiving the order request, condition checks are performed to check if it is normal or special order. After the type of order is identified, dispatch activity is performed and that is marked as the termination of the process.

