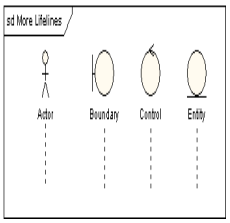
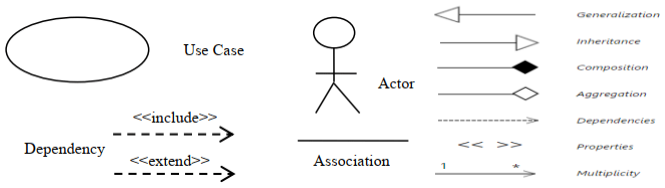
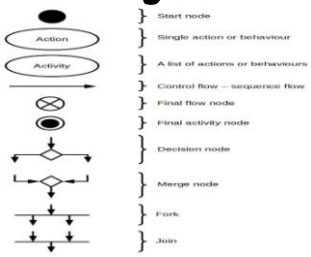
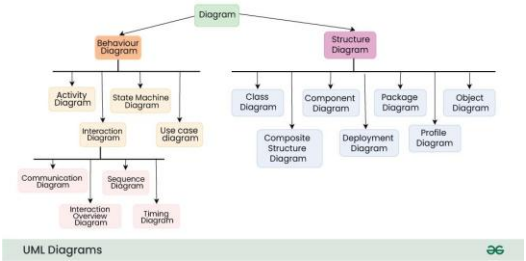




Software Engineering And Testing

Object Modeling using UML





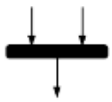
Activity Diagram

- Activity diagrams, along with use case and state machine diagrams, are considered behavior diagrams because they describe what must happen in the system being modeled.
- Activity diagram is defined as a UML diagram that focuses on the execution and flow of the behavior of a system instead of implementation.
- It is also called object-oriented flowchart.
- Illustrate a business process or workflow between users and the system.

How to draw Activity Diagram

- Activity diagram is a flowchart of activities.
- It represents the workflow between various system activities.
- Activity diagrams include swimlanes, branching, parallel flow, control nodes, expansion nodes, and object nodes.
- Activity diagram also supports exception handling.
- To draw an activity diagram, one must understand and explore the entire system.
- All the elements and entities that are going to be used inside the diagram must be known by the user.
- After analyzing all activities, these activities should be explored to find various constraints that are applied to activities.
- If there is such a constraint, then it should be noted before developing an activity diagram.

Activity Diagram Symbols

Symbol	Name	Description
	Start symbol	Represents the beginning of a process or workflow in an activity diagram. It can be used by itself or with a note symbol that explains the starting point.
	Activity symbol	Indicates the activities that make up a modeled process. These symbols, which include short descriptions within the shape, are the main building blocks of an activity diagram.
	Connector symbol	Shows the directional flow, or control flow, of the activity. An incoming arrow starts a step of an activity; once the step is completed, the flow continues with the outgoing arrow.
	Joint symbol/ Synchronization bar	Combines two concurrent activities and re-introduces them to a flow where only one activity occurs at a time. Represented with a thick vertical or horizontal line.

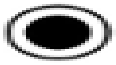
Activity Diagram Symbols

Symbol	Name	Description
	Fork symbol	Splits a single activity flow into two concurrent activities. Symbolized with multiple arrowed lines from a join.
	Decision symbol	Represents a decision and always has at least two paths branching out with condition text to allow users to view options. This symbol represents the branching or merging of various flows with the symbol acting as a frame or container.
	Note symbol	Allows the diagram creators or collaborators to communicate additional messages that don't fit within the diagram itself. Leave notes for added clarity and specification.
	Send signal symbol	Indicates that a signal is being sent to a receiving activity.

Activity Diagram Symbols

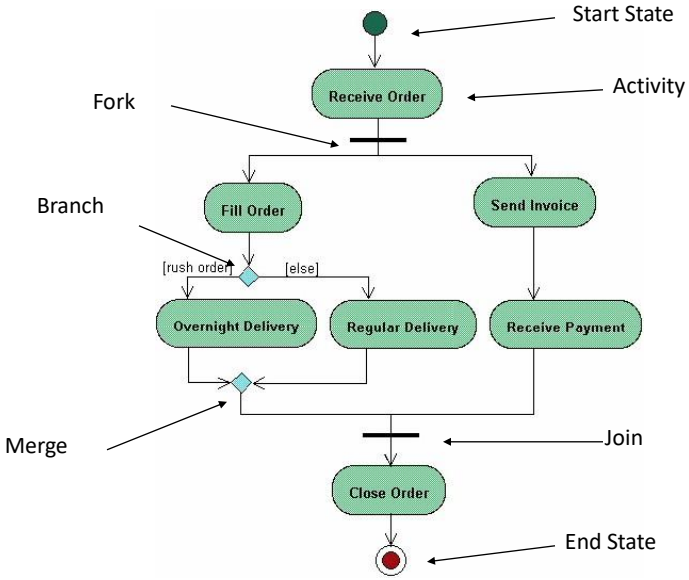
Symbol	Name	Description
	Receive signal symbol	Demonstrates the acceptance of an event. After the event is received, the flow that comes from this action is completed.
	Shallow history pseudostate symbol	Represents a transition that invokes the last active state.
	Option loop symbol	Allows the creator to model a repetitive sequence within the option loop symbol.
	Flow final symbol	Represents the end of a specific process flow. This symbol shouldn’t represent the end of all flows in an activity; in that instance, you would use the end symbol. The flow final symbol should be placed at the end of a process in a single activity flow.

Activity Diagram Symbols

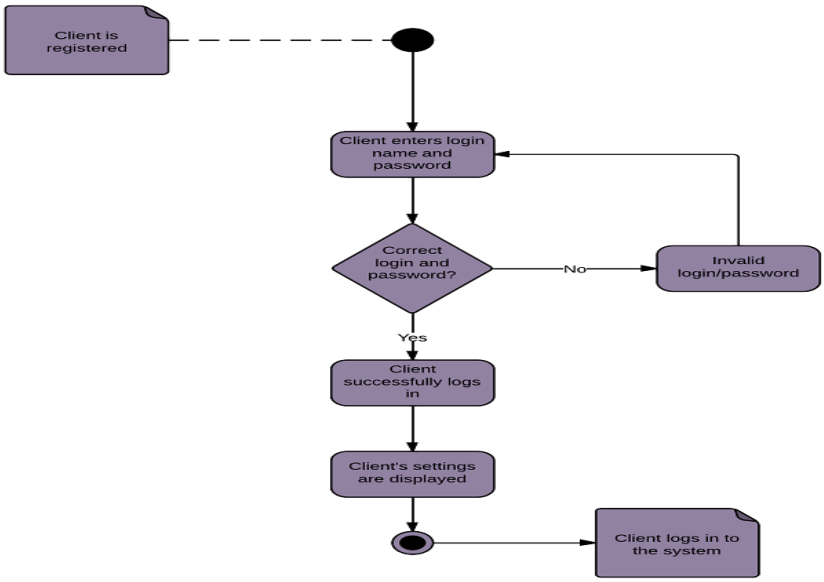
Symbol	Name	Description
[Condition]	Condition text	Placed next to a decision marker to let you know under what condition an activity flow should split off in that direction.
	End symbol	Marks the end state of an activity and represents the completion of all flows of a process.

- An activity partition or a swimlane is a high-level grouping of a set of related actions. A single partition can refer to many things, such as classes, use cases, components, or interfaces.
- If a partition cannot be shown clearly, then the name of a partition is written on top of the name of an activity.

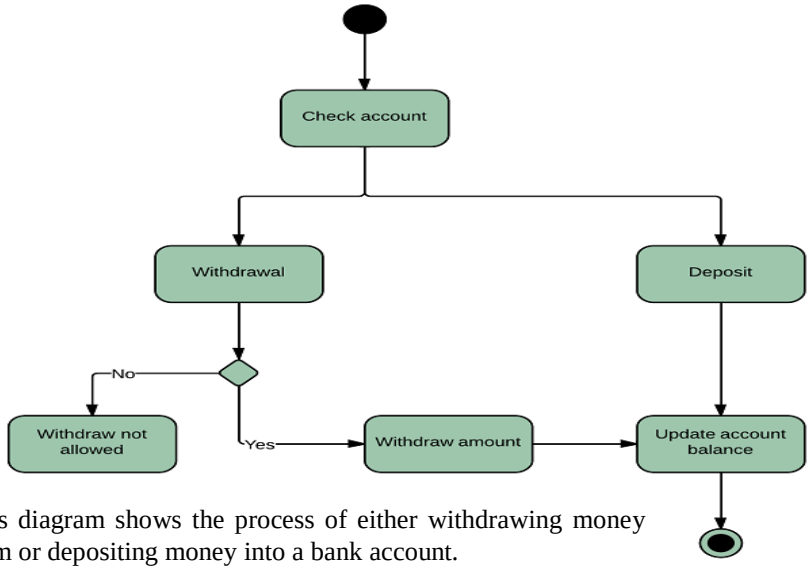
Activity Diagram Example



Activity Diagram for Login



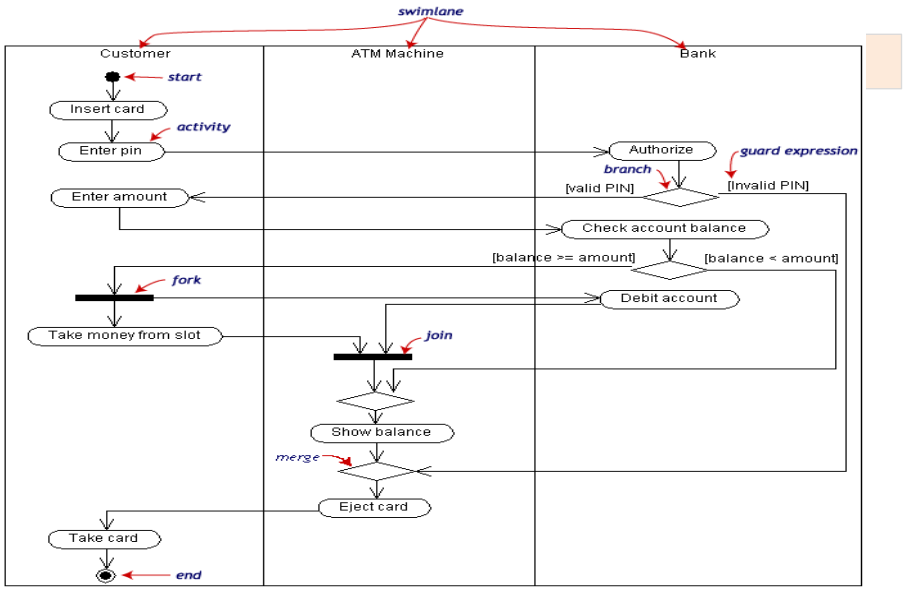
Activity Diagram for Banking system



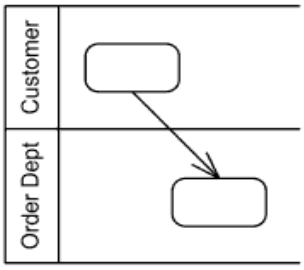
This diagram shows the process of either withdrawing money from or depositing money into a bank account.

Activity Diagram using swimlane

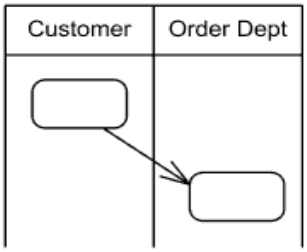
- Swimlanes (or activity partitions) indicate where activities take place.
- Swimlanes can also be used to identify areas at the technology level where activities are carried out
- Swimlanes allow the partition an activity diagram so that parts of it appear in the swimlane relevant to that element in the partition.
- Partitions may be constructed on the basis of:
  - the class and actor doing the activity
  - Partitioning by class and actor can help to identify new associations that have not been documented in the class model
  - Partitioning by use cases can help document how use cases interact.



Withdraw money from a bank account through an ATM

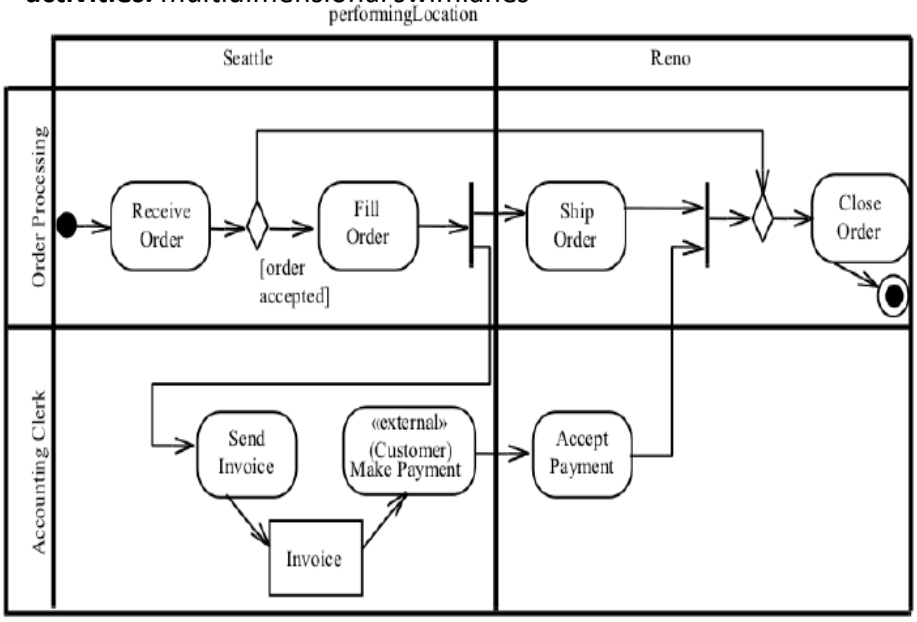


Activity partitions Customer and Order Dept as horizontal swimlanes



Activity partitions Customer and Order Dept as vertical swimlanes

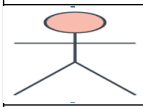
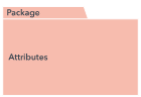
Swimlane : Indicate who is **responsible for each group of activities.** multidimensional swimlanes



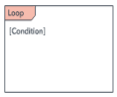
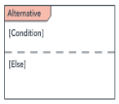
Sequence Diagram

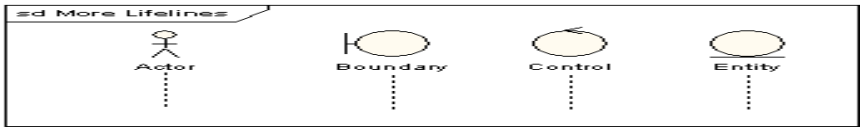
- A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together.
- Sequence diagrams are sometimes known as event diagrams or event scenarios.
- The purpose of a sequence diagram in UML is to visualize the sequence of a message flow in the system.

Sequence Diagram Symbols

Symbol	Name	Description
	Object symbol	Represents a class or object in UML. The object symbol demonstrates how an object will behave in the context of the system. Class attributes should not be listed in this shape.
	Activation box	Represents the time needed for an object to complete a task. The longer the task will take, the longer the activation box becomes.
	Actor symbol	Shows entities that interact with or are external to the system.
	Package symbol	Used in UML 2.0 notation to contain interactive elements of the diagram. Also known as a frame, this rectangular shape has a small inner rectangle for labeling the diagram.

Sequence Diagram Symbols

Symbol	Name	Description
	Lifeline symbol	Represents the passage of time as it extends downward. This dashed vertical line shows the sequential events that occur to an object during the charted process. Lifelines may begin with a labeled rectangle shape or an actor symbol.
	Option loop symbol	Used to model if/then scenarios, i.e., a circumstance that will only occur under certain conditions.
	Alternative symbol	Symbolizes a choice (that is usually mutually exclusive) between two or more message sequences. To represent alternatives, use the labeled rectangle shape with a dashed line inside.



Sequence Diagram

- Lifeline
  - A lifeline represents a single participant in an interaction. It describes how an instance of a specific classifier participates in the interaction.
  - A lifeline represents a role that an instance of the classifier may play in the interaction.
- Message
  - A message is a specific type of communication between two lifelines in an interaction. A message involves following activities,
  - A call message which is used to call an operation.
  - A message to create an instance.
  - A message to destroy an instance.

Message Symbol

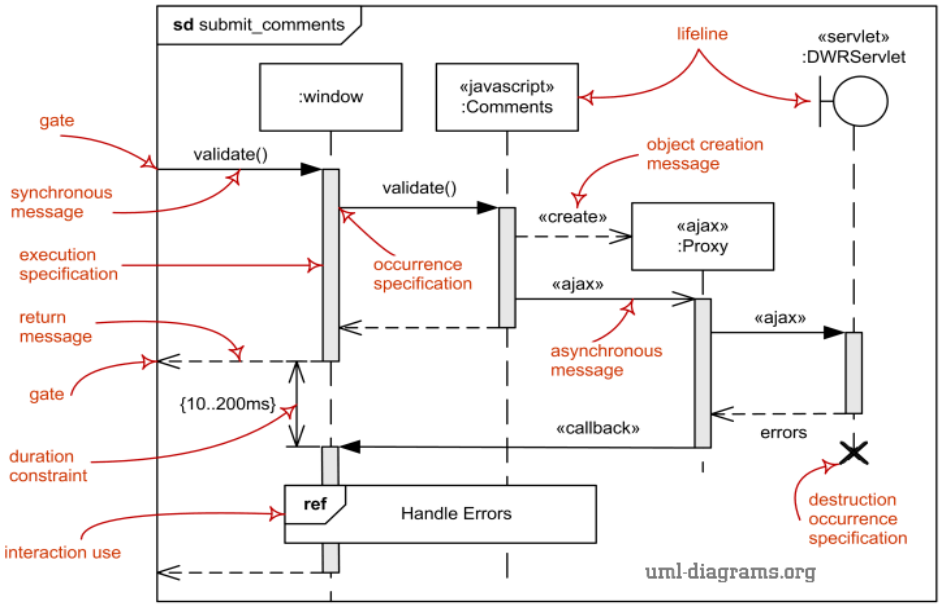
- Use the following arrows and message symbols to show how information is transmitted between objects. These symbols may reflect the start and execution of an operation or the sending and reception of a signal.

Symbol	Name	Description
	Synchronous message symbol	Represented by a solid line with a solid arrowhead. This symbol is used when a sender must wait for a response to a message before it continues. The diagram should show both the call and the reply.
	Asynchronous message symbol	Represented by a solid line with a lined arrowhead. Asynchronous messages don't require a response before the sender continues. Only the call should be included in the diagram.
	Asynchronous return message symbol	Represented by a dashed line with a lined arrowhead.

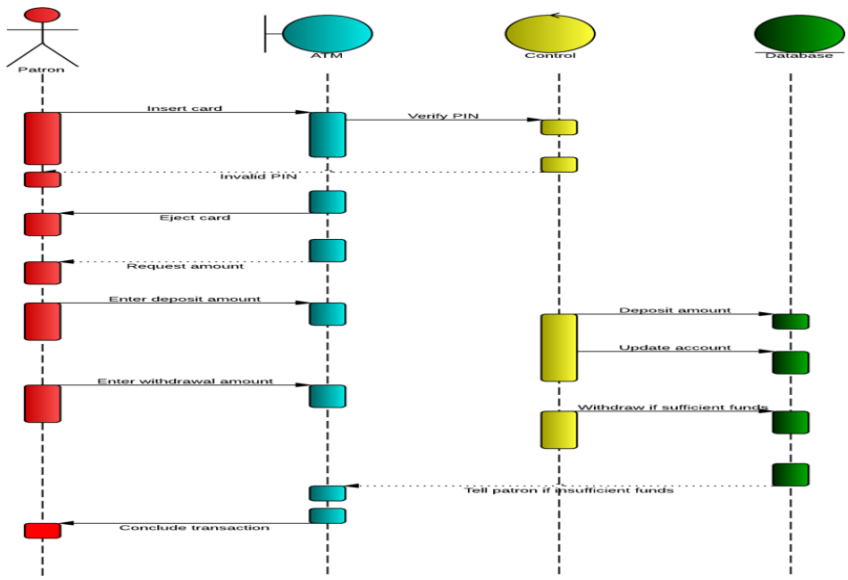
Message Symbol

Symbol	Name	Description
	Asynchronous create message symbol	Represented by a dashed line with a lined arrowhead. This message creates a new object.
	Reply message symbol	Represented by a dashed line with a lined arrowhead, these messages are replies to calls.
	Delete message symbol	Represented by a solid line with a solid arrowhead, followed by an X. This message destroys an object.

Sequence Diagram Example



Sequence Diagram of ATM System

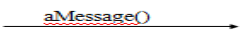


Collaboration Diagram

- The purpose of a collaboration diagram is to emphasize structural aspects of a system, i.e., how various lifelines in the system connects.
- They are used to understand the object architecture within a system rather than the flow of a message in a sequence diagram.
- An object an entity that has various attributes associated with it.
- There are multiple objects present inside an object-oriented system where each object can be associated with any other object inside the system.
- Collaboration or communication diagrams are used to explore the architecture of objects inside the system. The message flow between the objects can be represented using a collaboration diagram.

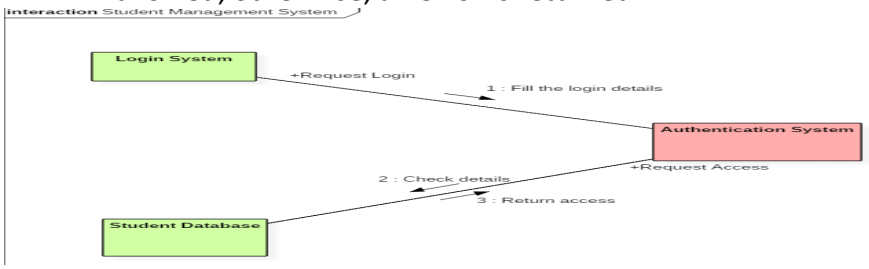
Collaboration Diagram

- There are three primary elements of a collaboration diagram:
  - Objects
  - Links
  - Messages

AN ACTOR	
AN OBJECT	
AN ASSOCIATION	
A MESSAGE	

Collaboration Diagram

- The collaboration diagram represents a student information management system. The flow of communication in the above diagram is given by
  - A student requests a login through the login system.
  - An authentication mechanism of software checks the request.
  - If a student entry exists in the database, then the access is allowed; otherwise, an error is returned.

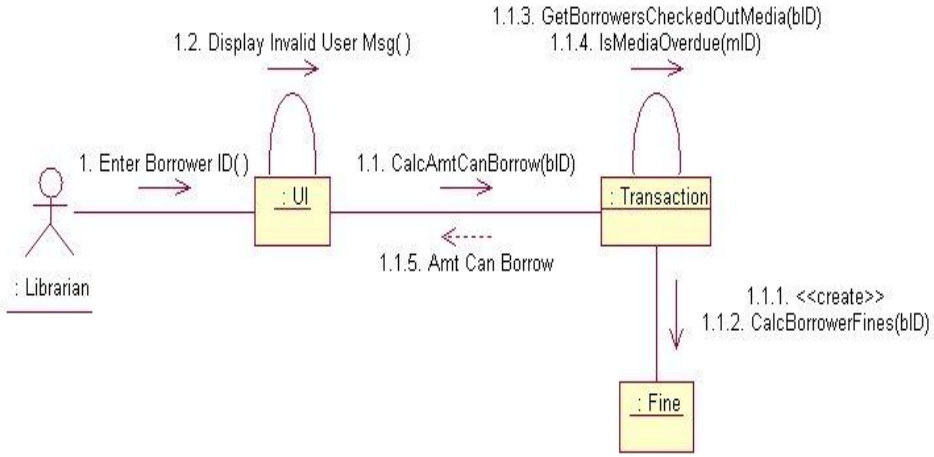


Objects

- Objects - rectangles containing the object signature - object signature:
  - object name : object Class
  - object name (optional) - starts with lowercase letter
- class name (mandatory) - starts with uppercase letter
- Objects connected by lines - actor can appear
- Objects participating in a collaboration come in two flavors—supplier and client
- Supplier objects are the objects that supply the method that is being called, and therefore **receive** the message
- Client objects call methods on supplier objects, and therefore **send** messages.

Objects

- Transaction object acts as a Supplier to the UI (User Interface) Client object. In turn, the Fine object is a Supplier to the Transaction Client object.



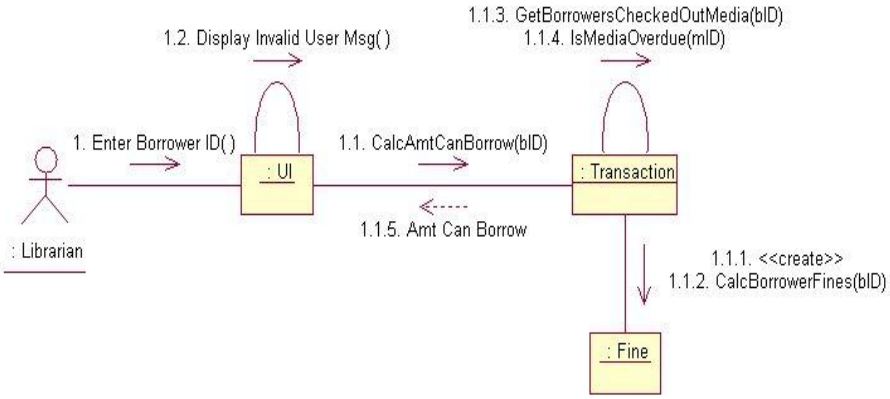
Message

- An interaction is implemented by a group of objects that collaborate by exchanging messages.
- Messages in collaboration diagrams are shown as arrows pointing from the Client object to the Supplier object.
- Typically, messages represent a client invoking an operation on a supplier object.
- Message icons have one or more messages associated with them
- Messages are composed of message text prefixed by a sequence number
- Time is not represented explicitly in a collaboration diagram, and as a result the various messages are numbered to indicate the sending order

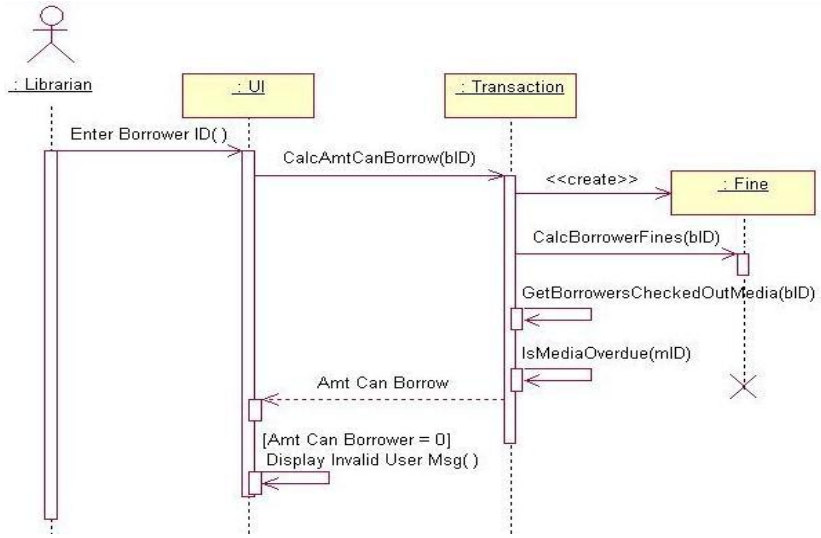
Objects Changing State

- State of on object can be indicated
- Initial state is indicated with <<create>>
- If an object changes significantly during an interaction, you can add a new instance of the object to the diagram, draw a link between them and add a message with the stereotype <<become>>

Objects Changing State

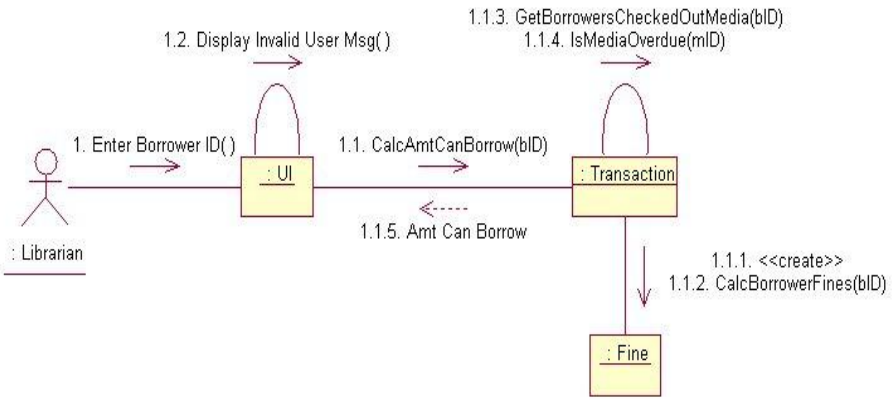


Sequence Vs Collaboration



Sequence diagram is better at 'time ordering'

Sequence Vs Collaboration



Collaboration diagram is better at showing the relationship between objects


THANKS