CSC 3380 Aymond

Section 1

2/17/2020

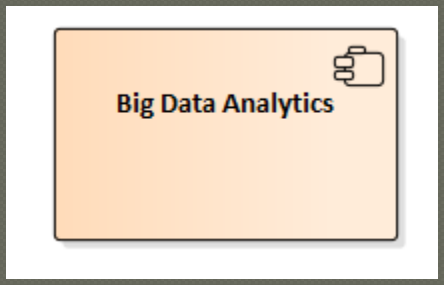
Homework Assignment
- Enterprise Architect Component Diagram, Due 11PM 2/19

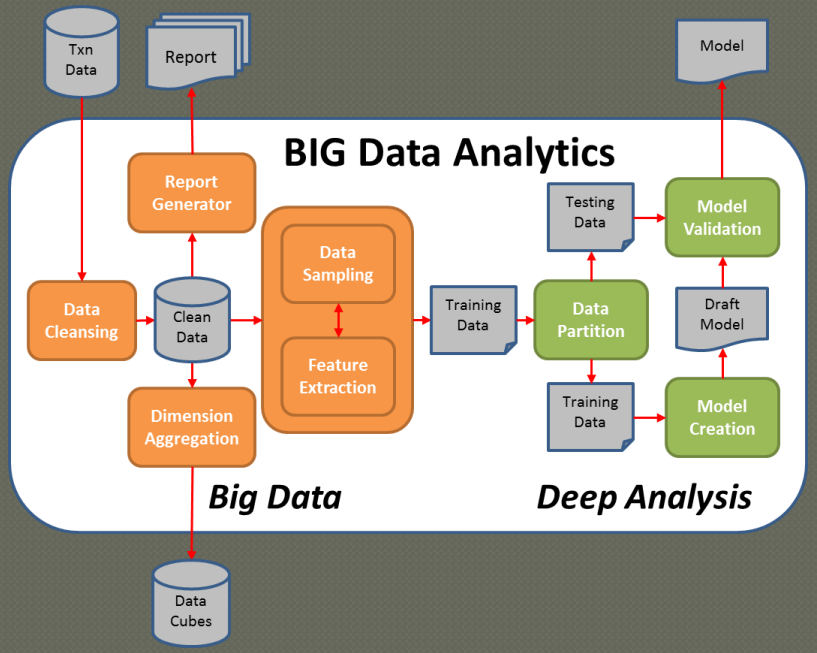
Project News
- Next Milestone: #2
  - Due Friday 2/21, 11PM
  - Upload to Moodle (1 upload for entire team)
  - Outline is in Project Kickoff Lecture Notes
  - BE SURE TO UPDATE SECTIONS FROM MILESTONE #1
  - All UML diagrams must be developed in EA
- In-class Milestone #2 Presentations
  - Monday, March 2
  - Chanuka's teams: 1240 PFT
  - Clinton's teams: 1245 PFT
  - Qing's teams: 1258 PFT

Midterm Exam
- Wednesday, March 4

Homework Assignment

- Assignment:
  - Convert the given architecture into the UML standard format, using Enterprise Architect (EA)
  - The architecture will be the same, but UML elements do not **look** like the elements in the architecture given
- Key things to consider in your mapping
  - The overall system, is a component
    - Since it contains components, we need to use the UML Packaging Component
  - Components within the system that contain other components should also be Packaging Components

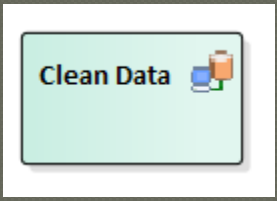




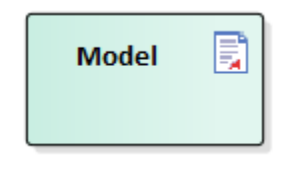
http://horicky.blogspot.com/2012/08/big-data-analytics.html

Homework Assignment

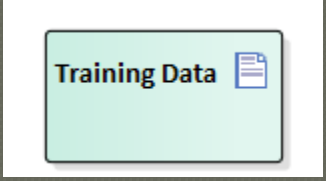
- Databases are represented as Database Connections in UML
  - This can be found in the EA Data Modeling toolbox

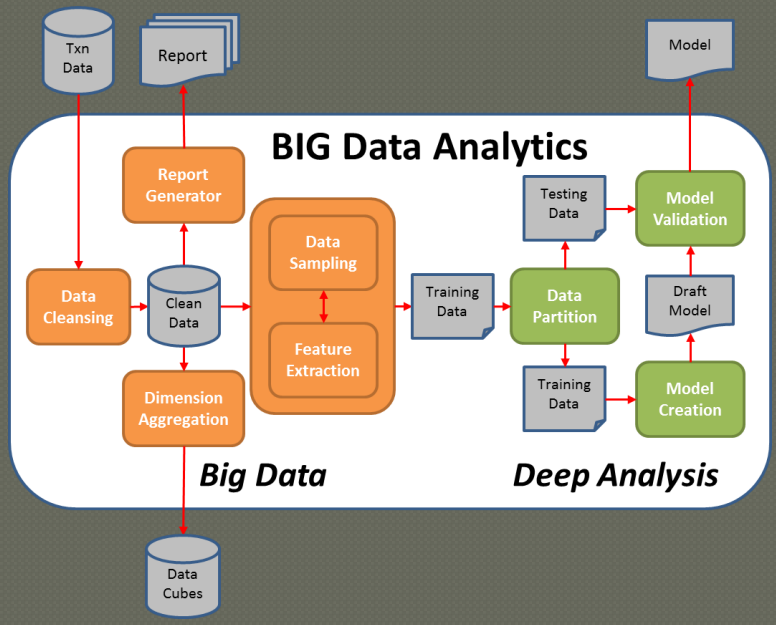


- Formatted files, such as reports (e.g., PDFs, Spreadsheets, Word documents) are represented as Documents in UML



- Text files, such as data logs and text I/O, are represented as Artifacts in UML

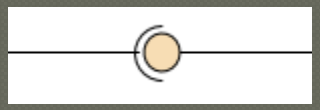




http://horicky.blogspot.com/2012/08/big-data-analytics.html

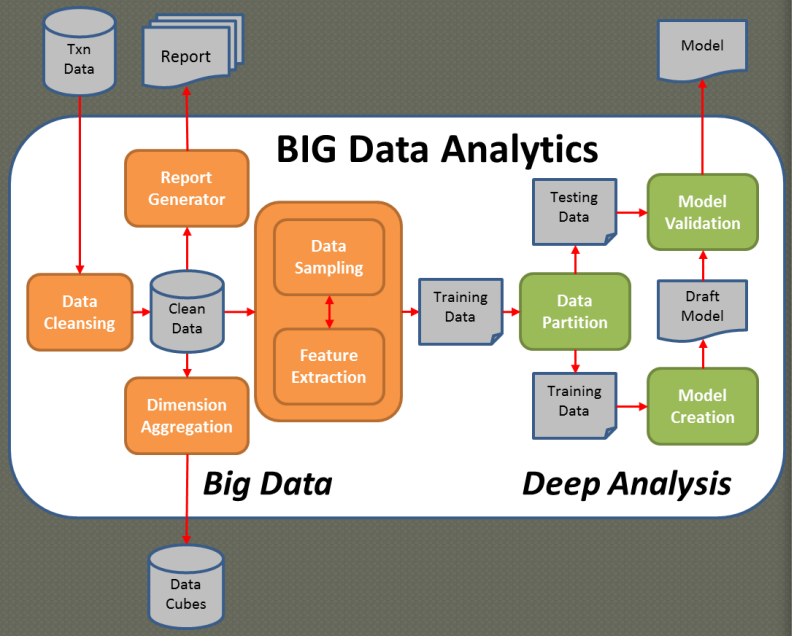
Homework Assignment

- Communication between components should be interfaces



  - Two-way component communication is represented with two interfaces, flowing in the two directions
- Component I/O (including file I/O) is data flow and should use the data flow relationship, which is a solid line with a filled in triangle for the arrow

  - You can use the Data Flow tool in the Data Flow Diagram set of tools.



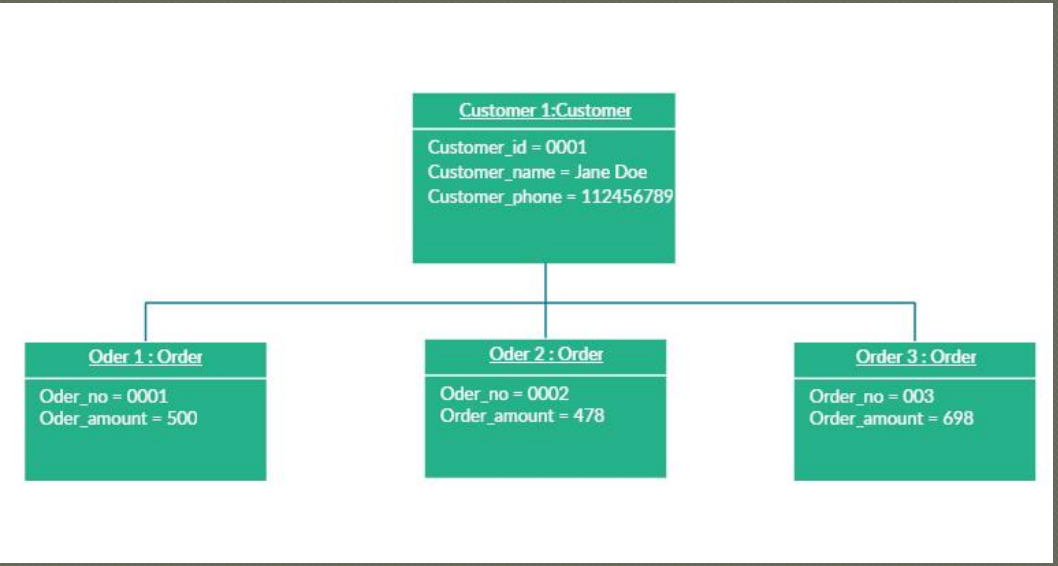
http://horicky.blogspot.com/2012/08/big-data-analytics.html

Structural Modeling

- Class Diagrams
  - Classes, features, and relationships
- **Object Diagrams**
  - Example configurations of instances
- Communication Diagrams
  - structural organization of the objects that send and receive messages
- Packages
  - Compile-time hierarchic structure

Object Diagram

- Shows a set of objects and their relationships
- Represent static snapshots of instances of things found in class diagrams
- Provides the object state
  - Values of member variables



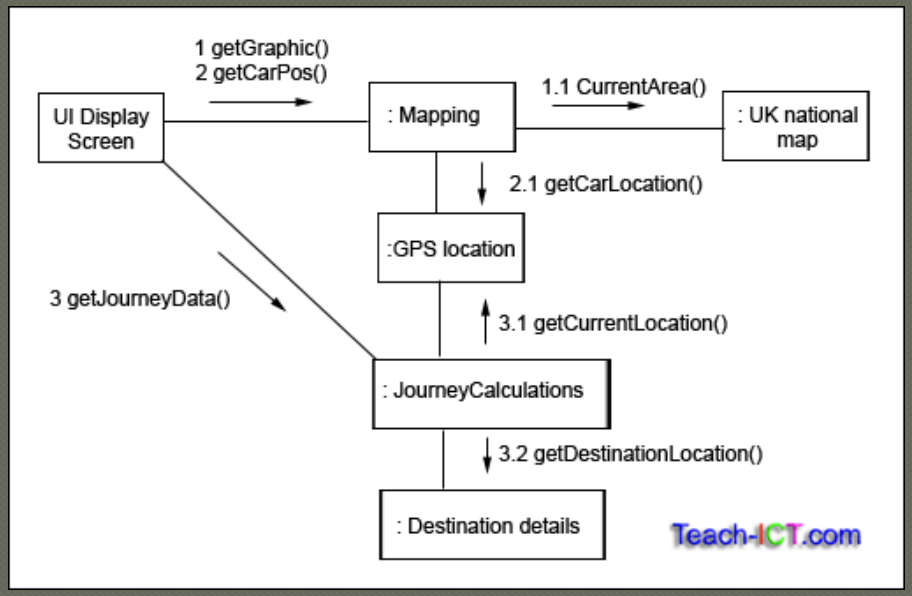
https://creately.com/blog/diagrams/uml-diagram-types-examples/

Structural Modeling

- Class Diagrams
  - Classes, features, and relationships
- Object Diagrams
  - Example configurations of instances
- **Communication Diagrams**
  - structural organization of the objects that send and receive messages
- Packages
  - Compile-time hierarchic structure

Communication Diagrams

- In UML 1.0, these diagrams were called collaboration diagrams
- Communication diagrams are used to show the messages that flow from one **object** to another within the system and the order in which they happen.



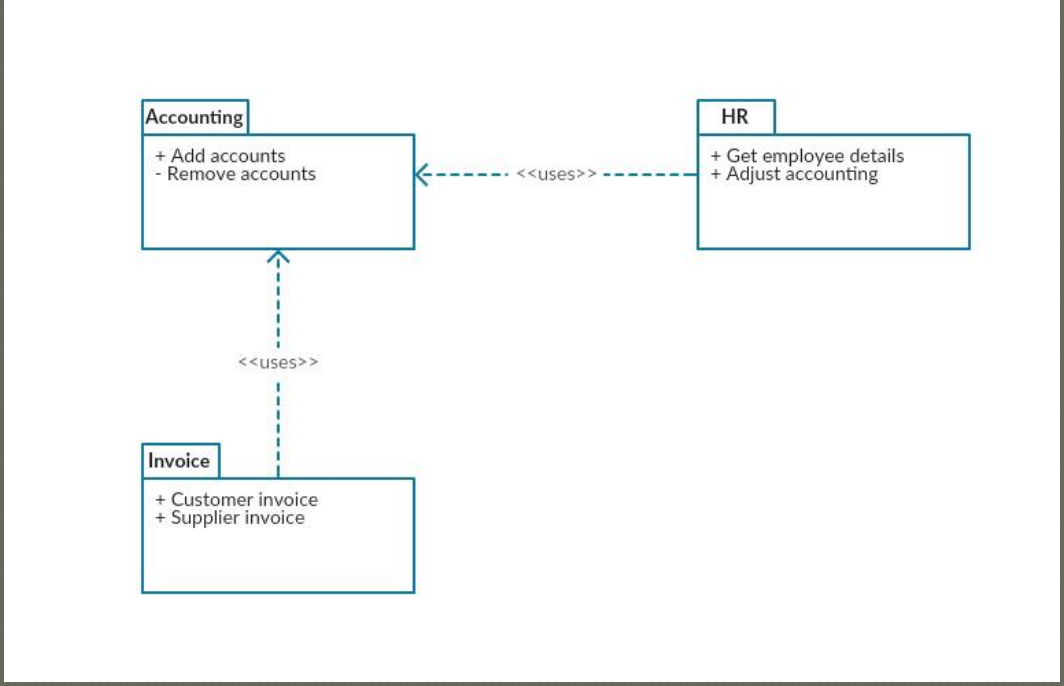
https://www.teach-ict.com/as_as_computing/ocr/H447/F453/3_3_6/uml/miniweb/pg15.htm

Structural Modeling

- Class Diagrams
  - Classes, features, and relationships
- Object Diagrams
  - Example configurations of instances
- Communication Diagrams
  - structural organization of the objects that send and receive messages
- **Package Diagrams**
  - Compile-time hierarchic structure

Package Diagram

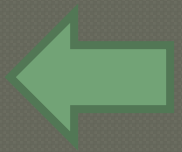
- A package diagram shows the dependencies between different packages in a system.



https://creately.com/blog/diagrams/uml-diagram-types-examples/

Unified Modeling Language (UML)

- Introduction to UML
- Architectural Modeling
- Structural Modeling
- **Behavioral Modeling**

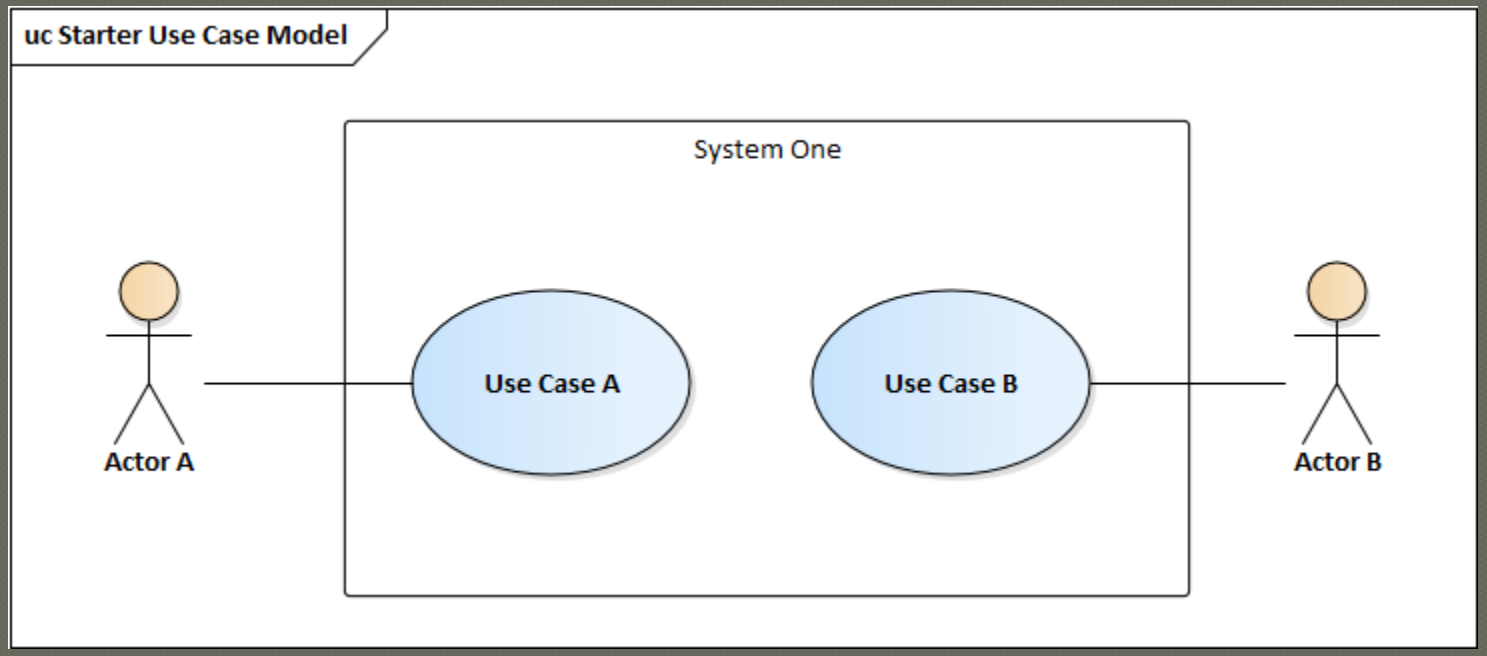


UML Diagrams

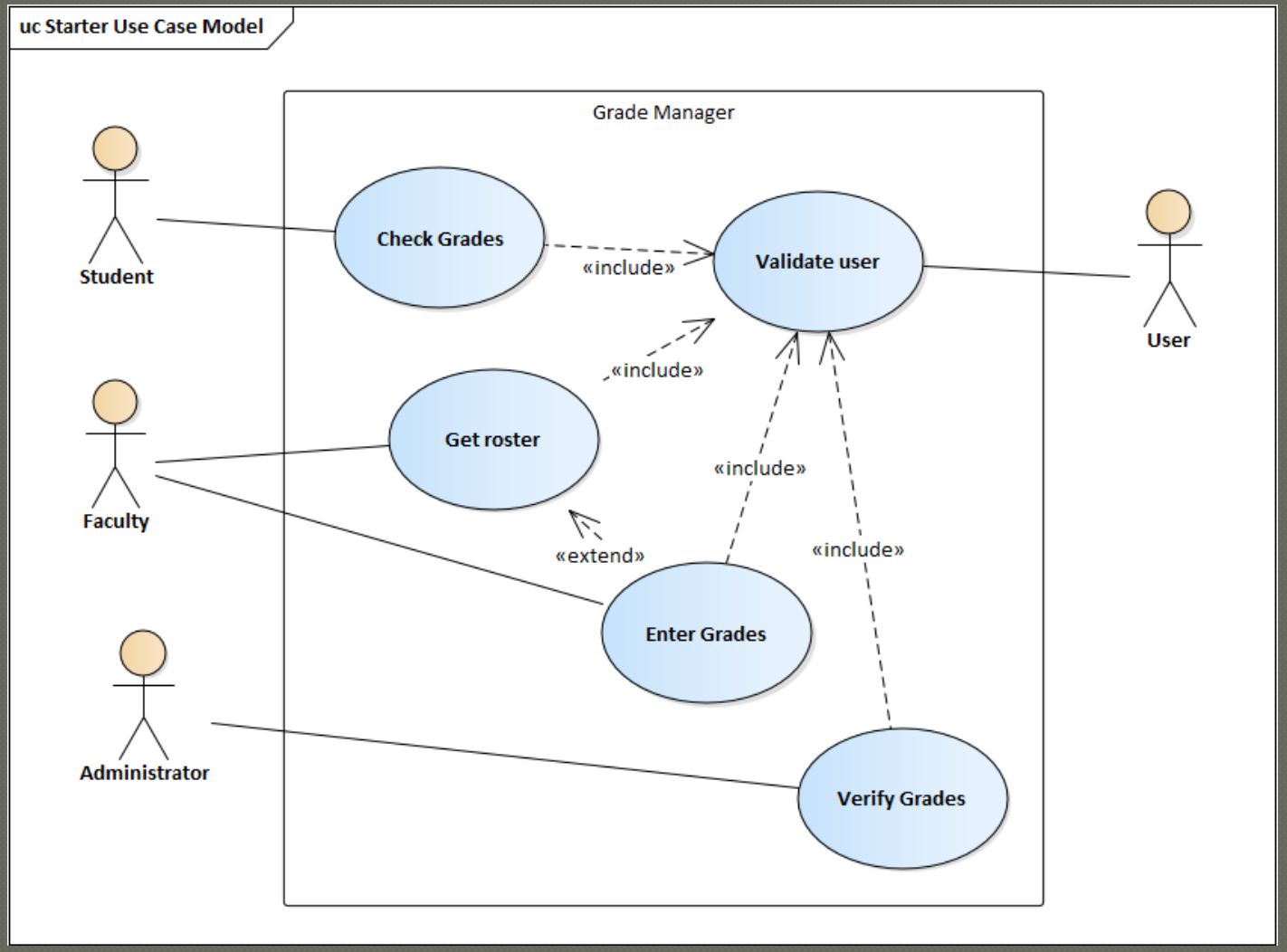
- Behavioral Modeling
  - **Use Case Diagrams**
  - Sequence Diagram
  - Activity Diagrams
  - State Machine Diagrams

Use Case Diagram

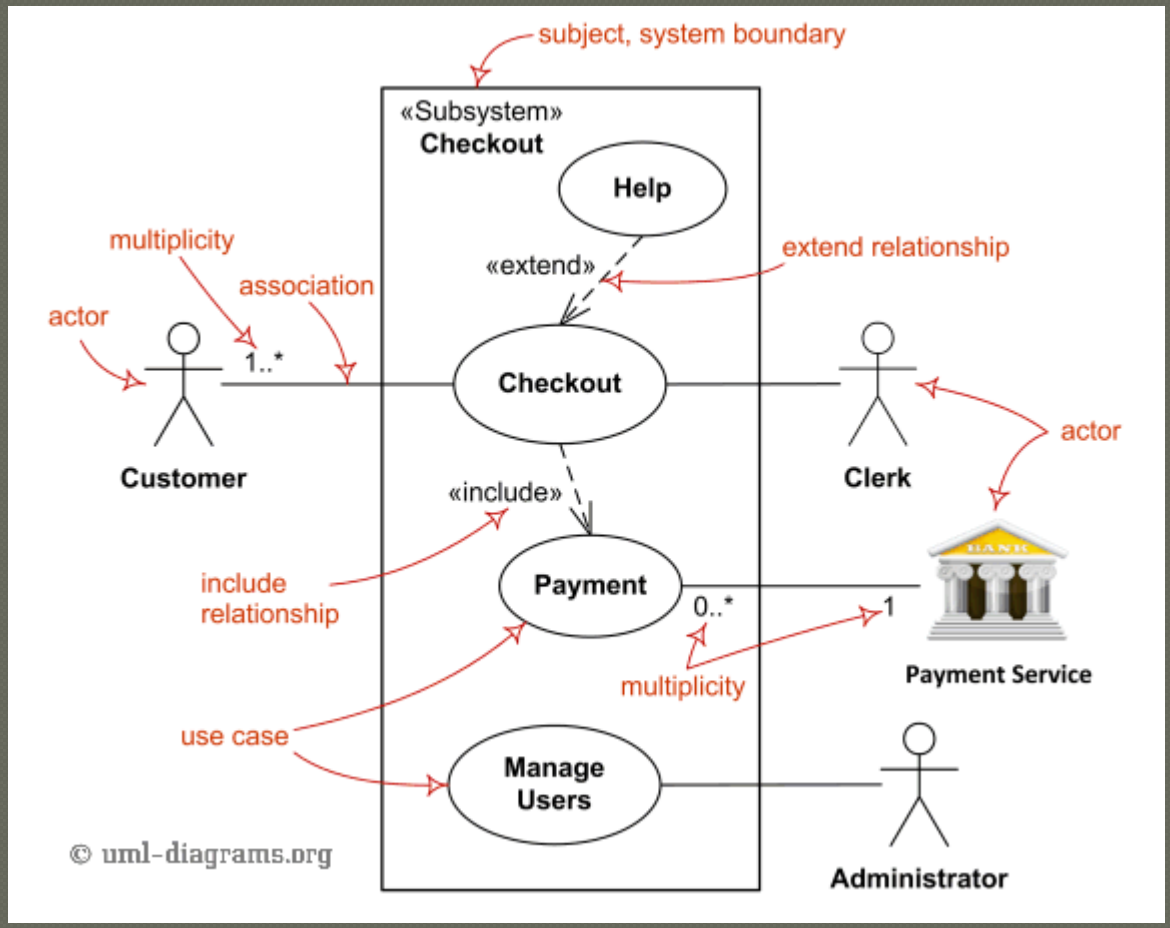
- An analysis/requirements specification diagram
- How users interact with the system
  - Actor: user, depicted as a stick person
  - Use Case: How a user uses the system, depicted as an ovel



Use Case Diagram: Grade Manager



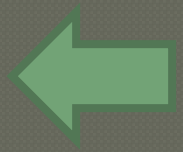
Use Case Diagram: Checkout



https://www.uml-diagrams.org/use-case-diagrams.html

UML Diagrams

- Behavioral Modeling
  - Use Case Diagrams
  - **Sequence Diagram**
  - Activity Diagrams
  - State Machine Diagrams



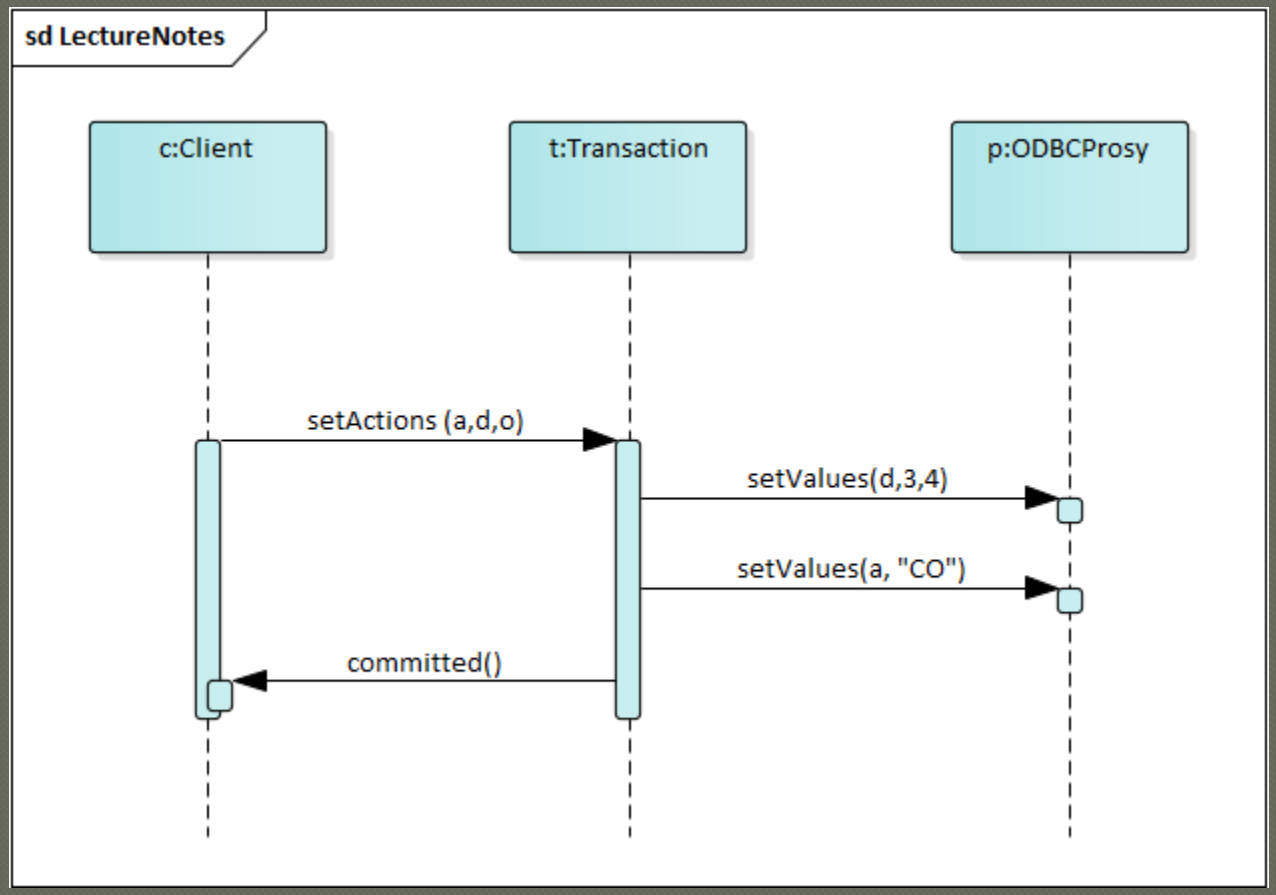
Sequence Diagram

- The most common form of interaction diagram
- A control flow diagram
- Emphasizes the time ordering of messages
- Shows a set of objects and the messages sent and received by those objects

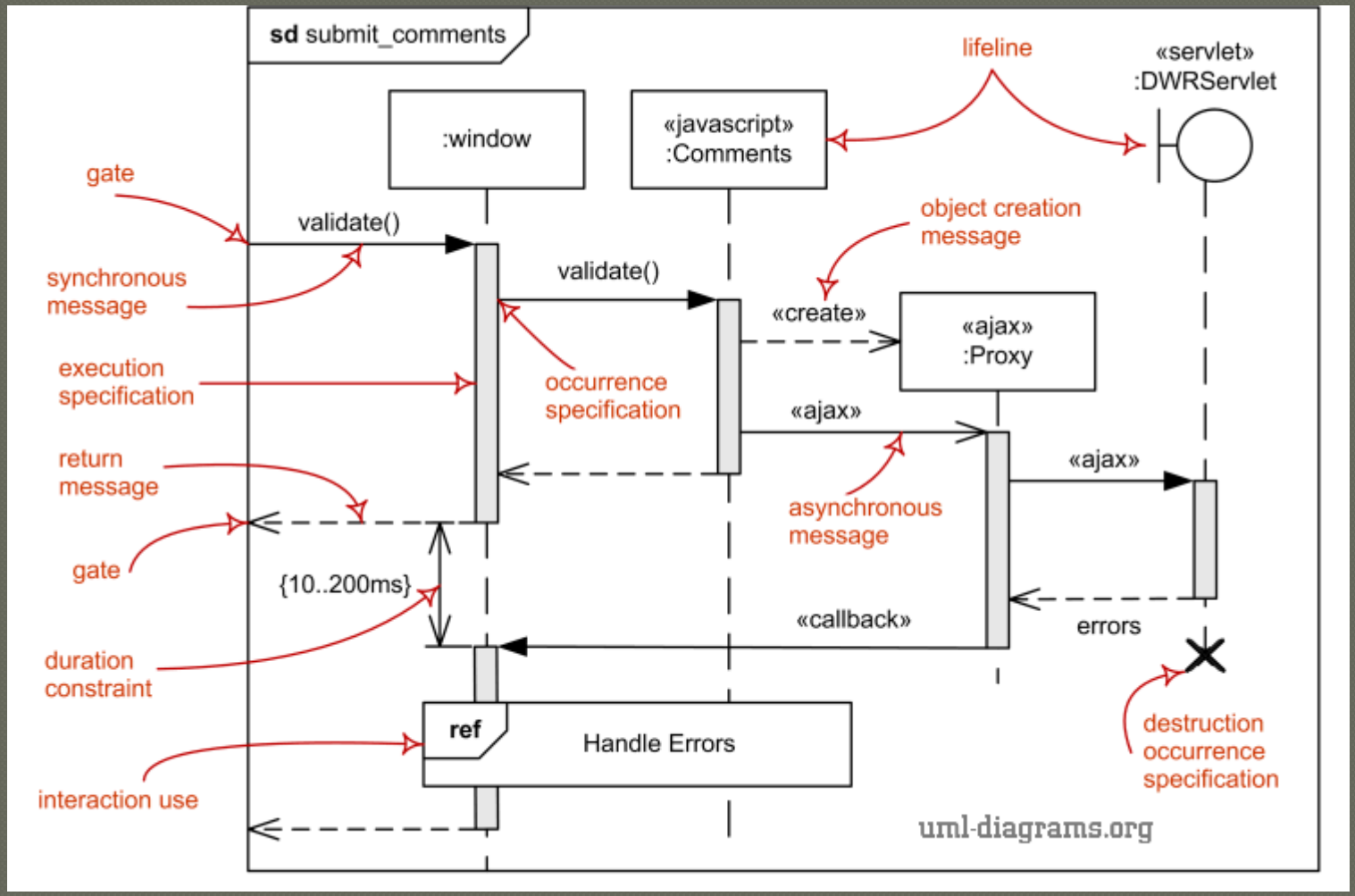
Major Sequence Diagram Elements

- Object 
- Object Instantiation 
- Lifeline 
- Interaction 

Sequence Diagram



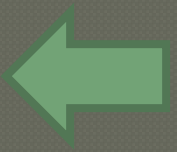
Sequence Diagram Example



https://www.uml-diagrams.org/sequence-diagrams.html

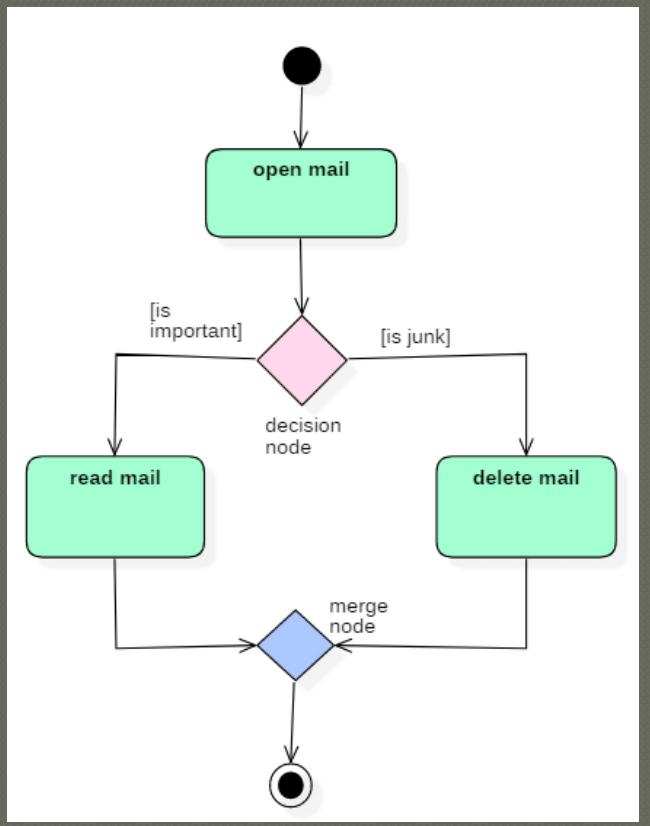
UML Diagrams

- Behavioral Modeling
  - Use Case Diagrams
  - Interaction Diagrams
  - Sequence Diagrams
  - **Activity Diagrams**
  - State Machine Diagrams



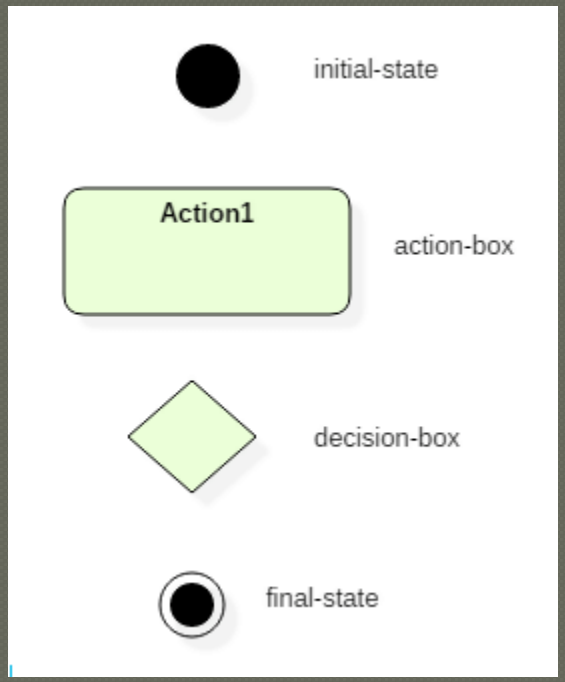
Activity Diagram

- A control flow diagram
- Similar to flowcharts, but with some key extensions
- Shows a set of activities, the sequential or branching flow from activity to activities, and objects that act and are acted upon



https://www.guru99.com/uml-activity-diagram.html

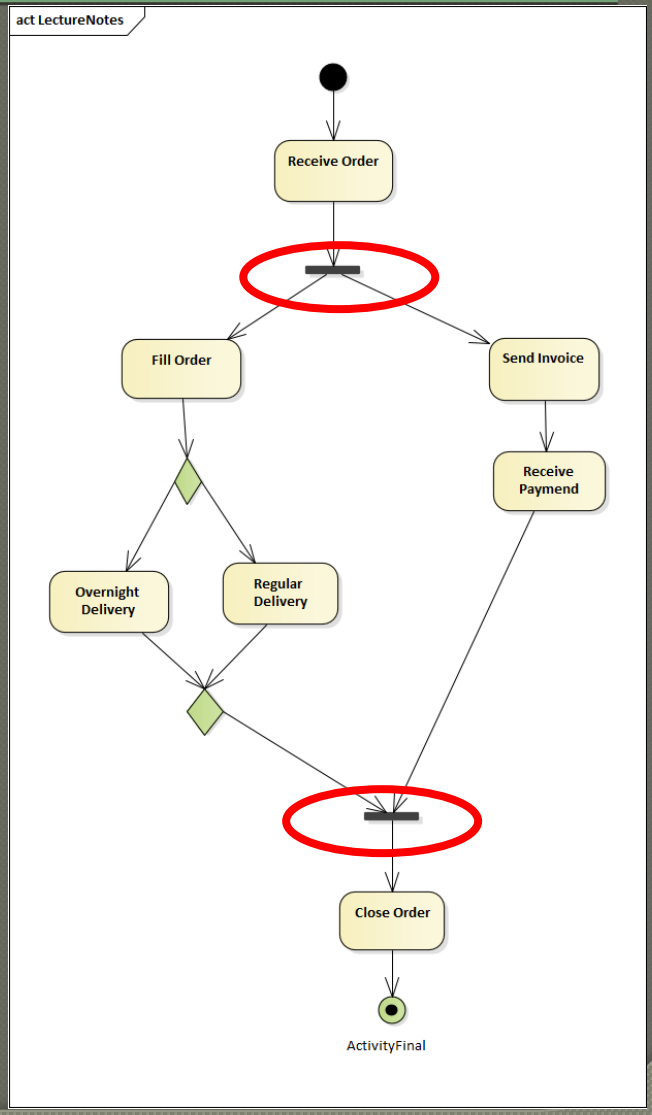
Activity Diagram Elements



https://www.guru99.com/uml-activity-diagram.html

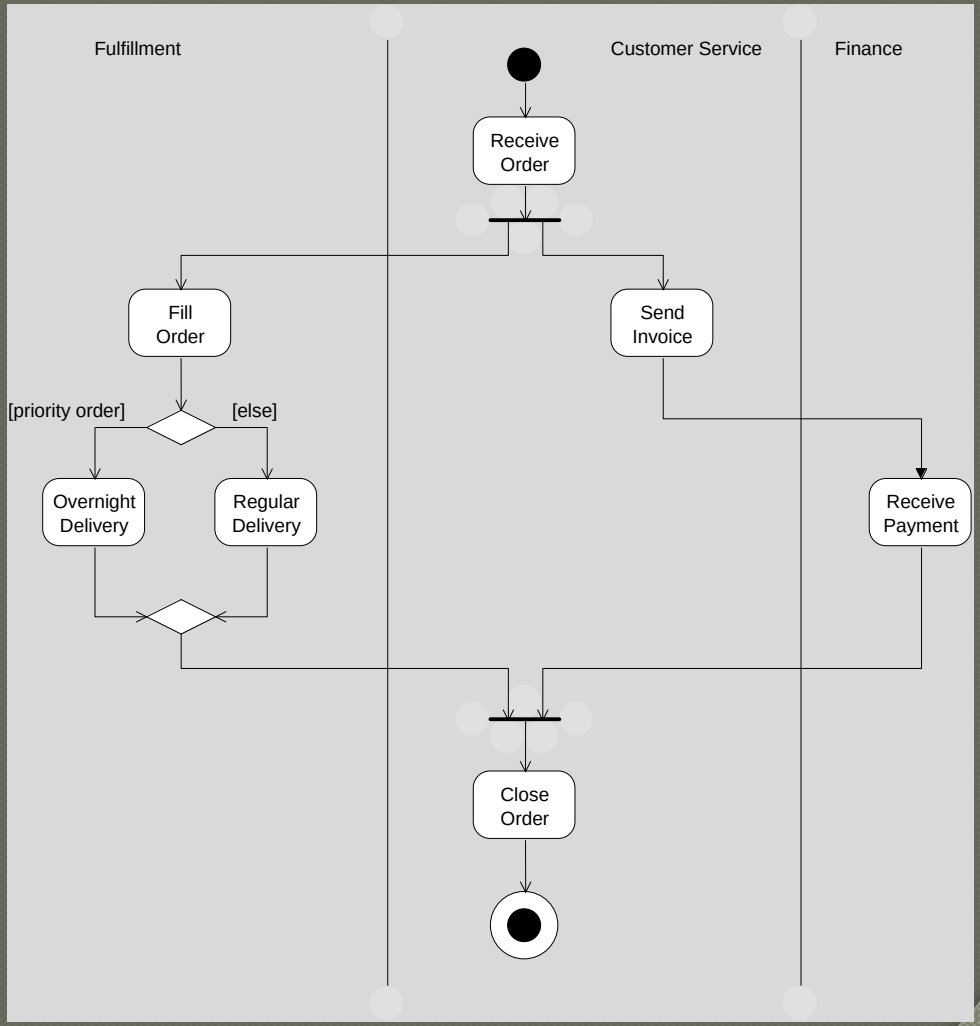
Fork/Join

- Fork: activities are performed independently and in parallel
- Join: All forked activities must be completed before continuing from that point



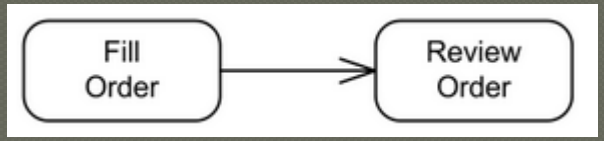
Partitions

- An activity partition is an a group of activities that have some common characteristic
- May represent activities performed by different components

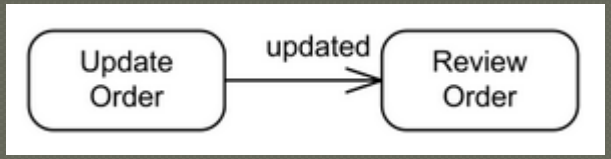


A Note on Activity Edges

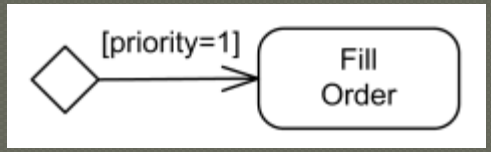
- Activity edge is notated by an open arrowhead line connecting two activity nodes



- If the edge has a name, it is notated near the arrow



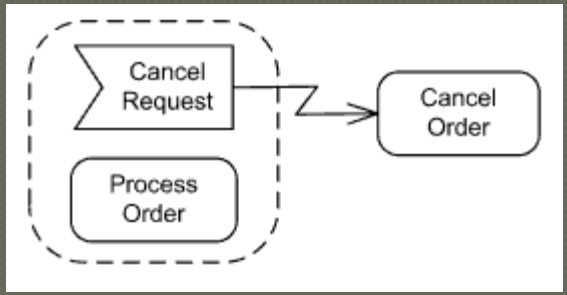
- Activity edge can have a guard
  - specification evaluated at runtime to determine if the edge can be traversed
  - The guard must evaluate to true for every token that is offered to pass along the edge
  - he guard of the activity edge is shown in square brackets that contain the guard.



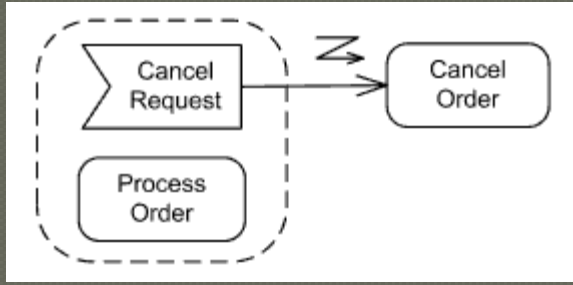
https://www.uml-diagrams.org/activity-diagrams.html

Interrupting Edge

- Interrupting edge is activity edge expressing interruption for regions having interruptions
  - It is rendered as a lightning-bolt



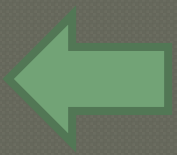
- An option for notating an interrupting edge is a zig zag adornment on a straight line



https://www.uml-diagrams.org/activity-diagrams.html

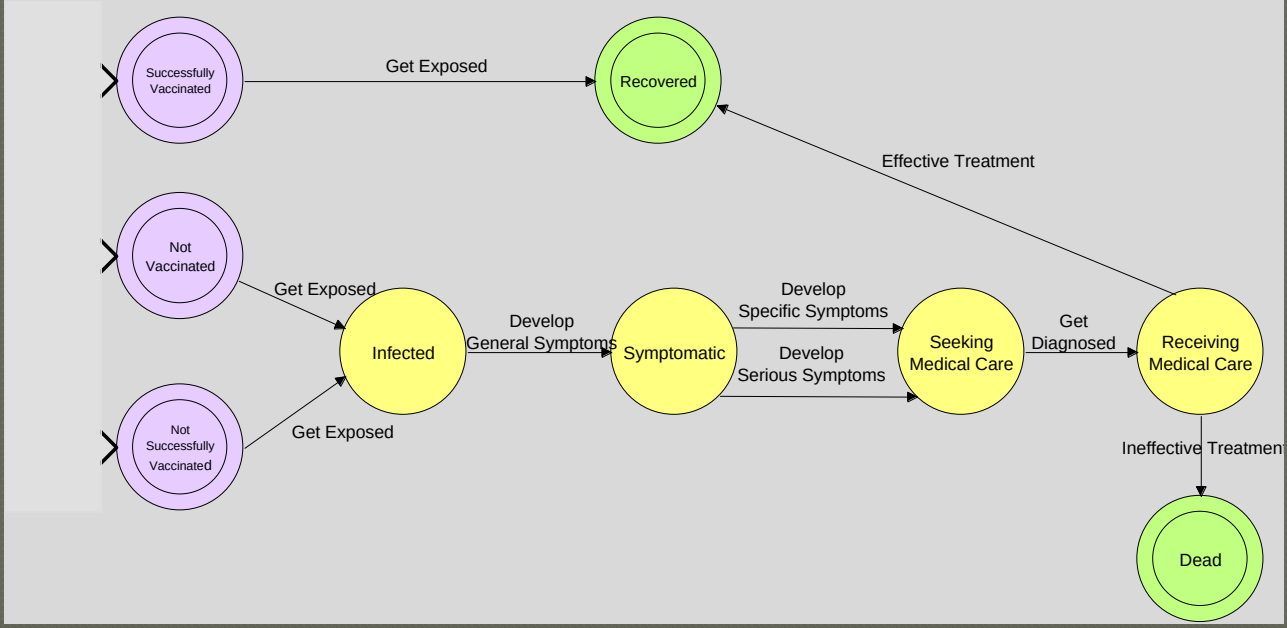
UML Diagrams

- Behavioral Modeling
  - Use Case Diagrams
  - Interaction Diagrams
  - Sequence Diagrams
  - Activity Diagrams
  - **State Machine Diagrams**



Finite State Machine

- A finite state machine is a mathematical construct
- It is a type of state transition diagram
  - Others include cellular automaton, petri nets, and Turing machines
  - Entities change state upon the trigger of an event
- Common uses
  - Discrete Event Simulations
  - Asynchronous Programming
  - UI Navigation



State Machine Diagram

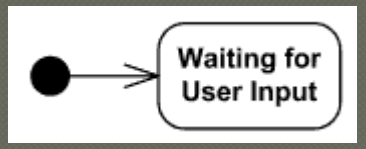
- A control flow diagram
- Shows discrete behavior of a part of a designed system through finite state transitions
- How events change an object over its life

Key State Machine Elements

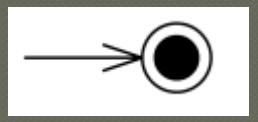
- Simple State: rounded rectangle

- State Transition, solid line with open arrow

- Initial (Start) State, solid circle

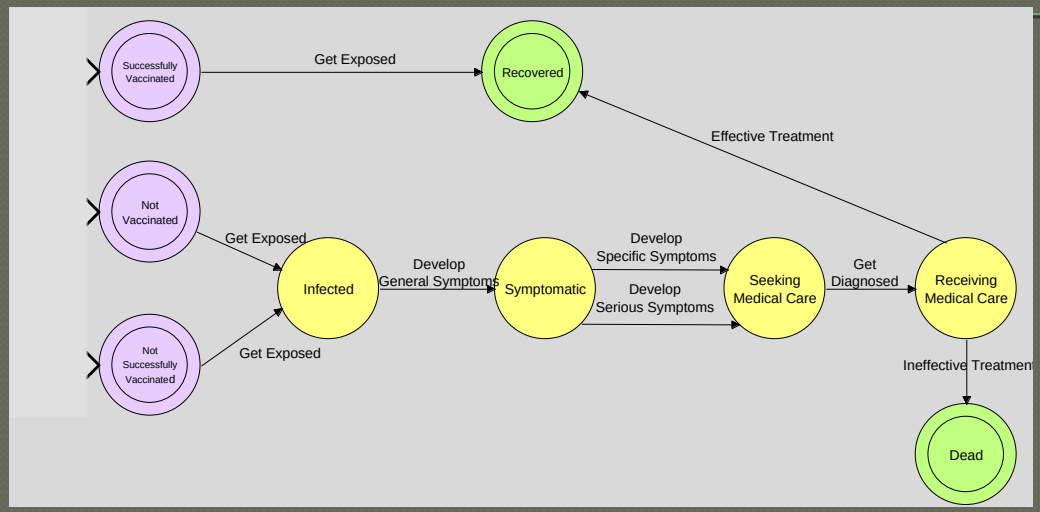


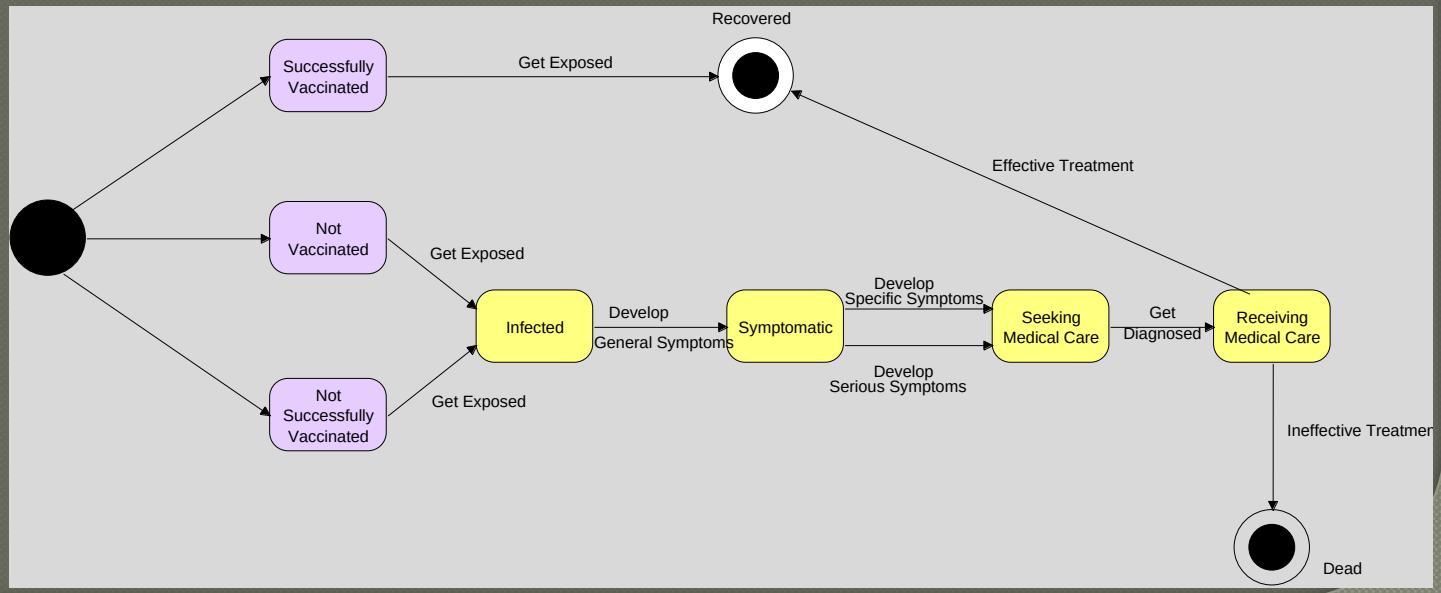
- Final (End) State, solid circle surrounded by solid line



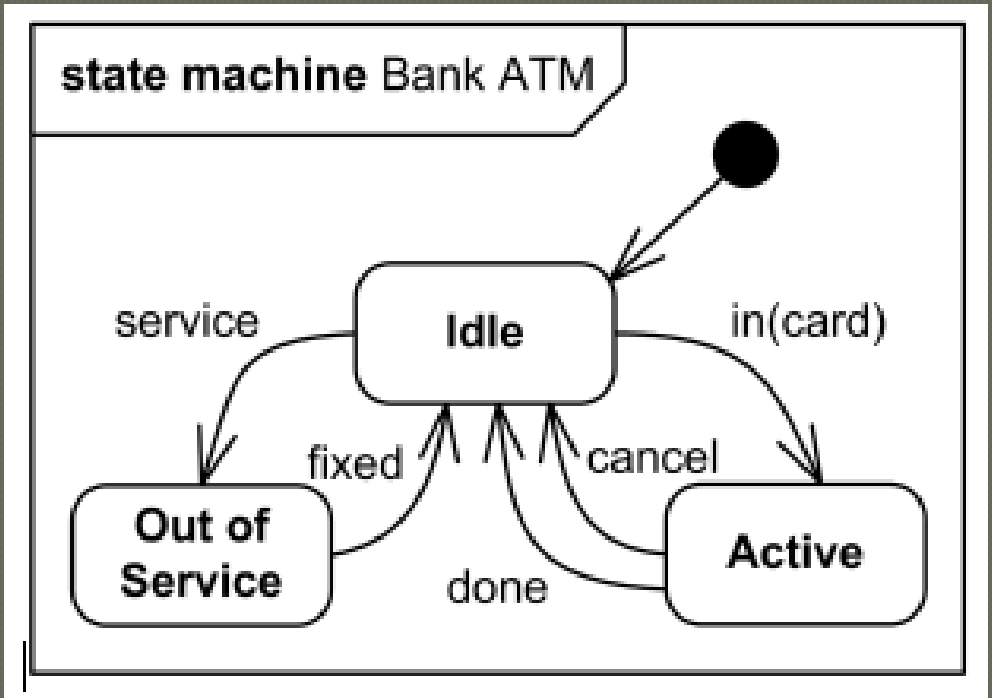
https://www.uml-diagrams.org/state-machine-diagrams.html

State Machine Diagram





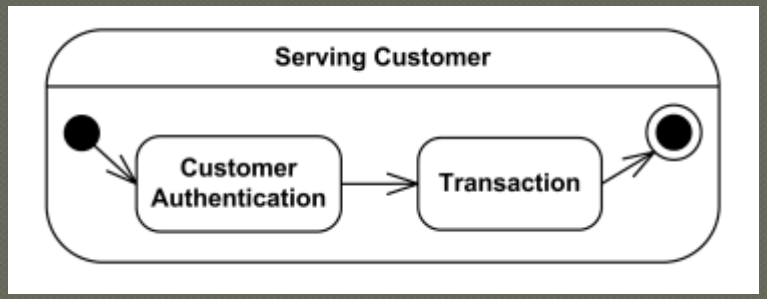
Example State Machine Diagram: Bank ATM



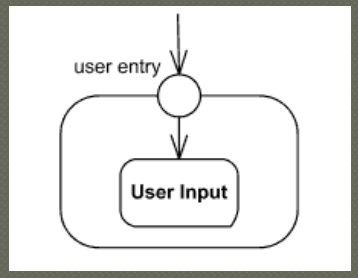
https://www.uml-diagrams.org/state-machine-diagrams.html

Composite State

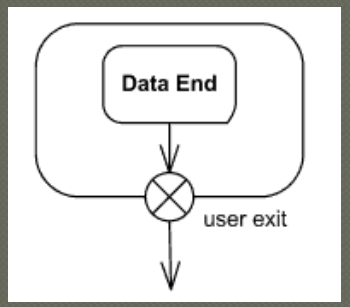
- Composite State



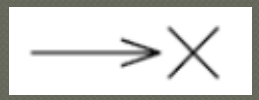
- Entry Point (for sub-state)



- Exit Point (for sub-state)



- Terminate State



https://www.uml-diagrams.org/state-machine-diagrams.html