

UML

Diagrams

Diagrams in UML

A *Diagram* is the graphical presentation of a set of elements, most often rendered as a connected graph of things and relationships.

1. Class Diagram, Package diagram, Object diagram

2. Use Case Diagram.

3. Sequence Diagram.

4. Collaboration Diagram.

5. State Chart Diagram.

6. Activity Diagram.

Diagram Types

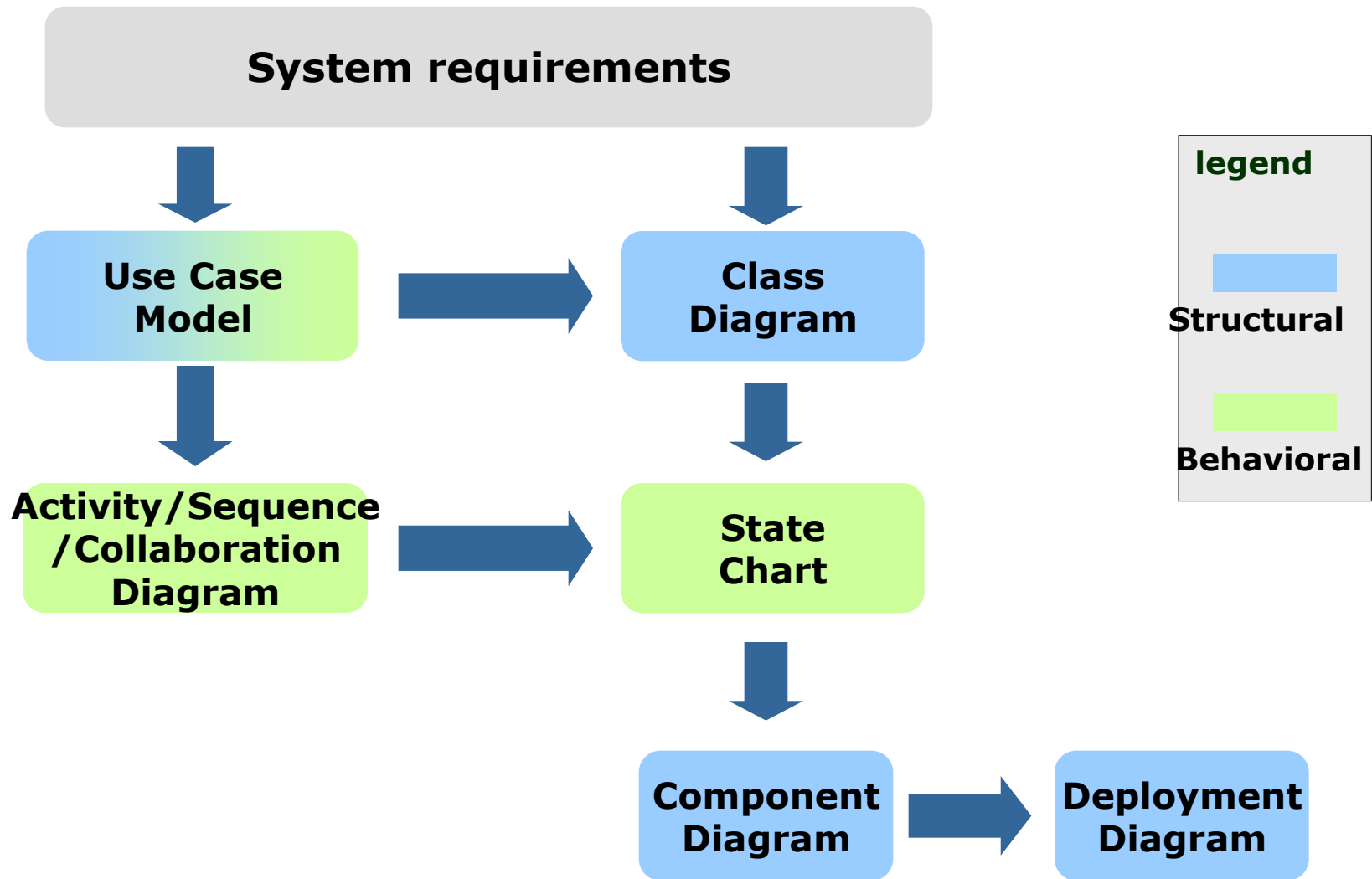
■ Structural Diagrams

- focus on static aspects of the software system
- Class, Object, Package, Component, Deployment

■ Behavioral Diagrams

- focus on dynamic aspects of the software system
- Use-case, Sequence, Collaboration, State Chart, Activity

Design Process



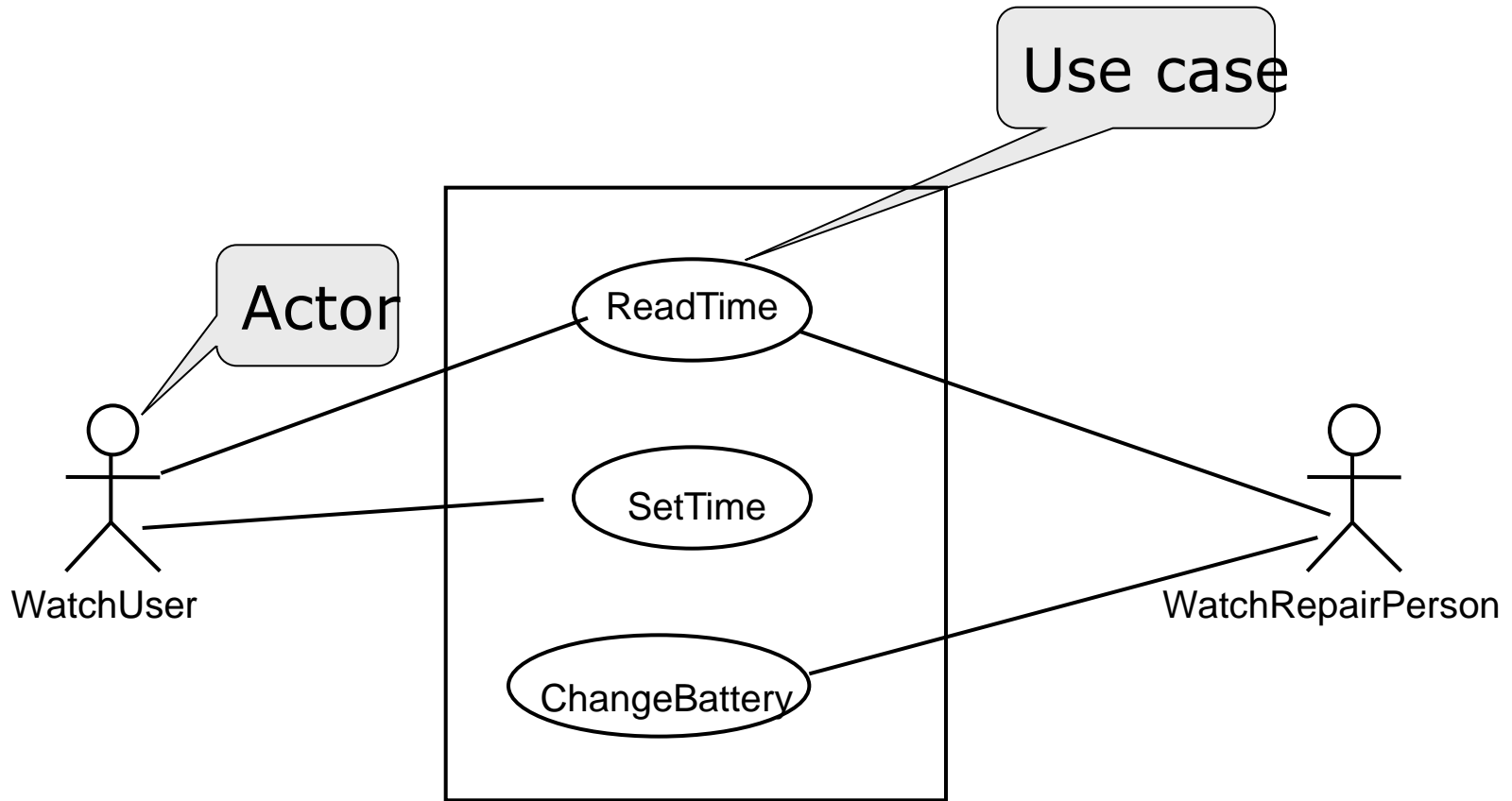
Interaction Diagrams

- Show how the software can be used/maintained
 - Usecase diagram
- Show how objects interact with one another
 - Sequence diagram
 - Collaboration diagram
- Show the major activities
 - Activity diagram
- Show the states of a few important classes/objects
 - State/Statechart diagram

UML First Pass

- Use case Diagrams
 - Describe the functional behavior of the system as seen by the user.
- Class diagrams
 - Describe the static structure of the system: Objects, Attributes, Associations
- Sequence diagrams
 - Describe the dynamic behavior between actors and the system and between objects of the system
- Statechart diagrams
 - Describe the dynamic behavior of an individual object (essentially a finite state automaton)
- Activity Diagrams
 - Model the dynamic behavior of a system, in particular the workflow (essentially a flowchart)

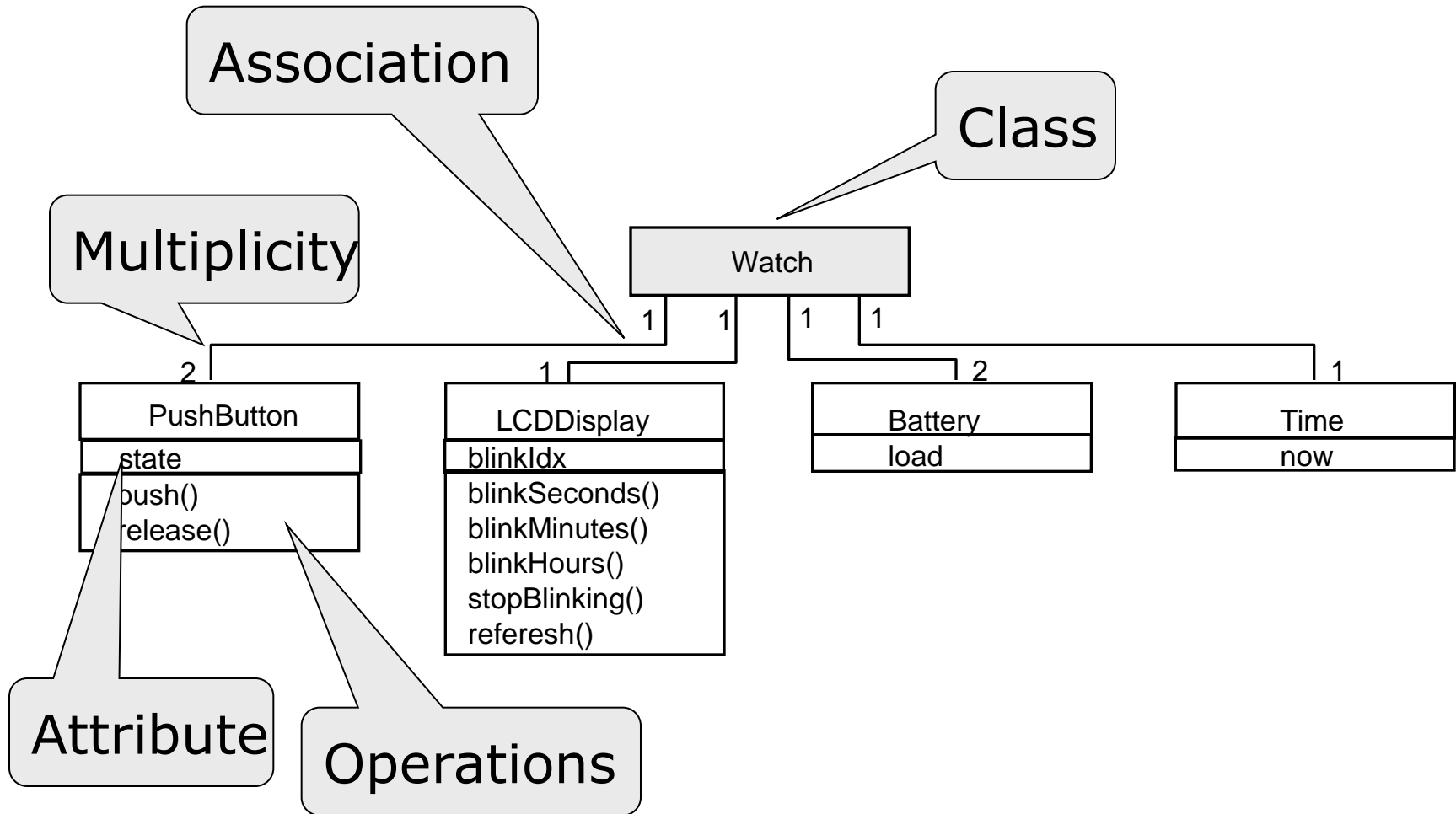
UML first pass: Use case diagrams



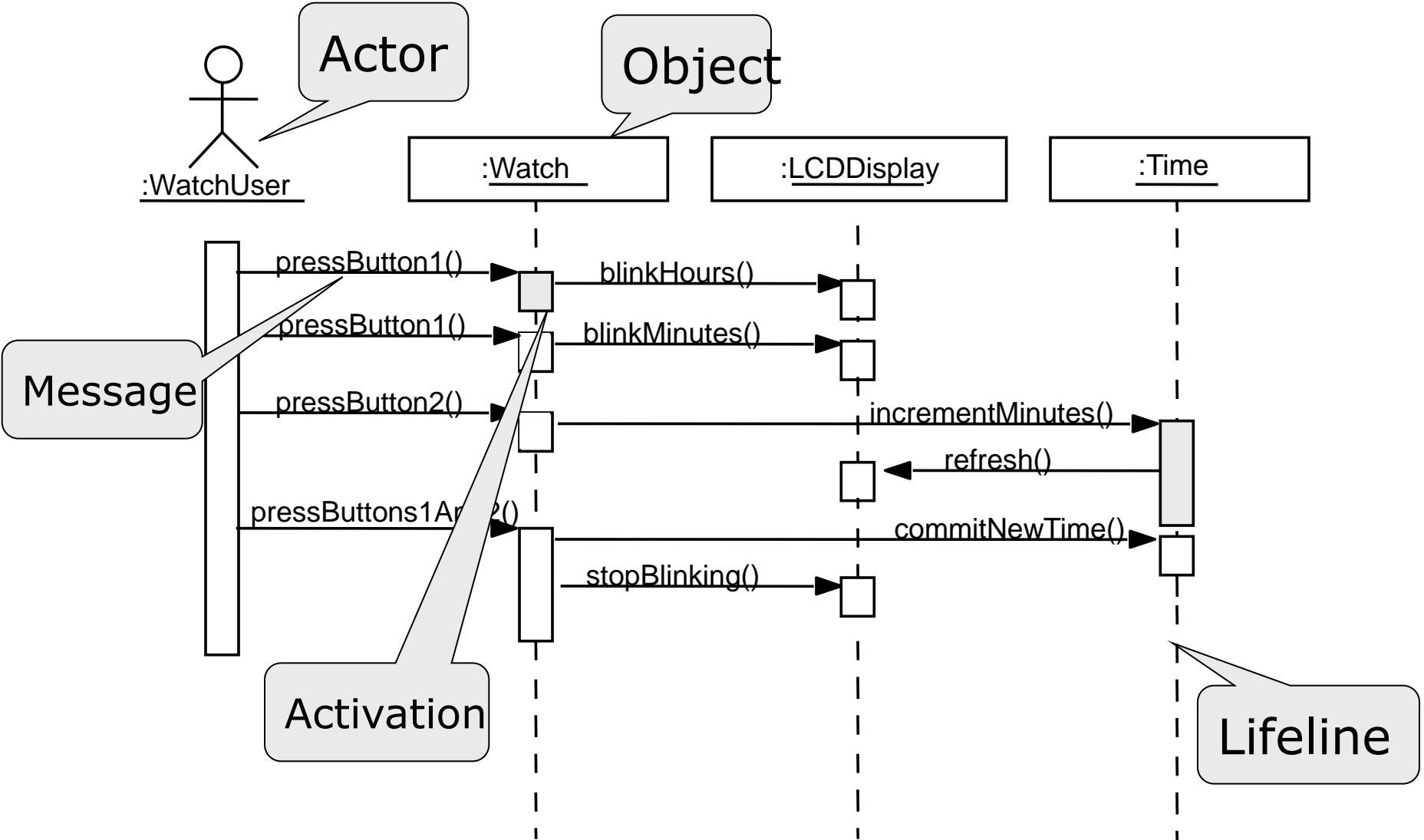
Use case diagrams represent the functionality of the system from user's point of view

UML first pass: Class diagrams

Class diagrams represent the structure of the system

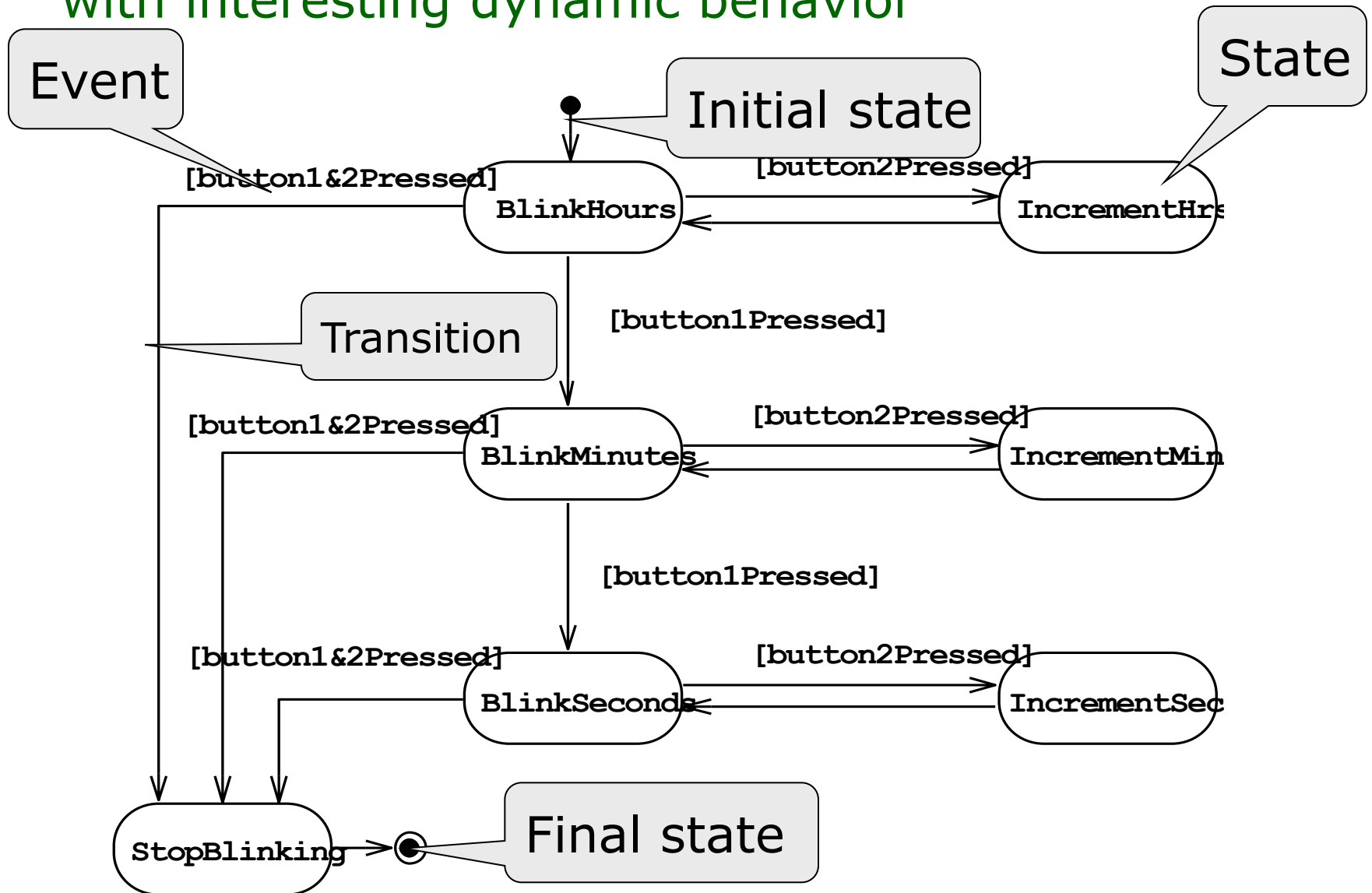


UML first pass: Sequence diagram



Sequence diagrams represent the behavior as interactions. Here it is giving details of incrementing minutes in setTime module

UML first pass: Statechart diagrams for objects with interesting dynamic behavior

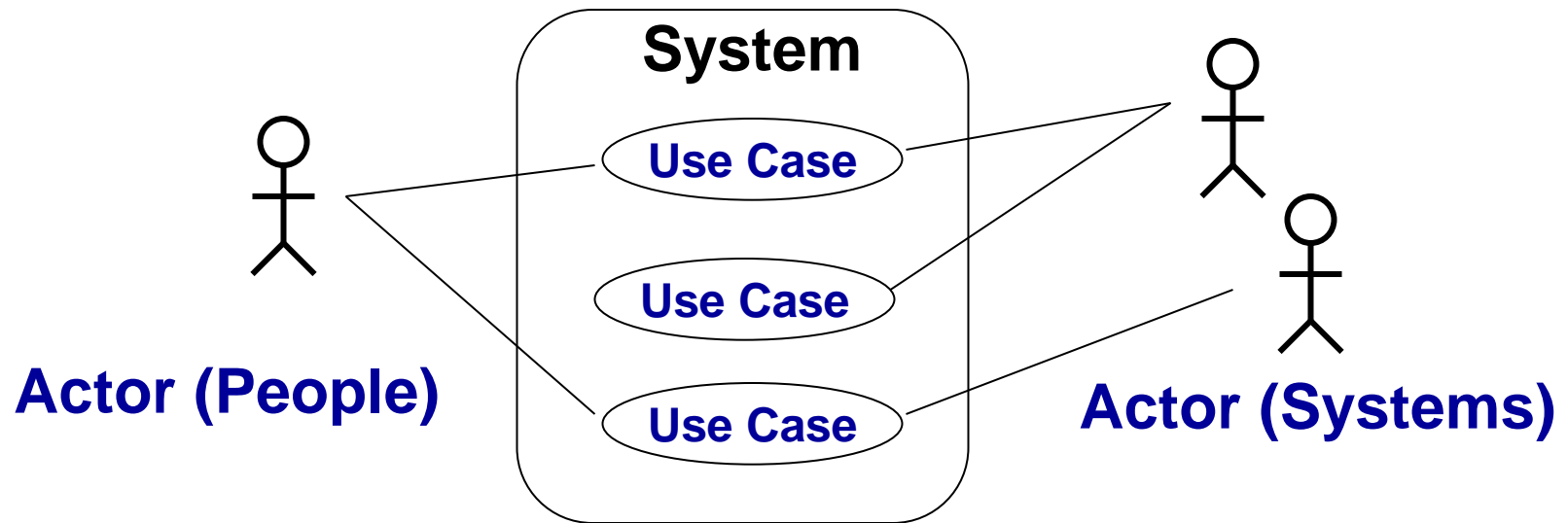


Represent behavior as states and transitions

Usecase diagram

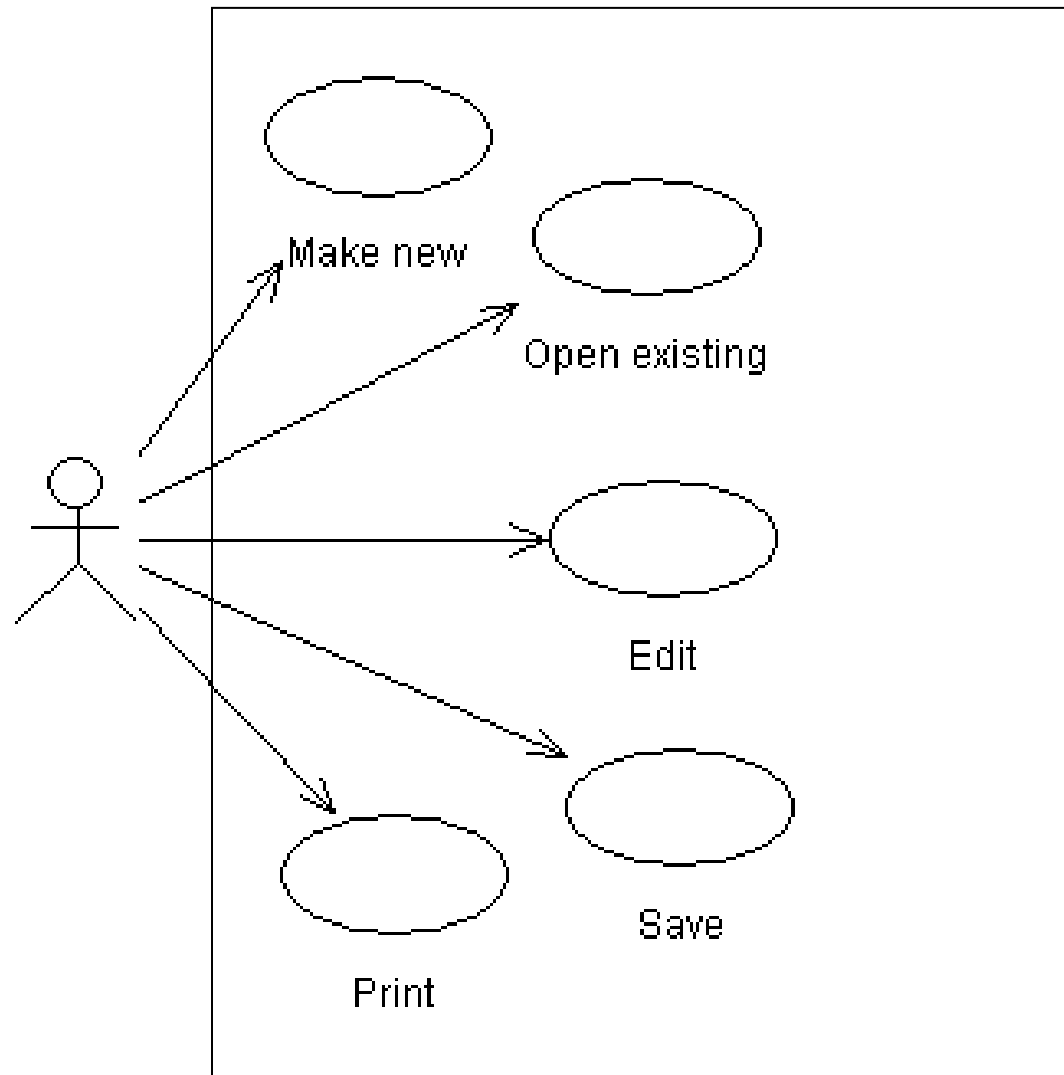
Use Cases

- Two types of Actors: Users and System administrators



Use case examples

(use cases for powerpoint.)

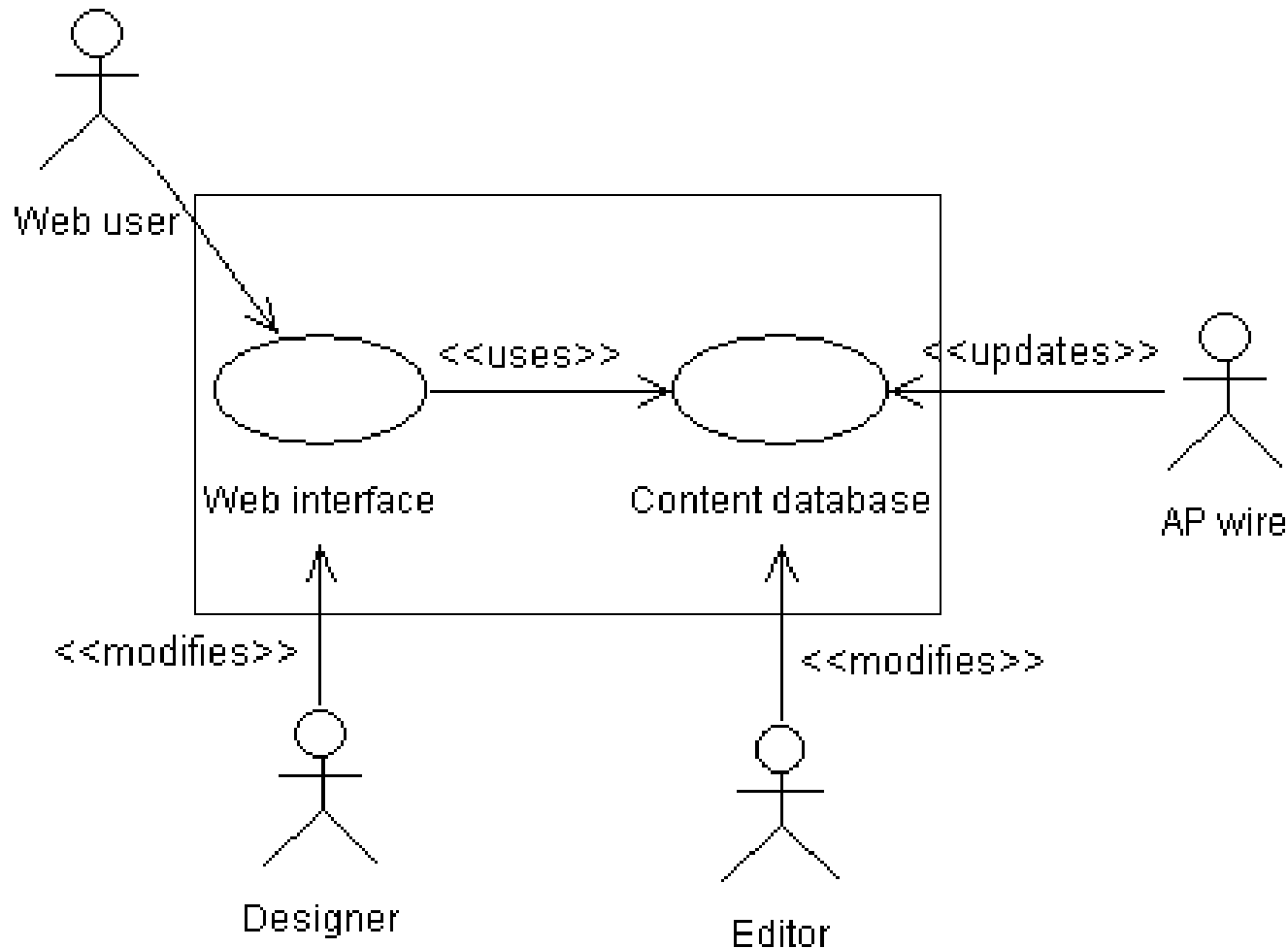


About the last example...

- Gives a view of powerpoint.
- focusses your attention to the key features

Use case examples

(Relationships in a news web site.)



About the last example...

- The last is more complicated and realistic use case diagram. It captures several key use cases for the system.
- Note the multiple actors. In particular, 'AP wire' is an actor, with an important interaction with the system, but is not a person (or even a computer system, necessarily).
- The notes between << >> marks are *stereotypes*: make the diagram more informative.

Usecase diagram

Give a Usecase diagram for an ATM machine:

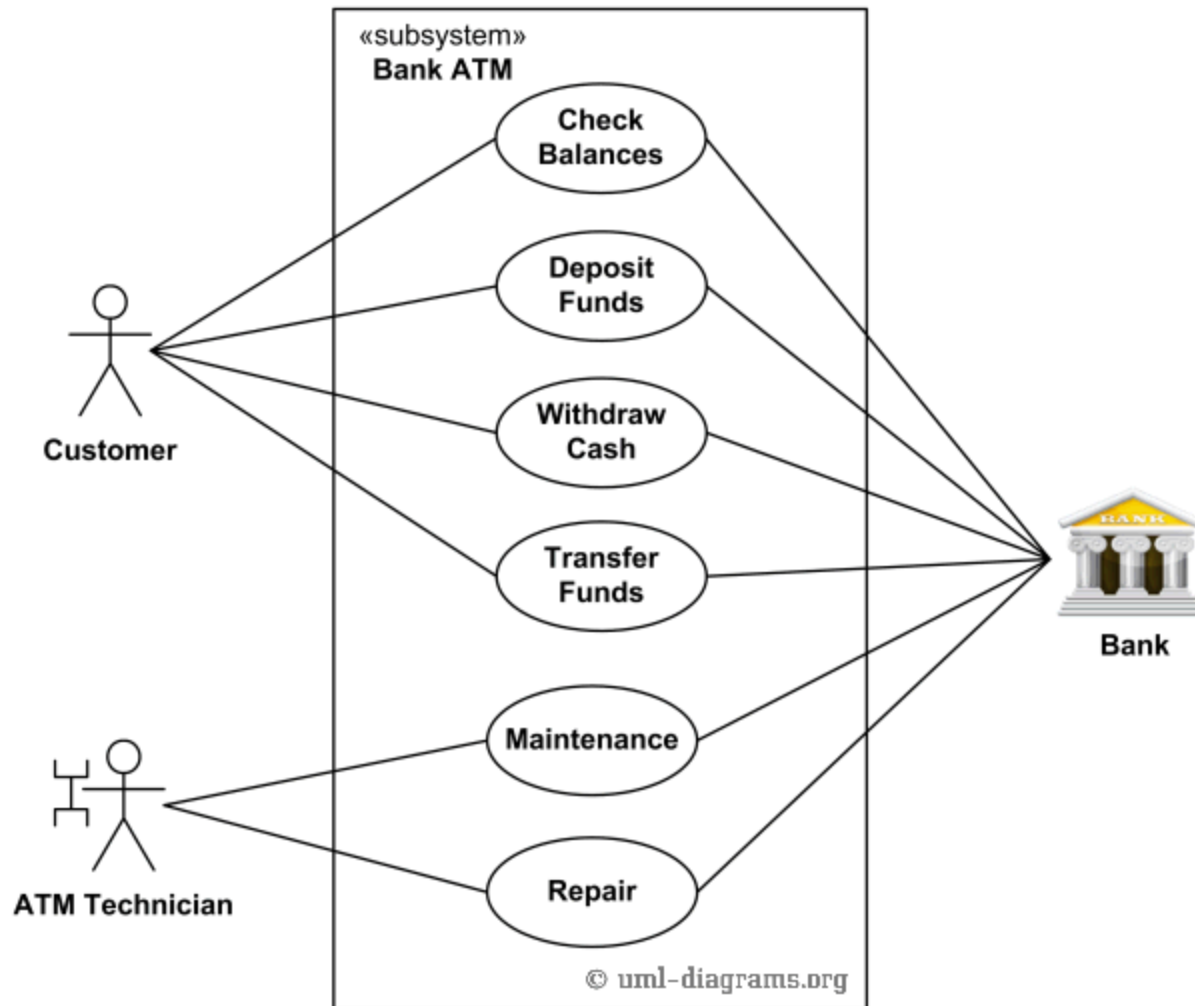
An automated teller machine (**ATM**) provides bank customers with access to financial transactions.

Customer uses bank ATM to *Check Balances* of his/her bank accounts, *Deposit Funds*, *Withdraw Cash* and/or *Transfer Funds*

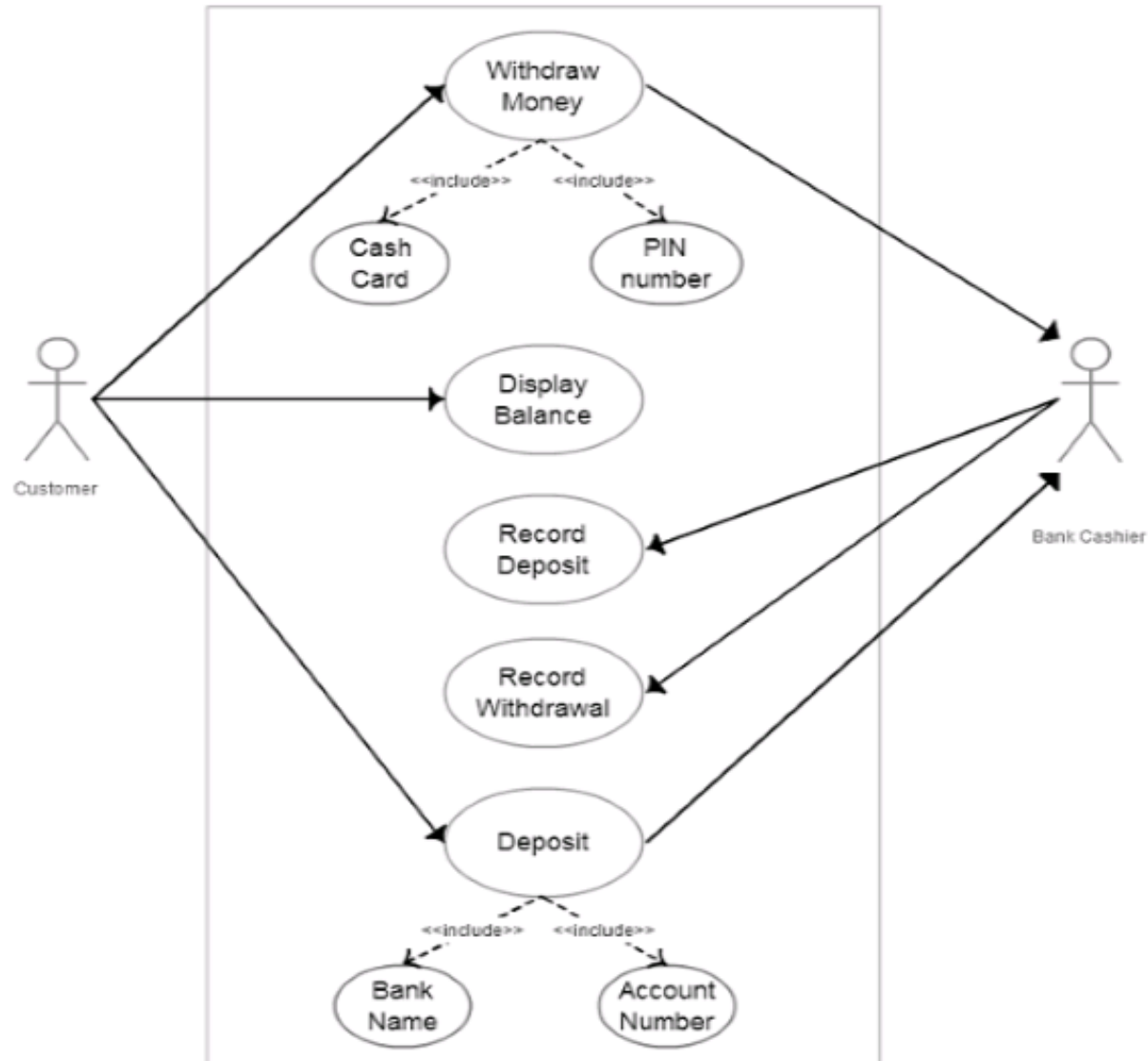
ATM Technician provides *Maintenance* and *Repairs*.

Bank actor: customer transactions or to the ATM servicing.

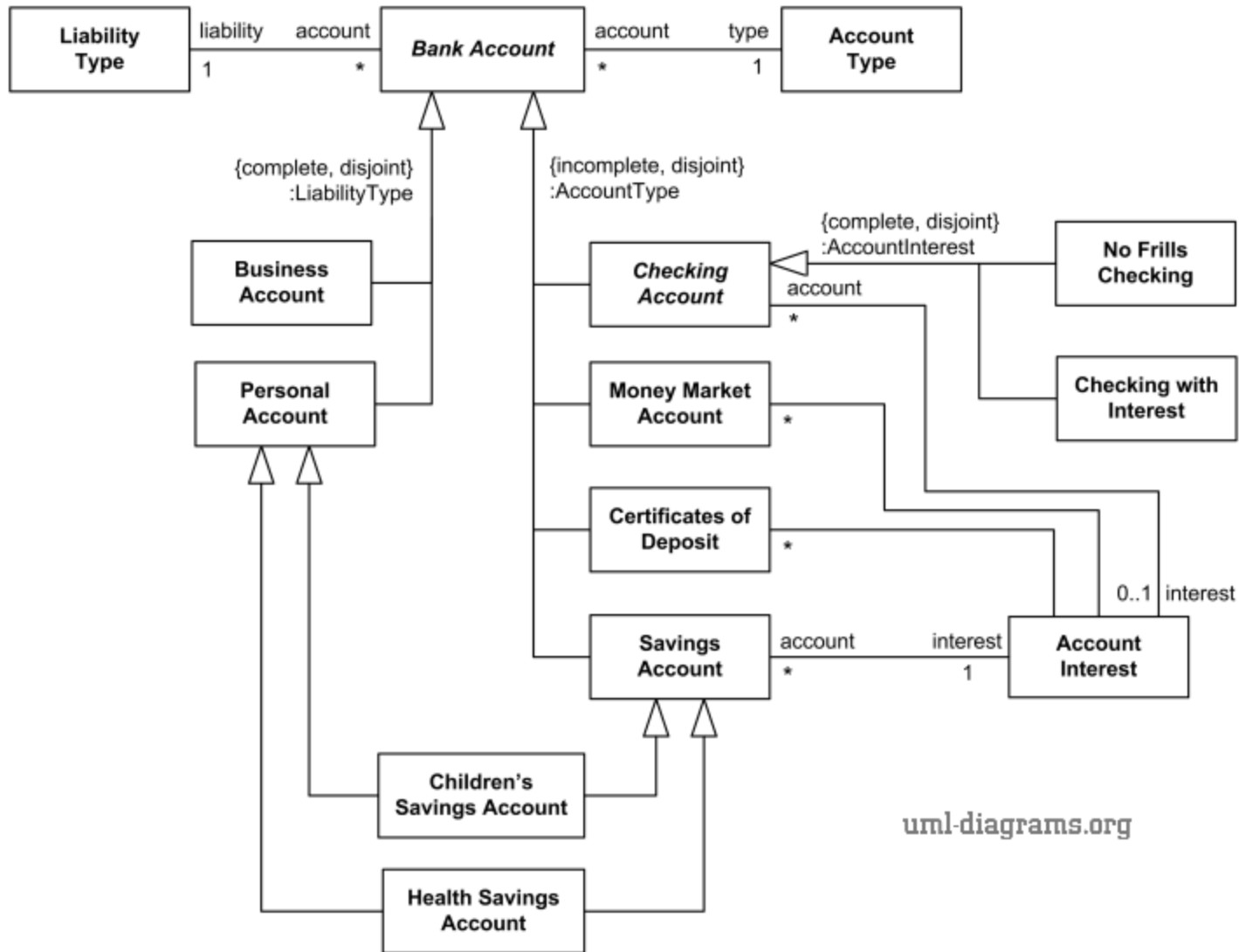
Usecase diagram for an ATM machine



Usecase diagram for an ATM machine



Class diagram



Practice problems

- Modify the Usecase diagram in Slide 7 to incorporate:
 - SetHours, setMinutes and setSeconds
 - SetTime will be a usecase which <<uses>> the usecases listed above
- Modify the usecase diagram in slide 18 to incorporate:
 - Two type of users: Bank customers and non-bank customers. Non-Bank customers can only withdraw and check balance
- Modify the Class diagram in slide 20 to make it possible to make code in Java. This requires multiple inheritance to be removed. Possible mechanism: Have multilayered hierarchy with duplication of classes.

Package diagram