

USE CASES

Software Requirements

Requirements (Functional)

- **Defining what the system should do**
 - What the clients needs (as opposed to wants)
 - Not how the solution should be designed or implemented
- **Many sources of requirements**
 - Interviews (structured vs. non-structured)
 - Stakeholder documents
 - Questionnaires
 - Existing or similar systems
 - Standards in that domain
 - Rapid prototypes
- **Often information is documented in text**
- **From these build OO requirements models**

Gathering Requirements

- **We recognize three iterative activities :**
 - Elicitation: capturing information from sources
 - Documentation: “putting it on paper”
 - Validation: confirming it meets users’ needs
- **Analysis (or definition) versus Specification**

What is Visual Modeling?

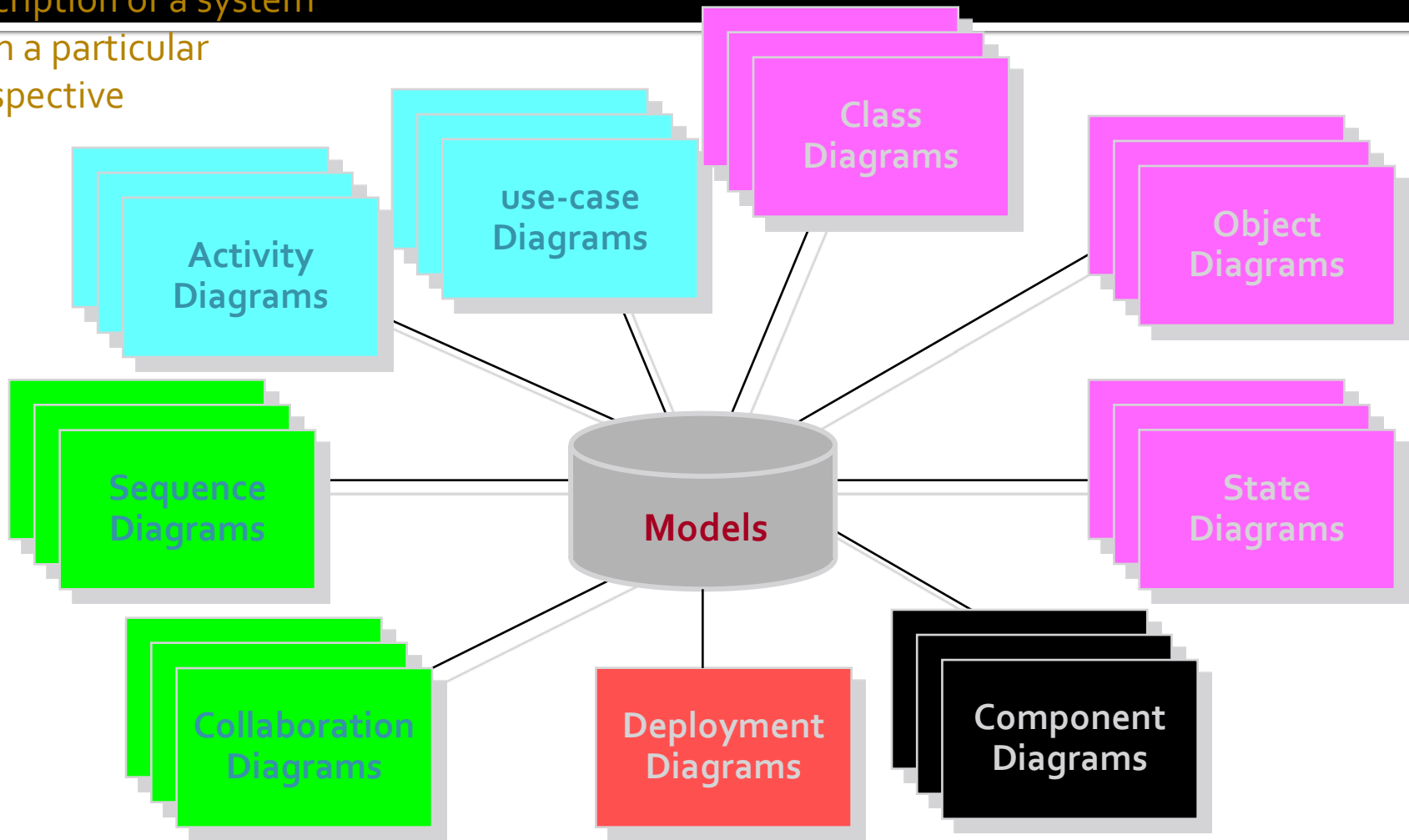
- *Visual modeling* is a way of thinking about problems using models organized around real-world ideas.
- Models are useful for
 - Understanding problems
 - Communicating with everyone involved with the project (customer, domain expert, analyst, designers, etc.)
 - Modeling enterprises
 - Preparing documentation
 - Designing programs and databases

What is the UML?

- The UML is a language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system
- Process and implementation independent
 - It can be used with a large number of processes, throughout the development life cycle, and across OO implementation technologies
- The standard for OO systems

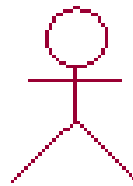
Diagrams Are Views of a Model

A model is a complete description of a system from a particular perspective



Actors

- An *actor* is anyone or anything that must interact with the system.
- Actors are NOT part of the system
 - Only input information to the system
 - Only receive information from the system
 - Both input to and receive information from the system
- In the UML, an actor is represented as a stickman.



Actor

Questions to Discover Actors?

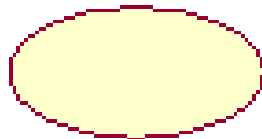
- Who is interested in a certain requirement?
- Where in the organization is the system used?
- Who will benefit from the use of the system?
- Who will supply the system with this information, use this information, and remove this information?
- Who will support and maintain the system?
- Does the system use an external resource?
- Does one person play several different roles?
- Does several people play the same role?
- Does the system interact with the legacy system?

Use Cases

- A *use case* is a description of a set of sequences of actions that a system performs to yield an observable result of value to an actor.
- A use case describes what a system does but it does not specify how it does it.
- A use case typically represents a major piece of functionality that is complete from beginning to end. A use case must deliver something of value to an actor.

Use Cases *(Cont'd)*

- A use case must deliver something of value to an actor.
- The collection of use cases for a system constitute all the defined ways the system may be used.
- In the UML, a use case is represented as an oval.



Use Case

Use Case Relationships

- Between Actor and Use Case
 - Association / Communication
 - Arrow direction indicates who initiates communication



Use Case Relationships cntd...

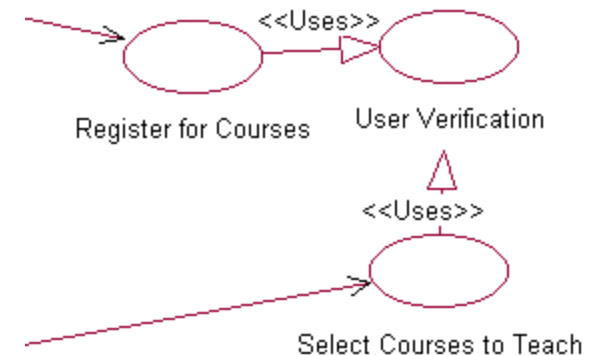
- Between Use Cases (Generalization):

- Uses

- Where multiple use cases share pieces of same functionality

- Extends

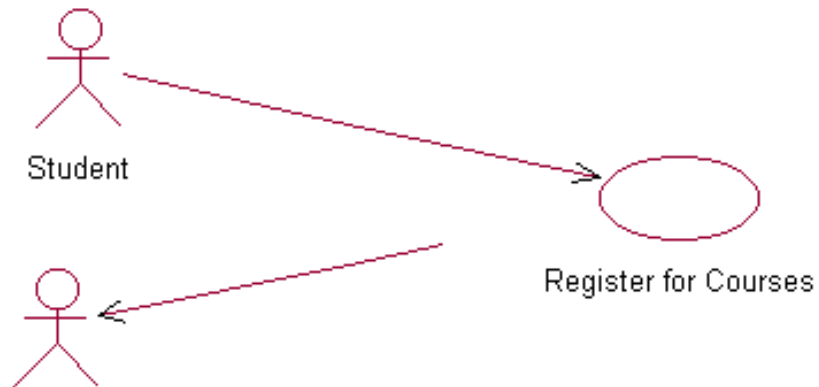
- Optional behavior
 - Behavior only run under certain conditions (such as alarm)
 - Several different flows run base on user selection



Use Case Diagram

- Complete course of events in the system, from the user's perspective
- Use Cases Model: Illustrates
 - (use cases) the system's intended functions
 - (actors) surroundings – external to the system
 - (use case diagrams) relationships between use cases and actors
- Use Case Model is an important communication vehicle between customers (they can understand it!) and developers

The collection of all use cases is everything that can be done to/with the system



ATM: Problem Description

- A bank has several automated teller machines (ATMs), which are geographically distributed and connected via a wide area network to a central server. Each ATM machine has a card reader, a cash dispenser, a keyboard/display, and a receipt printer. By using the ATM machine, a customer can withdraw cash from either a checking or savings account, query the balance of an account, or transfer funds from one account to another. A transaction is initiated when a customer inserts an ATM card into the card reader. Encoded on the magnetic strip on the back of the ATM card are the card number, the start date, and the expiration date. Assuming the card is recognized, the system validates the ATM card to determine that the expiration date has not passed, that the user-entered PIN (personal identification number) matches the PIN maintained by the system, and that the card is not lost or stolen. The customer is allowed three attempts to enter the correct PIN; the card is blocked if the third attempt fails. Cards that have been reported lost or stolen are also blocked.

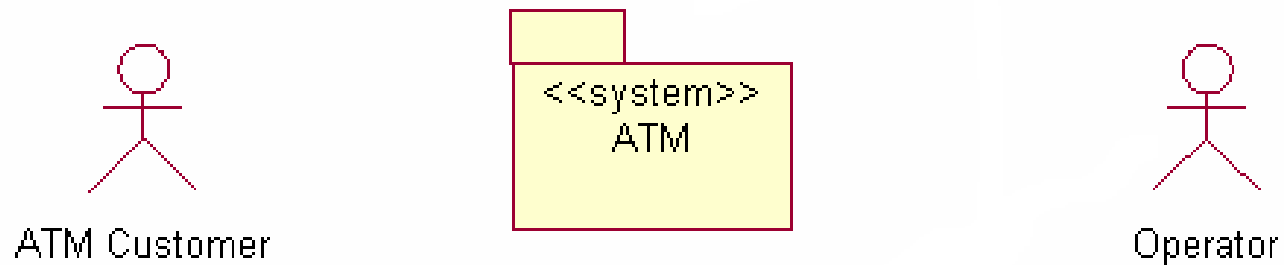
ATM: Problem Description *(Cont'd)*

- If the PIN is validated satisfactorily, the customer is prompted for a withdrawal, query, or transfer transaction. Before a withdrawal transaction can be approved, the system determines that sufficient funds exist in the requested account, that the maximum daily limit will not be exceeded, and that there are sufficient funds at the local cash dispenser. If the transaction is approved, the requested amount of cash is dispensed, a receipt is printed containing information about the transaction, and the card is ejected. Before a transfer transaction can be approved, the system determines that the customer has at least two accounts and that there are sufficient funds in the account to be debited. For approved query and transfer requests, a receipt is printed and the card is ejected. A customer may cancel a transaction at any time; the transaction is terminated and the card is ejected. Customer records, account records, and debit card records are all maintained at the server.

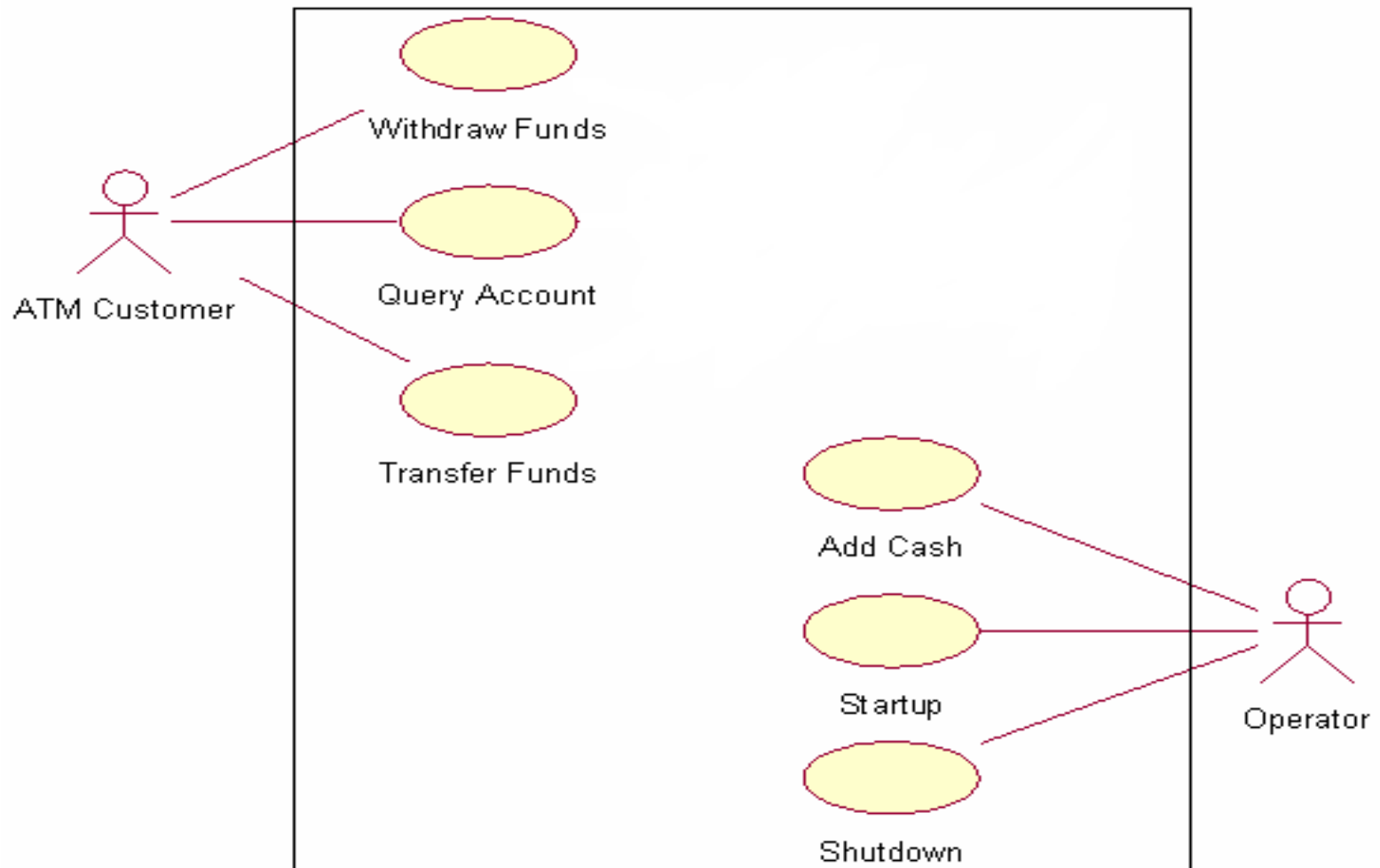
ATM: Problem Description *(Cont'd)*

- An ATM operator may start up and close down the ATM to replenish the ATM cash dispenser and for routine maintenance. It is assumed that functionally to open and close accounts and to create, update, and delete customer and debit card records is provided by an existing system and is not part of this problem.

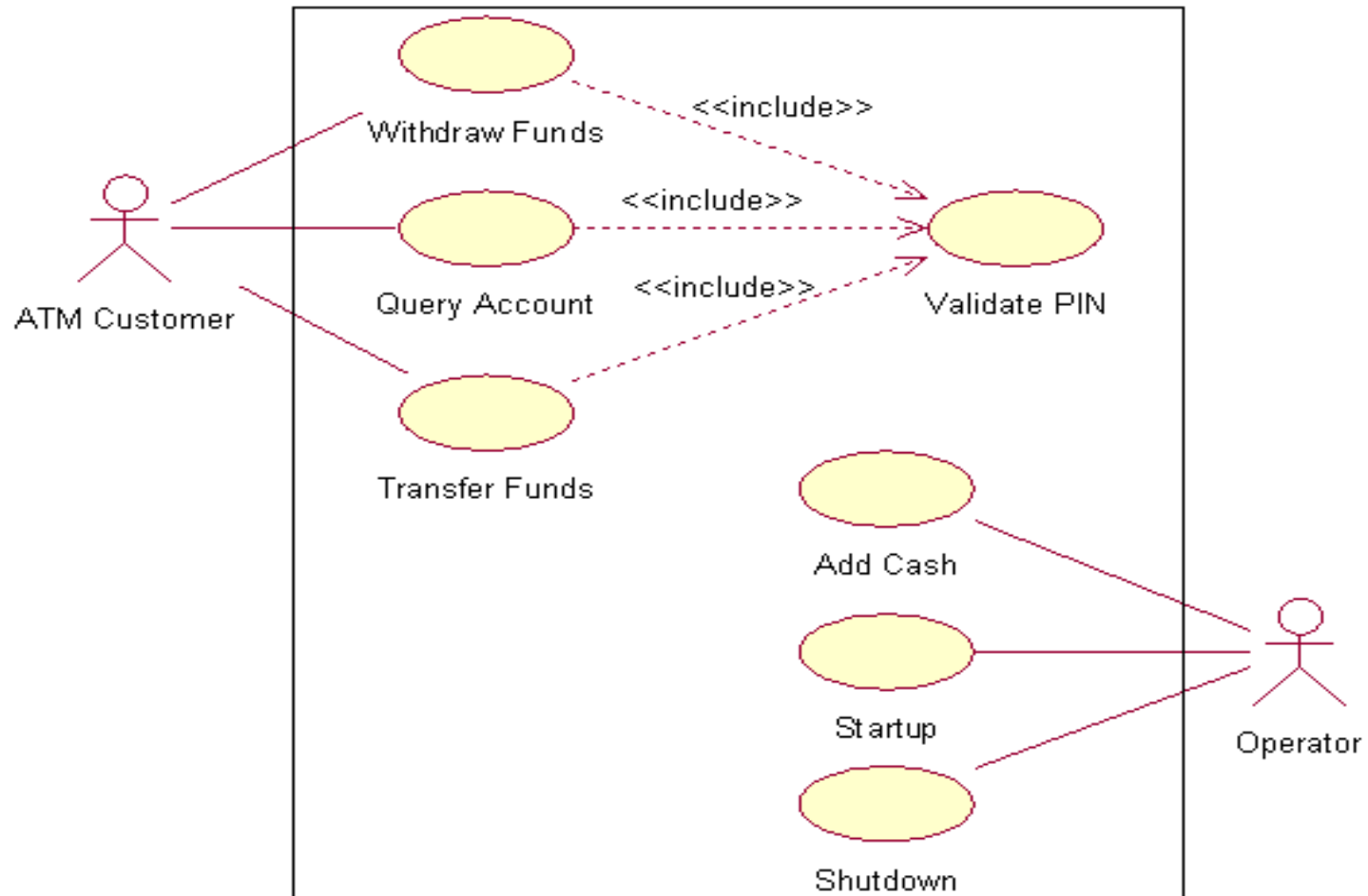
ATM System



ATM: Use Case Model



ATM: Use Case Model



Examples/Exercises

- Library management
- Registration System
- Recycling machine
- Order Processing

Use Case Diagram

