Use Cases

Concepts, Specifications, and Diagrams

Introduction

- "Invented" by Ivar Jacobson in the late 1960's (where have we seen his name before?)
- Introduced to the OO community in the late 1980's
- Alistair Cockburn has extended Jacobson's model
- Is a way to specify functional requirements
- Is notated using a **use case specification**
- Is not part of the **Unified Modeling Language (UML)**, but is many times used in conjunction with it

What is a Use Case? (Cockburn)

- A use case captures a contract between the **stakeholders** of a system about its **behavior**.

- Describes the system's behavior under various conditions as the system responds to a request from one of the stakeholders called the **primary actor**.

1. The primary actor initiates some interaction with the system to accomplish some goal.
2. The system responds, *protecting the interests of all of the stakeholders*.
3. Different sequences of behaviors, or **scenarios,** can unfold, depending on the requests and the conditions surrounding the request. The use case gathers these scenarios together.

Use Case Specification: Natural Language Example

Use Case 1. Withdraw Money

The system displays the account types available to be withdrawn from and the user indicates the desired type. The system asks for the amount to be withdrawn and the user specifies it. Next, the system debits the user's account and dispenses the money. The user removes the money, the system prints a receipt, and the user removes the receipt. Then the system displays a closing message and dispenses the user's ATM card. After the user removes his card, the system displays the welcome message.

Use Case Specification Template*

Number		
Name		
Summary		
Priority		
Preconditions		
Postconditions		
Primary Actor(s)		
Secondary Actor(s)		
Trigger		
Main Scenario	**Step**	**Action**
Extensions	**Step**	**Branching Action**
Open Issues		

*Adapted from A. Cockburn, “Basic Use Case Template”

Use Case Specification Template*

Number	*Unique use case number*	
Name	*Brief verb-noun phrase*	
Summary	*Brief summary of use case major actions*	
Priority	*1-5 (1 = lowest priority, 5 = highest priority)*	
Preconditions		
Postconditions		
Primary Actor(s)		
Secondary Actor(s)		
Trigger		
Main Scenario	**Step**	**Action**
Extensions	**Step**	**Branching Action**
Open Issues		

*Adapted from A. Cockburn, “Basic Use Case Template”

Use Case Specification Template*

Number	
Name	
Summary	
Priority	
Preconditions	*What needs to be true before the use case "executes"*
Postconditions	*What will be true after the use case successfully "executes"*
Primary Actor(s)	

Precondition: y != 0

Postcondition: x / y

```
double divide(double x, double y) {
    return (x / y);
}
```

Precondition: None

Postcondition: if y==0 "Illegal", else x / y

```
double divide(double x, double y) {
    if (y == 0) cout << "Illegal\n";
        else return (x / y);
}
```

Extensions	**Step**	**Branchi**
Open Issues		

*Adapted from A. Cockburn, "Basic Use Case Template"

Use Case Specific

Number		
Name		
Summary		
Priority		
Preconditions		
Postconditions		
Primary Actor(s)	*Primary actor name(s)*	
Secondary Actor(s)	*Secondary actor name(s)*	
Trigger		
Main Scenario	**Step**	**Action**
Extensions	**Step**	**Branching Action**
Open Issues		

Actor

- Anyone or anything with behavior
- May be a person or system
- **Primary**: The stakeholder who or which initiates an interaction with the system to achieve a goal. Is generally a category of individuals (a **role**).
- **Secondary**: Provides a service to the system. Is almost never a person.

*Adapted from A. Cockburn, “Basic Use Case Template”

Use Case Specification Template*

Number		
Name		
Summary		
Priority		
Preconditions		
Postconditions		
Primary Actor(s)		
Secondary Actor(s)		
Trigger	*The action that caused the use case to be invoked*	
Main Scenario	**Step**	**Action**
	Step #	*This is the "main success scenario" or "happy path"*
	Step #	*Description of steps in successful use case* "execution"
	Step #	*This should be in a "system-user-system, etc." format*
Extensions	**Step**	**Branching Action**
Open Issues		

*Adapted from A. Cockburn, "Basic Use Case Template"

Use Case Specification Template*

Number		
Name		
Summary		
Priority		
Preconditions		
Postconditions		
Primary Actor(s)		
Secondary Actor(s)		
Trigger		
Main Scenario	**Step**	**Action**
Extensions	**Step**	**Branching Action**
	Step #	*Alternative paths that the use case may take*
Open Issues		

Extension

- Could be an optional path(s)
- Could be an error path(s)
- Denoted in use case diagrams (UML) by **<<extend>>**

*Adapted from A. Cockburn, “Basic Use Case Template”

Use Case Specification Template*

Number		
Name		
Summary		
Priority		
Preconditions		
Postconditions		
Primary Actor(s)		
Secondary Actor(s)		
Trigger		
Main Scenario	**Step**	**Action**
Extensions	**Step**	**Branching Action**
Open Issues	*Issue #*	*Issues regarding the use case that need resolution*

*Adapted from A. Cockburn, “Basic Use Case Template”

Use Case Specification Template*

Number	*Unique use case number*	
Name	*Brief noun-verb phrase*	
Summary	*Brief summary of use case major actions*	
Priority	*1-5 (1 = lowest priority, 5 = highest priority)*	
Preconditions	*What needs to be true before use case "executes"*	
Postconditions	*What will be true after the use case successfully "executes"*	
Primary Actor(s)	*Primary actor name(s)*	
Secondary Actor(s)	*Secondary actor name(s)*	
Trigger	*The action that causes this use case to begin*	
Main Scenario	**Step**	**Action**
	Step #	*This is the "main success scenario" or "happy path."*
	…	*Description of steps in successful use case* "execution"
	…	*This should be in a "system-user-system, etc." format.*
Extensions	**Step**	**Branching Action**
	Step #	*Alternative paths that the use case may take*
Open Issues	*Issue #*	*Issues regarding the use case that need resolution*

*Adapted from A. Cockburn, "Basic Use Case Template"

Use Case Specification Template Example

Number	1
Name	Withdraw Money
Summary	User withdraws money from one of his/her accounts
Priority	5
Preconditions	User has logged into ATM
Postconditions	User has withdrawn money and received a receipt
Primary Actor(s)	Bank Customer
Secondary Actor(s)	Customer Accounts Database

Continued …

Trigger	User has chosen to withdraw money	
Main Scenario	**Step**	**Action**
	1	System displays account types
	2	User chooses account type
	3	System asks for amount to withdraw
	4	User enters amount
	5	System debits user's account and dispenses money
	6	User removes money
	7	System prints and dispenses receipt
	8	User removes receipt
	9	System displays closing message and dispenses user's ATM card
	11	User removes card
	10	System displays welcome message
Extensions	**Step**	**Branching Action**
	5a	System notifies user that account funds are insufficient
	5b	System gives current account balance
	5c	System exits option
Open Issues	1	Should the system ask if the user wants to see the balance?

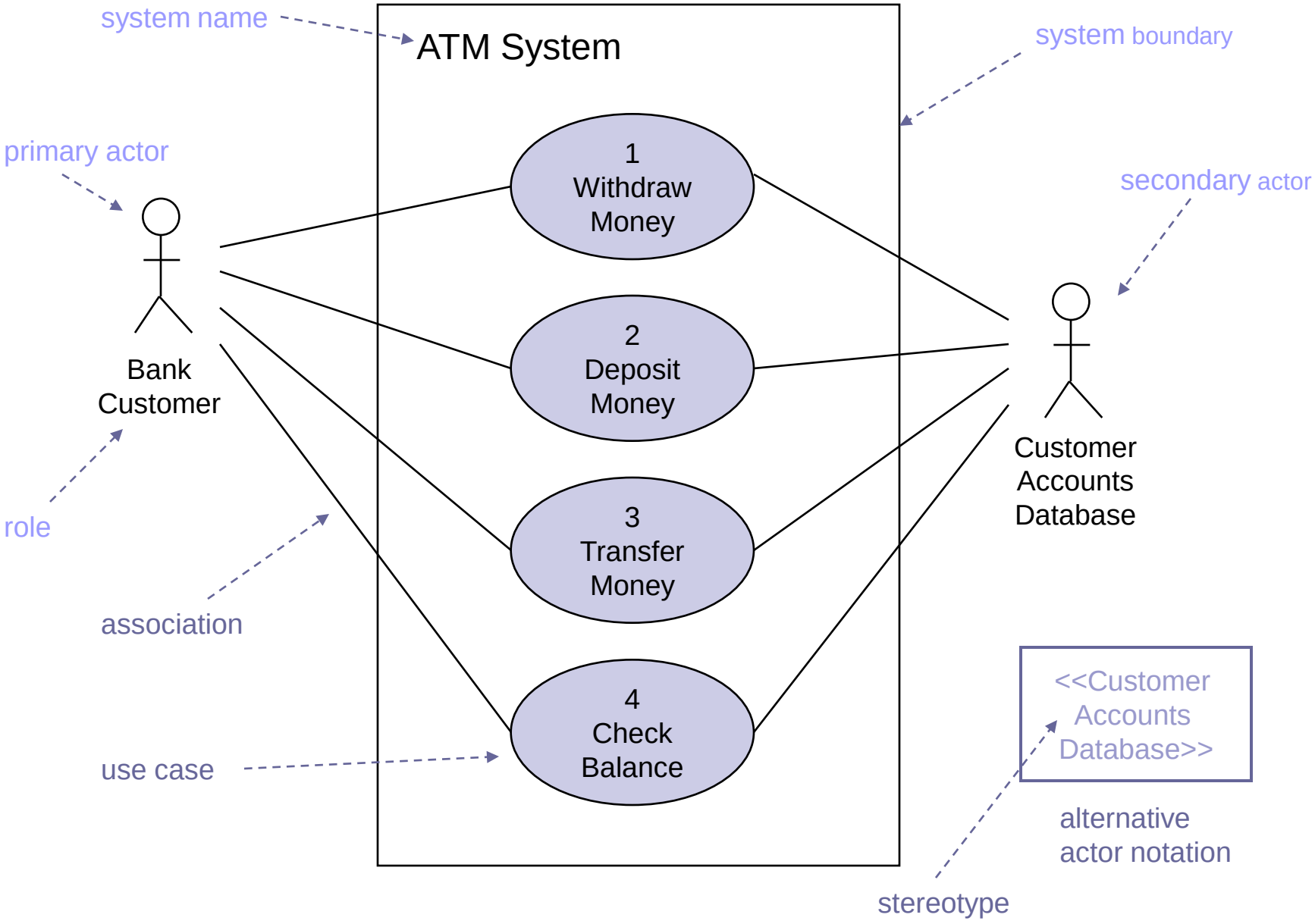
Specification Writing Guidelines

- No trace of design
- Describes what the use case will do, *not* how it will do it (e.g., UI type is irrelevant)
- A dialogue between the user and the system
- Complete, clear, and consistent

Use Case Diagrams

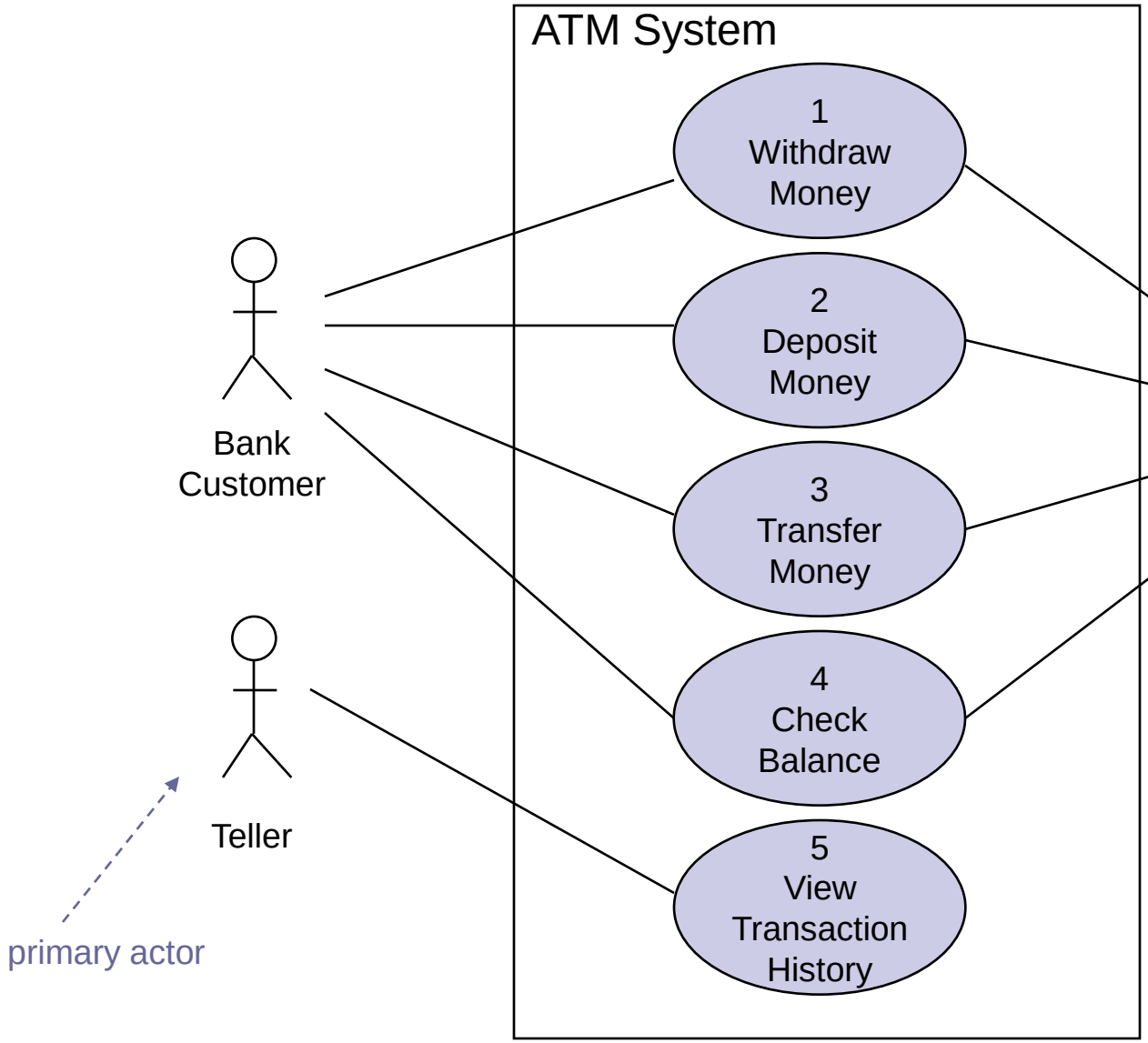
- A way of visualizing the relationships
  - between actors and use cases
  - among use cases

- “A graphical table of contents for the use case set” (Fowler)


system name
ATM System
system boundary
primary actor
1
Withdraw
Money
secondary actor
2
Deposit
Money
Bank
Customer
Customer
Accounts
Database
role
3
Transfer
Money
association
4
Check
Balance
use case
<<Customer
Accounts
Database>>
alternative
actor notation
stereotype

Using Use Case Specifications in Conjunction with Use Case Diagrams

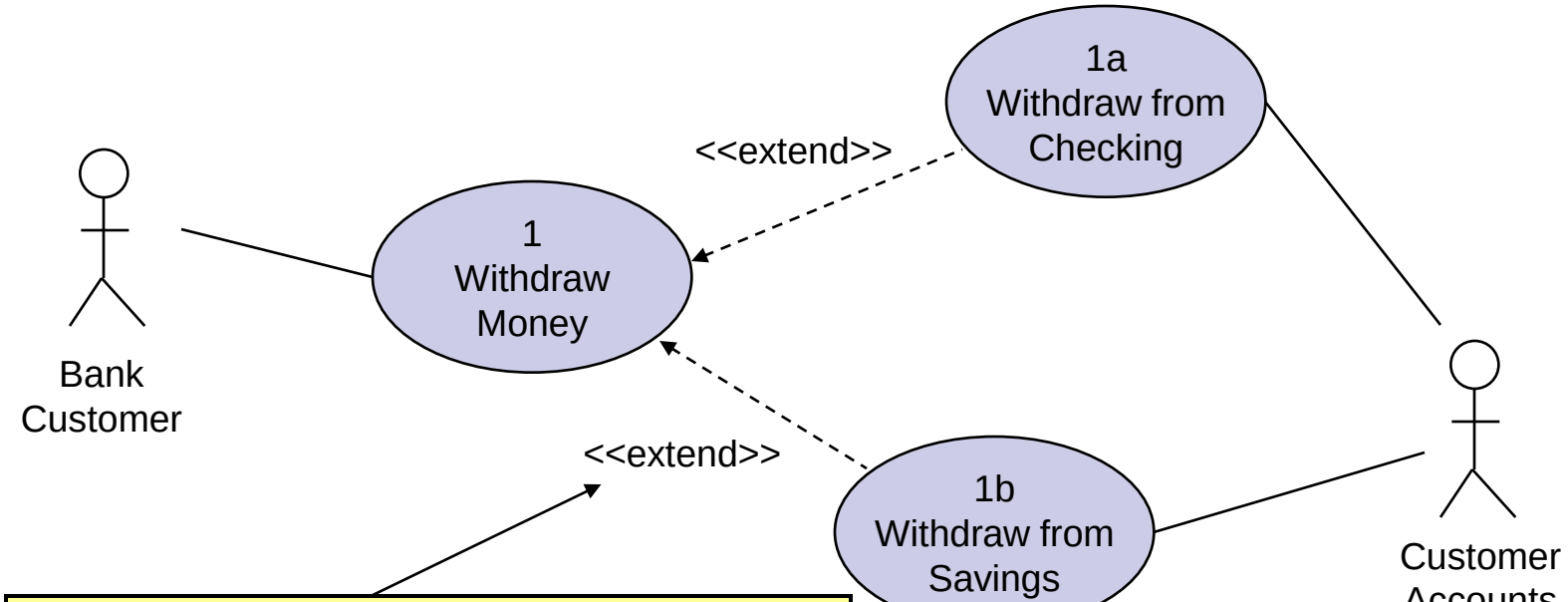
- UML is a graphical modeling tool only.
- Use case specifications are not part of the UML
- But, since each ellipse in a UML use case diagram represents a functional requirement, it may in turn have an associated use case specification.



Why can't a Teller do the things that a Bank Customer can do? Especially if he *is* a customer?

He can. But he must "step into" the role of a Bank Customer.

Sub-use Case Diagram



This is an **extend dependency**.

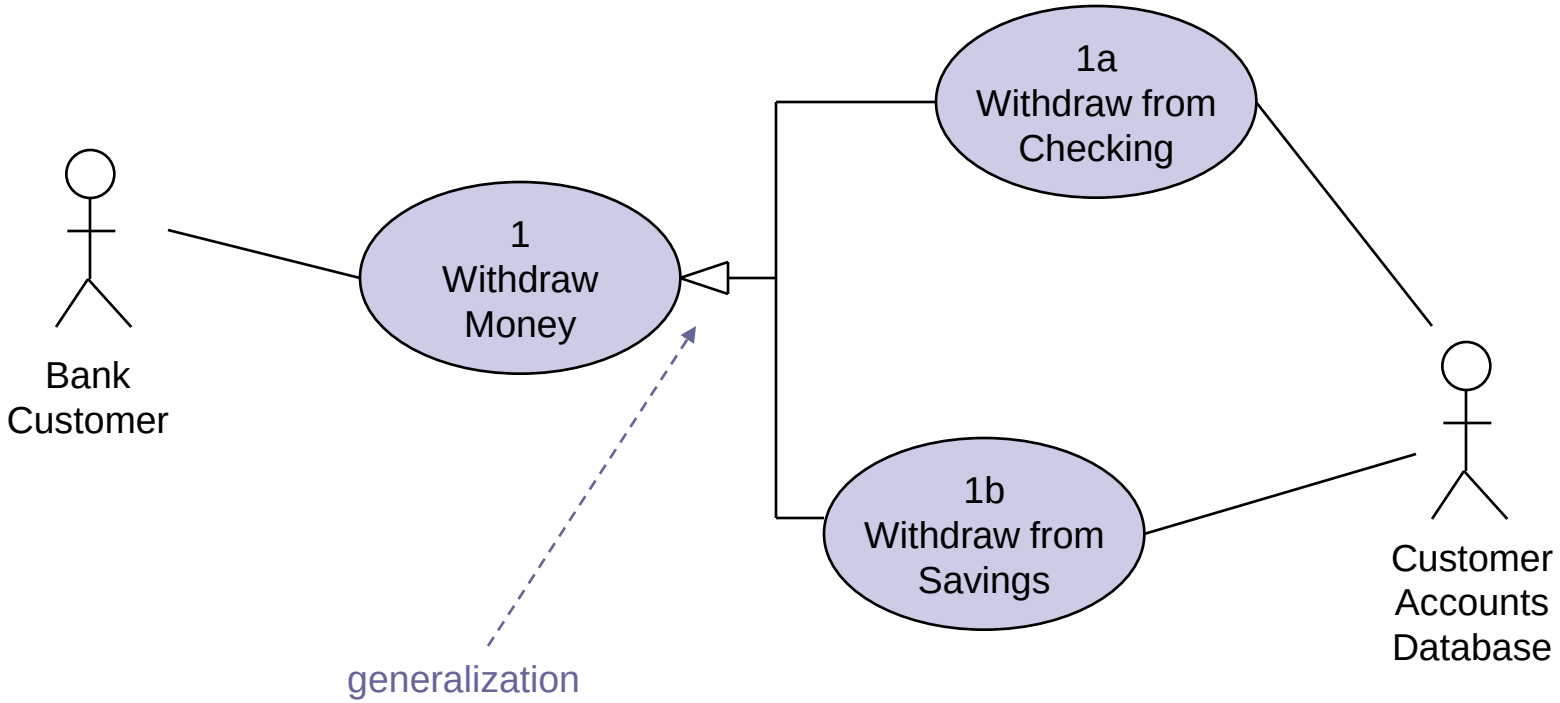
It indicates that use case 1b is part of use case 1, but it may or may not be invoked.

The same is true of use case 1a.

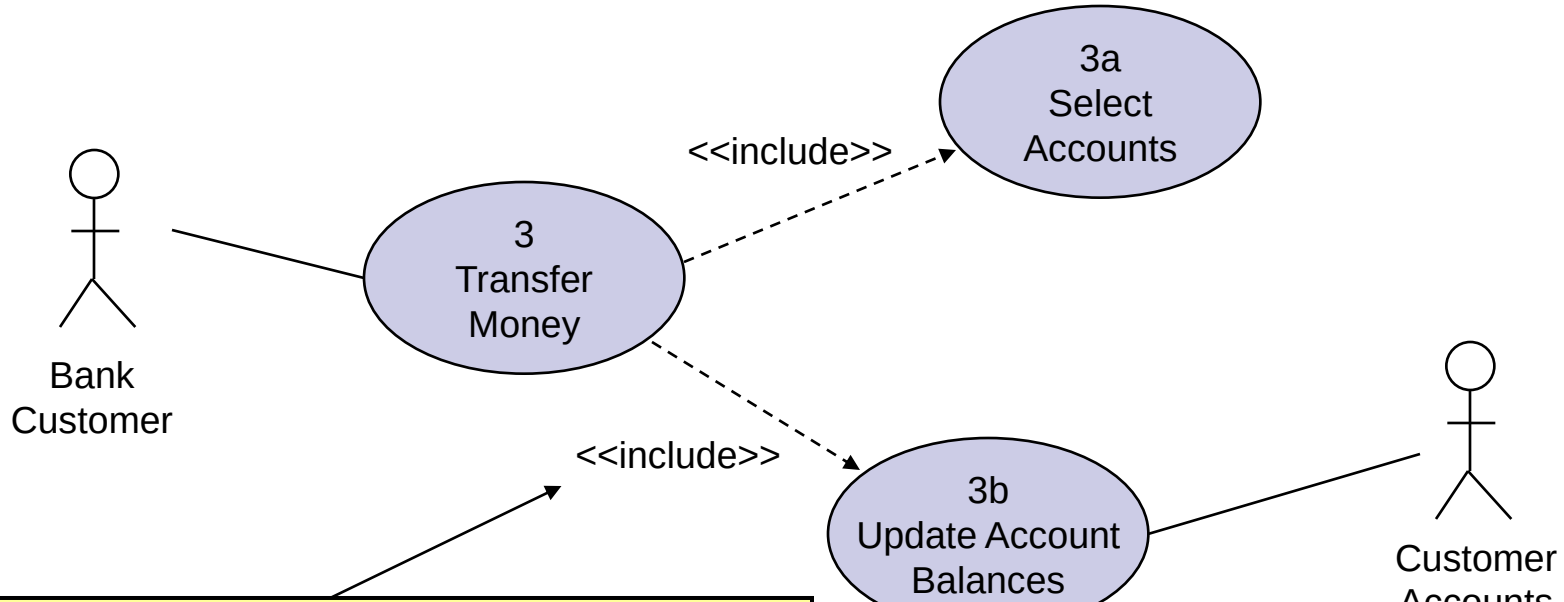
All dependencies are extend unless stereotyped otherwise.

note/comment

Sub-use Case Diagram



Sub-use Case Diagram



This is an **include dependency**.

It indicates that use case 3b is "included" in use case 3 and will be invoked.

The same is true of use case 3a.

References

- Cockburn, A., *Writing Effective Use Cases*. New York: 2001, Addison-Wesley.
- Cockburn, A., *Resources for Writing Use Cases*. http://alistair.cockburn.us/index.php/Resources_for_writing_use_cases, accessed 9/18/07.
- Cockburn, A., *Basic Use Case Template*. 1998, Humans and Technology.
- Cockburn, Alistair, WWW home page, http://alistair.cockburn.us/index.php/Main_Page
- Fowler, M., *UML Distilled*. 3rd ed. 2004, New York: Addison Wesley.
- Fowler, M., WWW home page, http://martinfowler.com
- Jacobson, Ivar, WWW home page, http://www.ivarjacobson.com/locales/ivars-corner.cfm