Work Related Project (CO599)

Week 5 – Modelling Interactions of Users with the System

Previously……

- **Whaler Project**
- Elicitation and analysis of information from different sources using different techniques
- With a view to establishing, specifying and documenting requirements
- From which we can start designing and developing an appropriate “system” *(meeting the functional and non-functional requirements)*

Justifying our Design

- **Make reference to:**
  - **Why the proposed design is fit-for-purpose**
  - **In relation to the requirements**
  - **In comparison to other Websites/systems**
  - **Applying good principles of Website design (with citation of relevant sources)**
  - **Paying attention to the needs and expectations of different users**

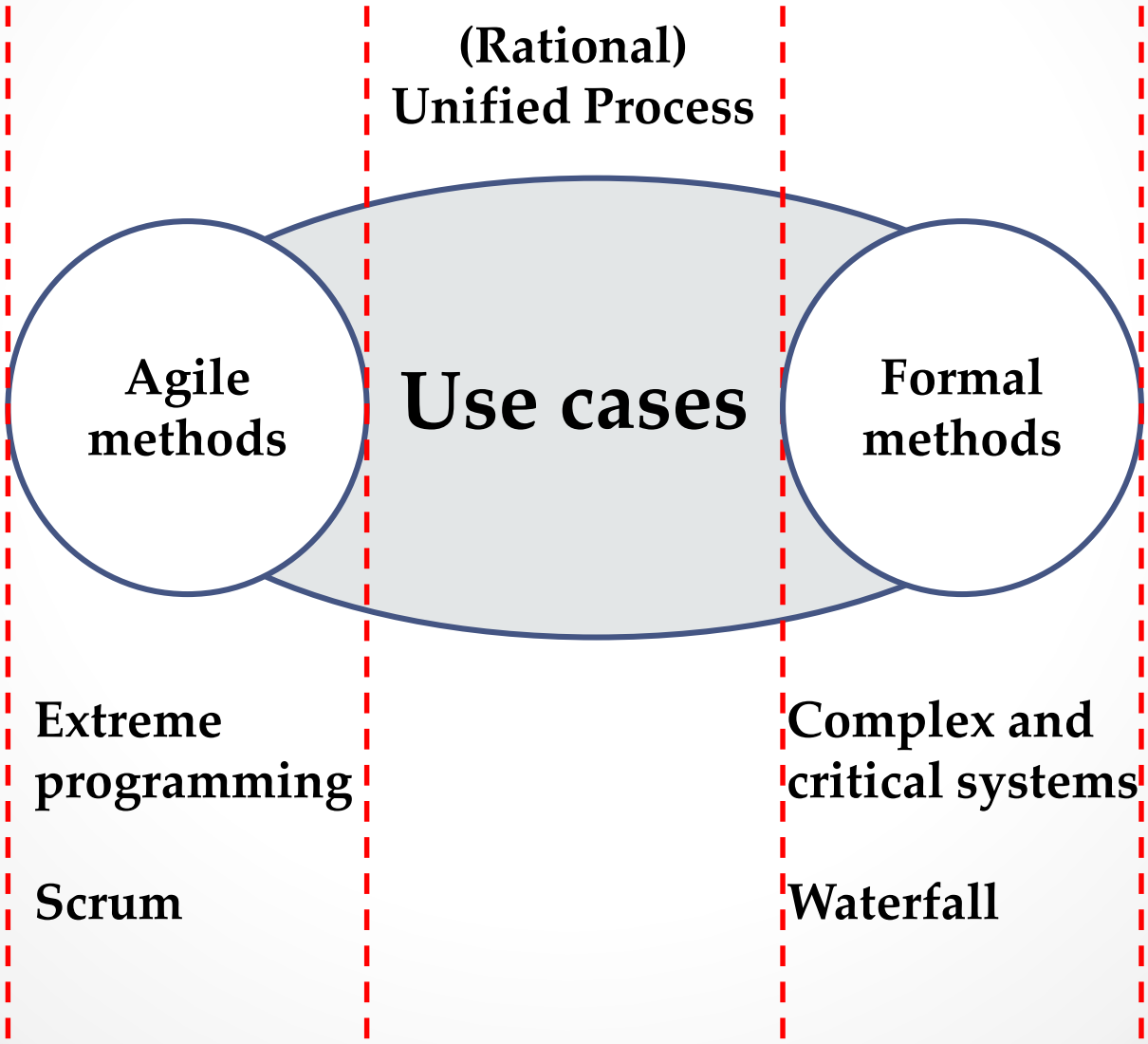
This Week……

- **Continue to develop ideas related to the Whaler Project**
- **Moving on to modelling interactions between the users and the “system”**

The Story So Far

- Business analysts have an important role to play in reviewing the business and justifying potential projects
- They also contribute in the elicitation of requirements
- Previously, we have talked about different techniques for gathering information

Different Approaches for Developing Software Reqs



E-Mail From Project Manager

From: Irene Davies <IreneD@restech.com>
To: The Whaler project requirements development team
cc:
Subject: Requirements development approach

Dear Whaler requirements development team,

I have gone through your analysis and recommendations of the approach for the Whaler project and I wanted to:

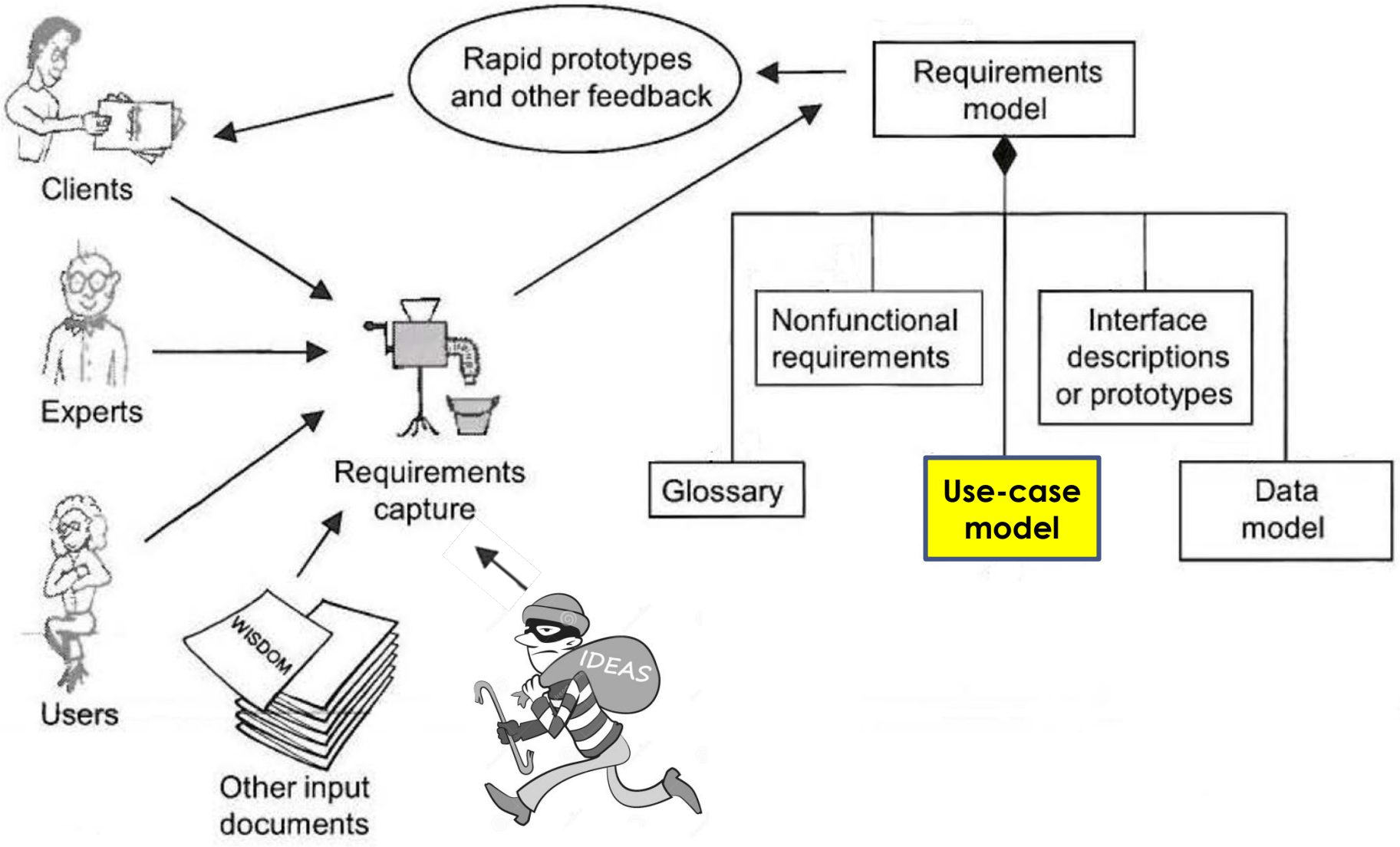
1. Express my appreciation to the team for a job well done.

2. Tell you that I have considered all of your recommendations and I have decided to go with the "use case" method to develop the requirements. This also fits well with the ResTech process for requirements development that we developed after attending the Learning Tree International course User Requirements for Software Development.

Regards,
Irene Davies
Whaler Project Manager

Let us consider use case to be half way between agile and formal.

ResTech's Use-Case Reqs Process



Modeling Processes & Interactions

- **Objectives for you**

- Identify what requirements are represented as use cases
- Describe and understand the process for writing use cases
- List the key elements of a use case
- Write use cases

- **Communication, communication, communication**

Good Communication & Articulation - Identify what is being described

Good Communication & Articulation - Identify what is being described

- About 30 – 40 feet long

Good Communication & Articulation - Identify what is being described

- About 30 – 40 feet long
- Usually cylindrical but may have different cross-sections

Good Communication & Articulation - Identify what is being described

- About 30 – 40 feet long
- Usually cylindrical but may have different cross-sections
- Mostly made of wood but could be another material

Good Communication & Articulation - Identify what is being described

- About 30 – 40 feet long
- Usually cylindrical but may have different cross-sections
- Mostly made of wood but could be another material
- A number of people are needed to raise it off the ground

Good Communication & Articulation - Identify what is being described

- About 30 – 40 feet long
- Usually cylindrical but may have different cross-sections
- Mostly made of wood but could be another material
- A number of people are needed to raise it off the ground
- Its carriers run with it as fast as they can

(NO MORE CLUES…….)

Communication

- What kind of statement helped you get it?
- Why didn't you get it after the first statement?
- What is special about the helpful statements?

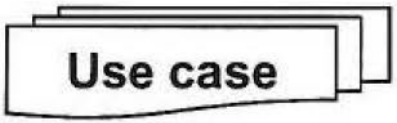
Why Use-Cases?

- Provide an easy vehicle for users to describe
  - Their needs
  - How the system is to be used
- Help us fill the gap between user needs and system functionality
- Help us manage the complexity of a system by focusing on a single aspect at a time

Why Use-Cases?

- Establish a framework to create user acceptance criteria and test cases
- Are the foundation of a successful system implementation
- Interface and data requirements can be gathered around them
- Can be used to estimate budget and schedules

Advantages of Use Cases



Clients must agree on them:
They should be able to understand them ✓

Written in natural languages and in client's terms

Developers must use them:
They should be "precise enough" and not open to misinterpretations ✓

Structured to allow clear translation to design

Testers must make sure we have implemented them:
They should be testable ✓

Easily translatable to test cases

Describe the behavioral as well as the nonbehavioral characteristics ✓

Behavioral

✗ Nonbehavioral

What Are Use Cases?

A use case is a collection of possible sequences of interactions between the system under discussion and its external actors, related to a particular goal.

—Alistair Cockburn

A use case is a sequence of transactions in a system whose task is to yield a measurable value to an individual actor of the system.

—Ivar Jacobson

You apply use cases to capture the intended behavior of the system you are developing, without having to specify how that behavior is implemented. Use cases provide a way for your developers to come to a common understanding with your system's end users and domain experts. In addition, use cases serve to help validate your architecture and to verify your system as it evolves during development.

—Grady Booch

Simplified Definition

- A **use case**
  - Is a sequence of interactions of the system with outside entities, such as users or other systems
  - Describes behaviour from a user's point of view
- Use cases describe what the system will do, *not* how it will be done
- A list of steps, typically defining interactions between a role (an "actor") and a system, to achieve a goal
- The "actor" can be a human or an external system

Example: Use Case (Scenario)

- Withdrawing funds from an ATM:

Key Parts of a Use Case

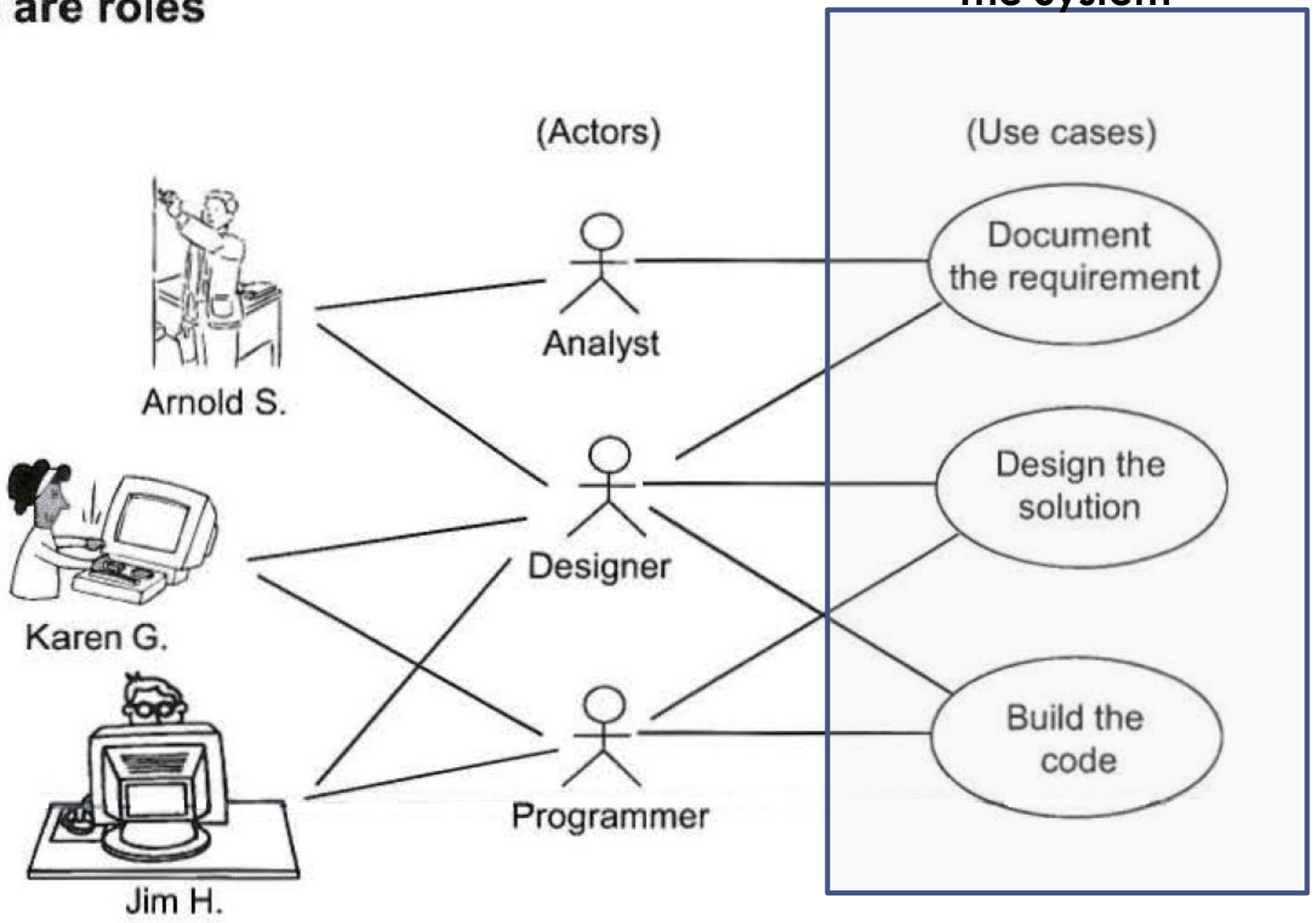
- Actor
- Goal
- Main flow

Actors

- **What is an actor?**
  - An entity that is external to the system under construction and that interacts with it
    - The customer
  - A role played by a user or an external entity that interacts with the system in order to obtain some value from that interaction
  - Actors are
    - People
    - Organizations
    - Other software or hardware systems
    - Time-events that happen at, or after, some time
  - Actors are nondeterministic
    - Should not assume actors always correctly stimulate the system
- **All of the actors of all the use cases = all the external entities that interact with the system**

Actors + Use Cases

- **Actors are roles**



Goals

- **The goal**
  - An objective to achieve, usually the name of the use case
    - "Withdraw money"
- **The goal for the use case is what has to be achieved**
  - It is expressed by the use-case name
- **Goals help identify and sequence the events**
- **Examples:**
  - Withdraw money from my account
  - Place an order

Main Flow

- The main flow is the expected or most frequently used path through the use case
  - It describes the entire interaction between the actor(s) and the system to accomplish the goal

Example: Use Case (Scenario)

- **Withdrawing funds from an ATM (MAIN FLOW):**

Withdraw funds

- The customer inserts their ATM card and the system asks the customer to select a language . The customer selects their preferred language and the system requests a PIN. The customer enters their "PIN Number" and the system asks the customer to select a transaction. The customer selects "withdraw money". The system asks the customer to pick an amount, the customer picks an amount and the system dispenses the money. The system asks if the customer wants another transaction and the customer selects "no". The system dispenses the ATM card, prints the.. receipt, records the transaction and closes the door.

Use Cases & Use Case Diagrams

- **UML (Unified Modelling Language)** – the use of various diagrams to a describe a system at different levels – analysis, design, implementation.

- A **USE CASE (textual description)** is a list of steps, typically defining interactions between a role (an actor) and a system to achieve a goal. The actor can be a human or another (external) system.

- Use Cases, Actors and Relationships are compiled together in a **Use Case Diagram**.

Example Project – Create a Knowledge Base

- Imagine that we've been asked to write a simple **knowledge base** for an organisation
  - A centralised repository for information
  - A database used for managing and sharing information
- We've talked to the customer, written a few casual use cases, brainstormed some actors of this new system that we plan to build
- **Use Case Diagram** – an overview of several use cases and multiple actors, and the related interactions between them

Use Case Titles (in no particular order)

Search Articles

View Article

Manage Users

Create Article

View Analytics

Use Case Titles (in no particular order)

Search Articles

View Article

Manage Users

Create Article

View Analytics

Analysis of data for patterns and trends as an aid to decision making

Our Actors are all Role Based

Visitor

Search Articles

View Article

Contributor

Manage Users

Create Article

Administrator

View Analytics

Stick Figures Represent the Primary Actors

Visitor

Search Articles

Contributor

View Article

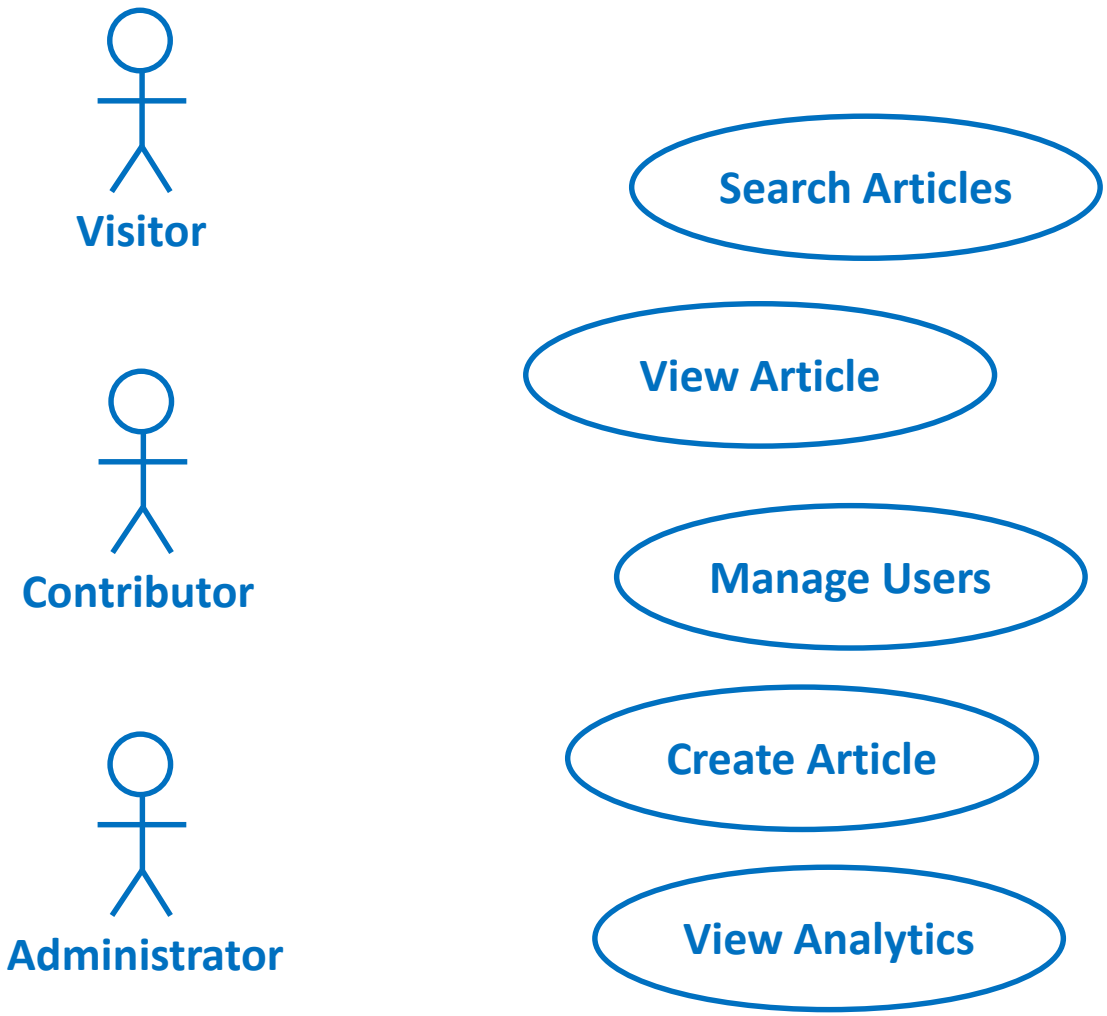
Manage Users

Administrator

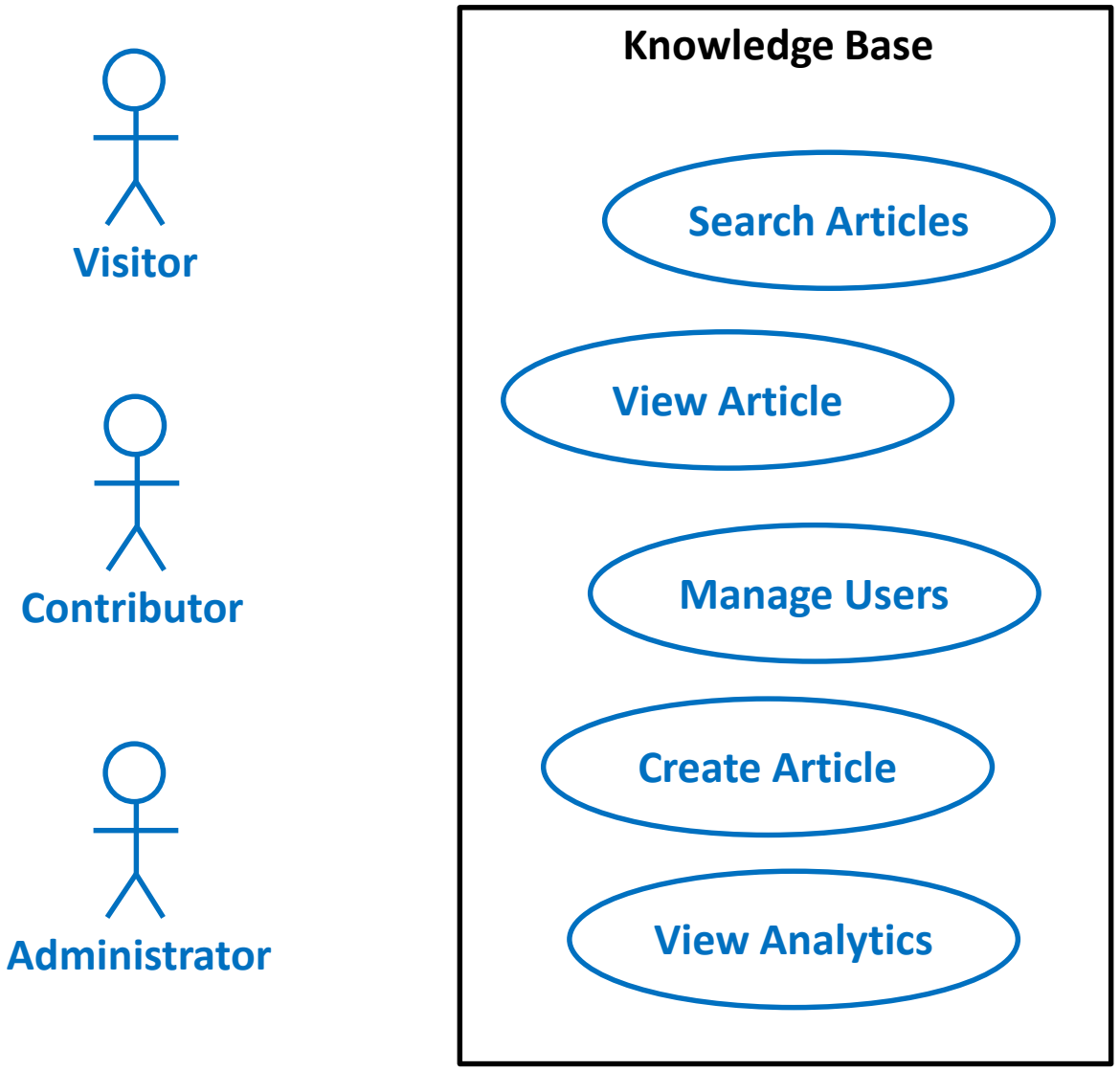
Create Article

View Analytics

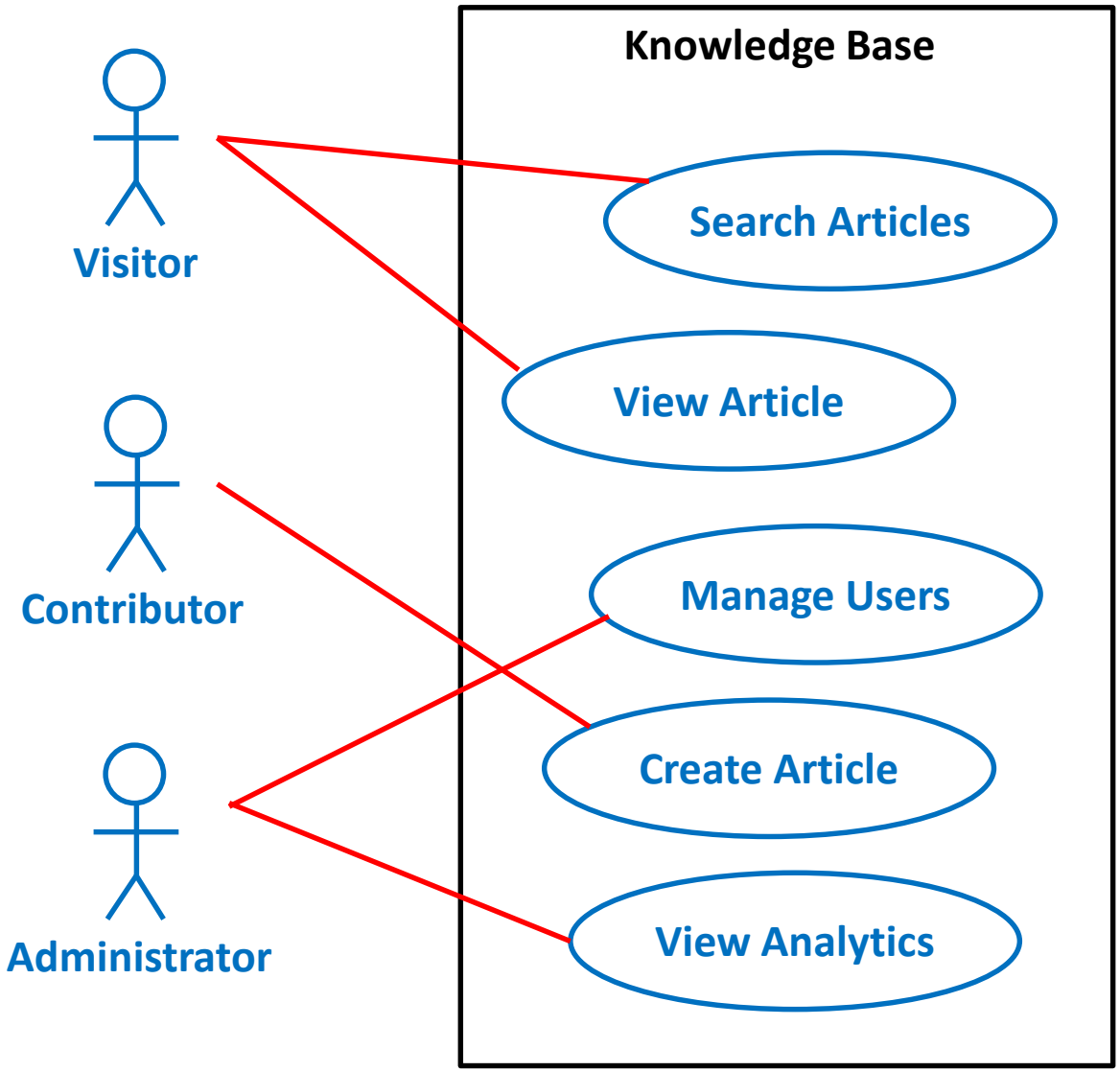
Ellipses or Ovals around Use Case Titles



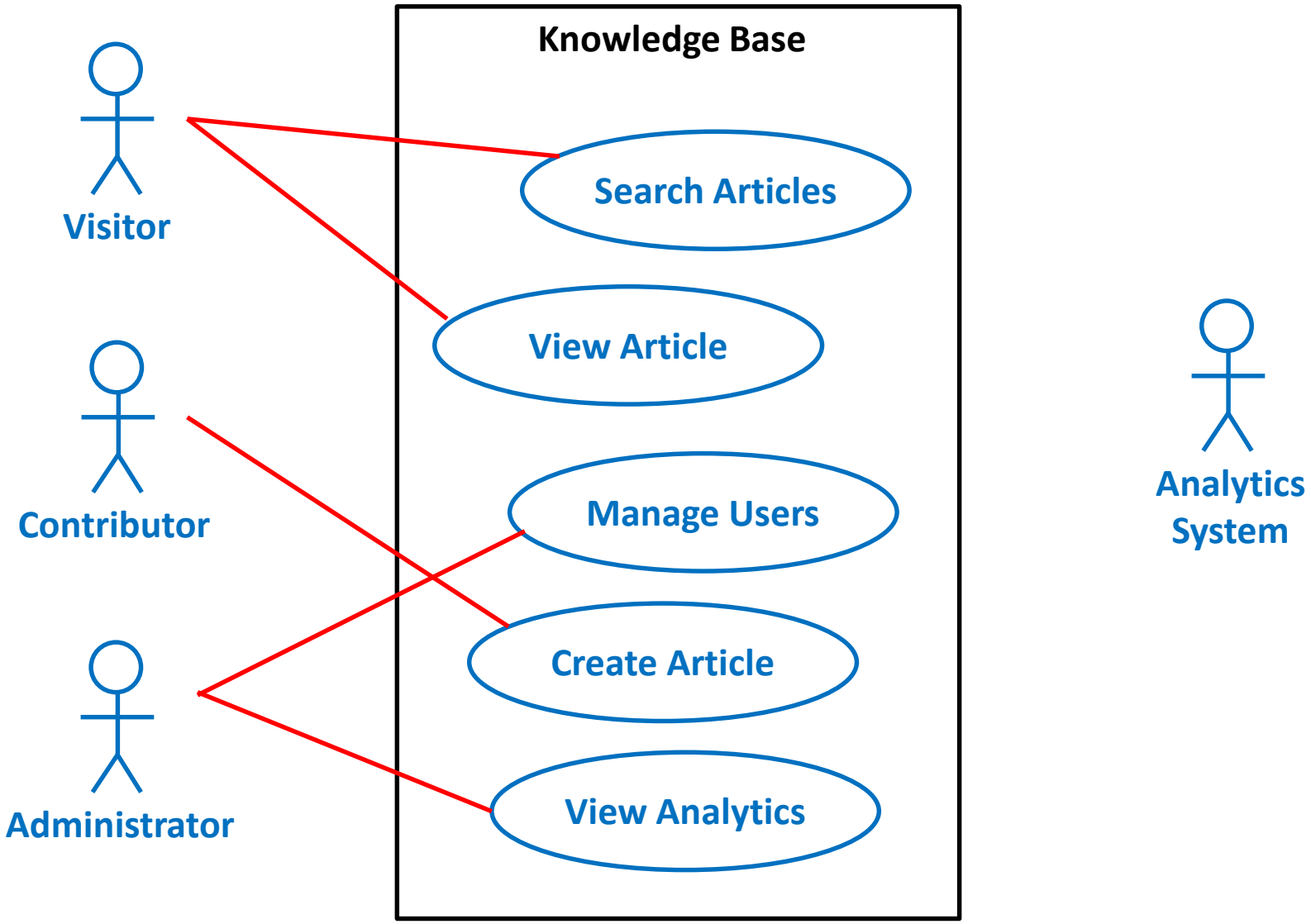
Box to Represent Boundaries of System



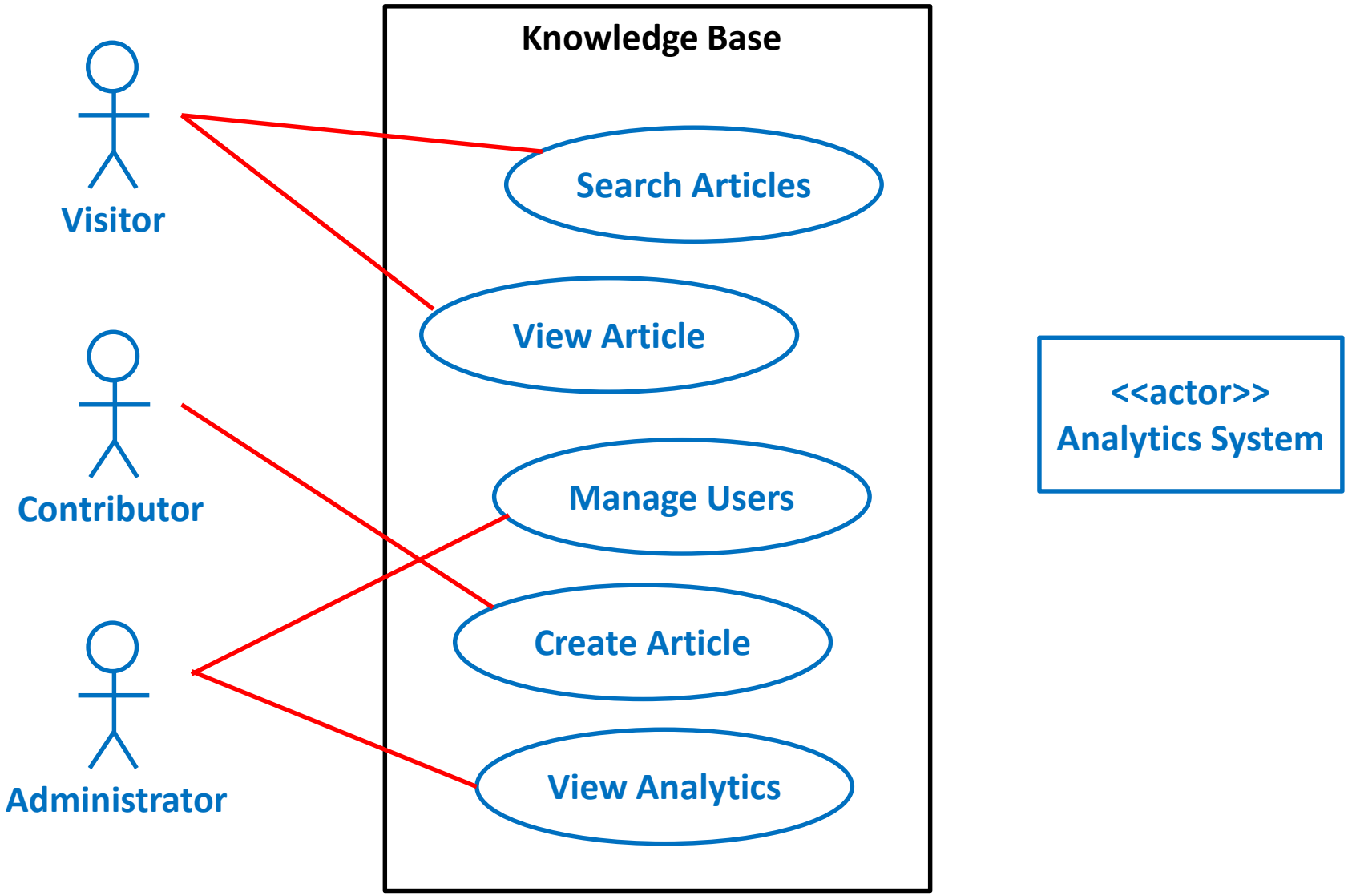
Lines of Interaction (Relationships)



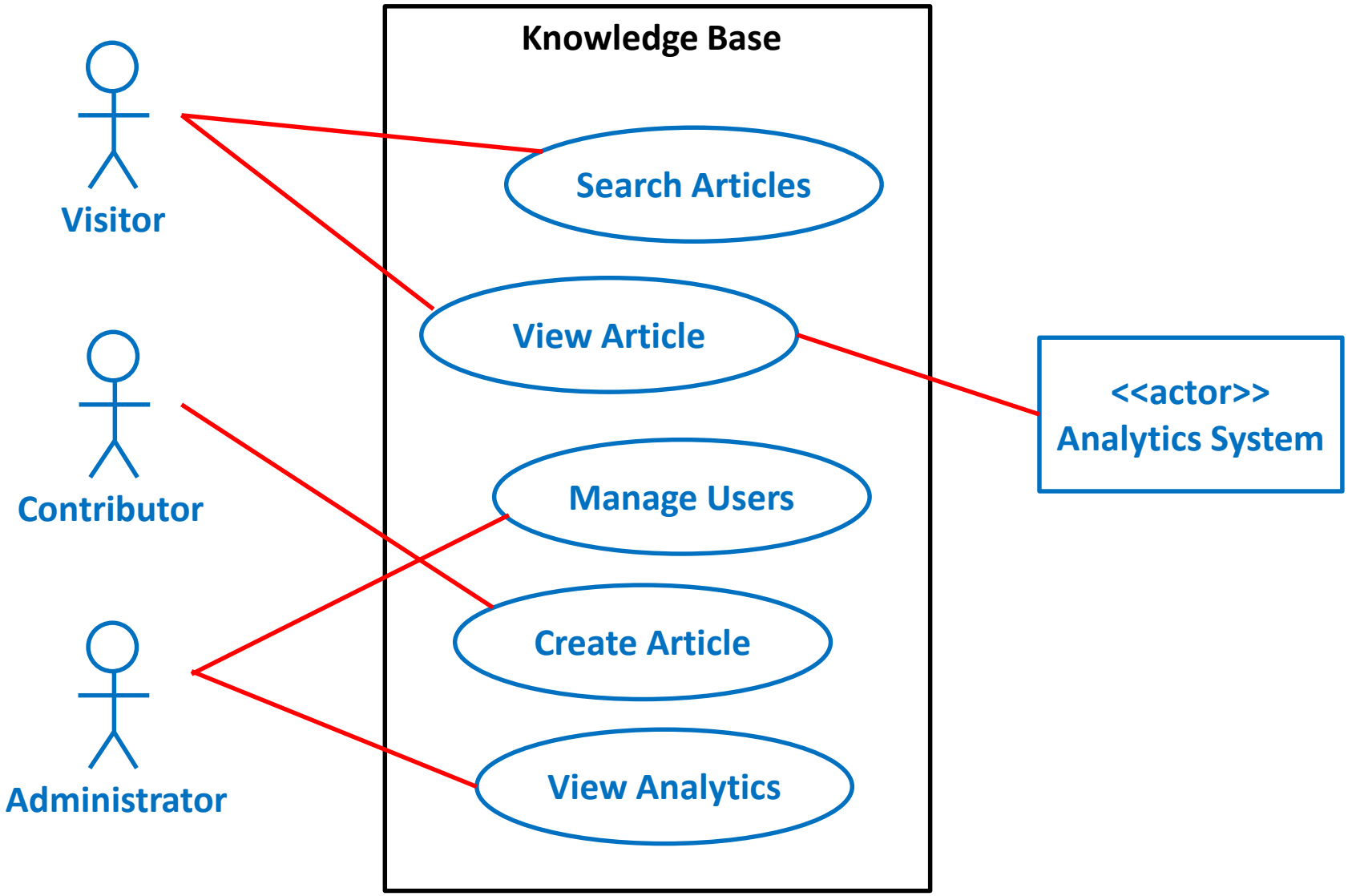
Separate Computer System Stores Analytics Data



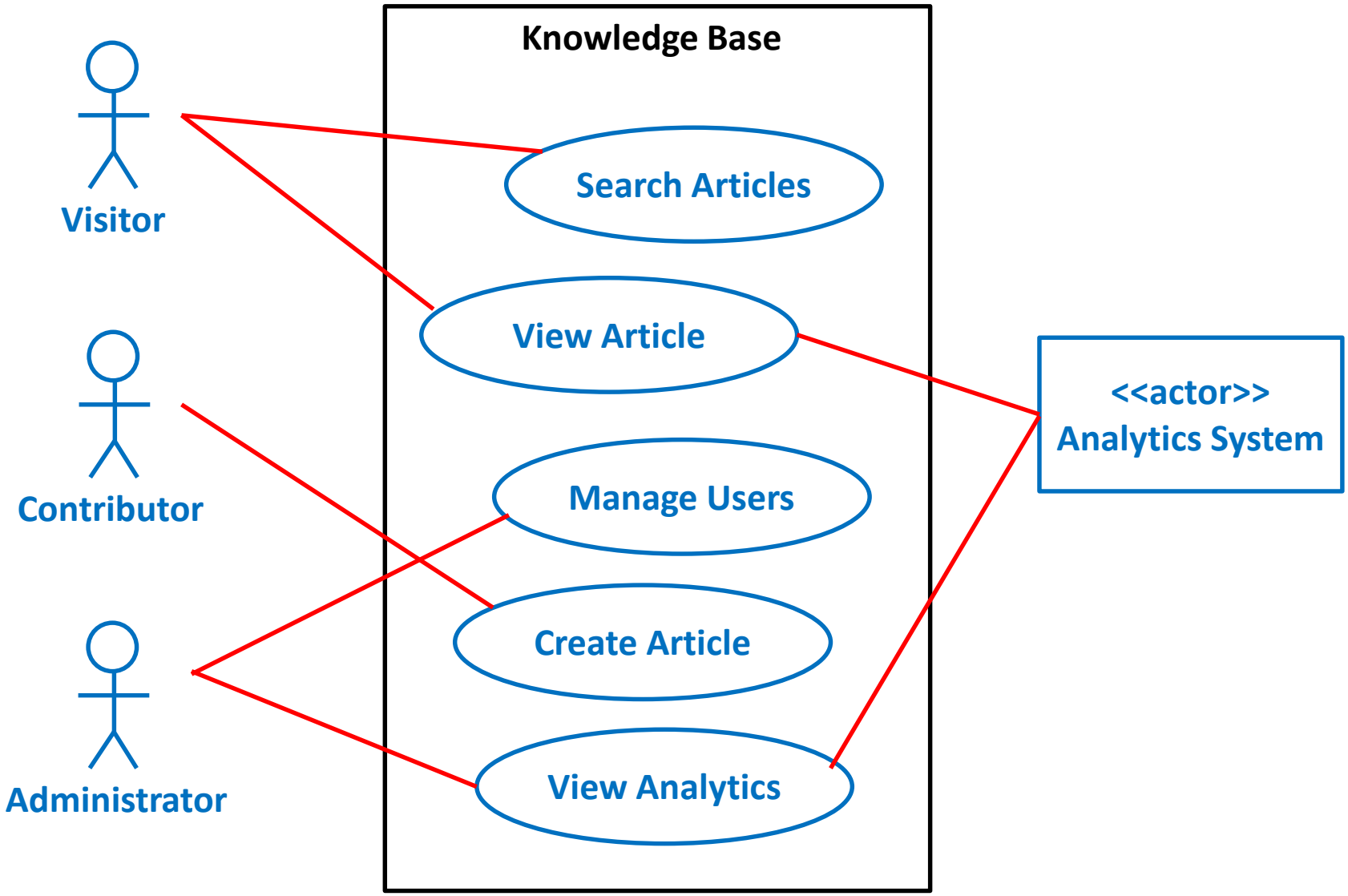
Box to Represent a Non-Human Actor



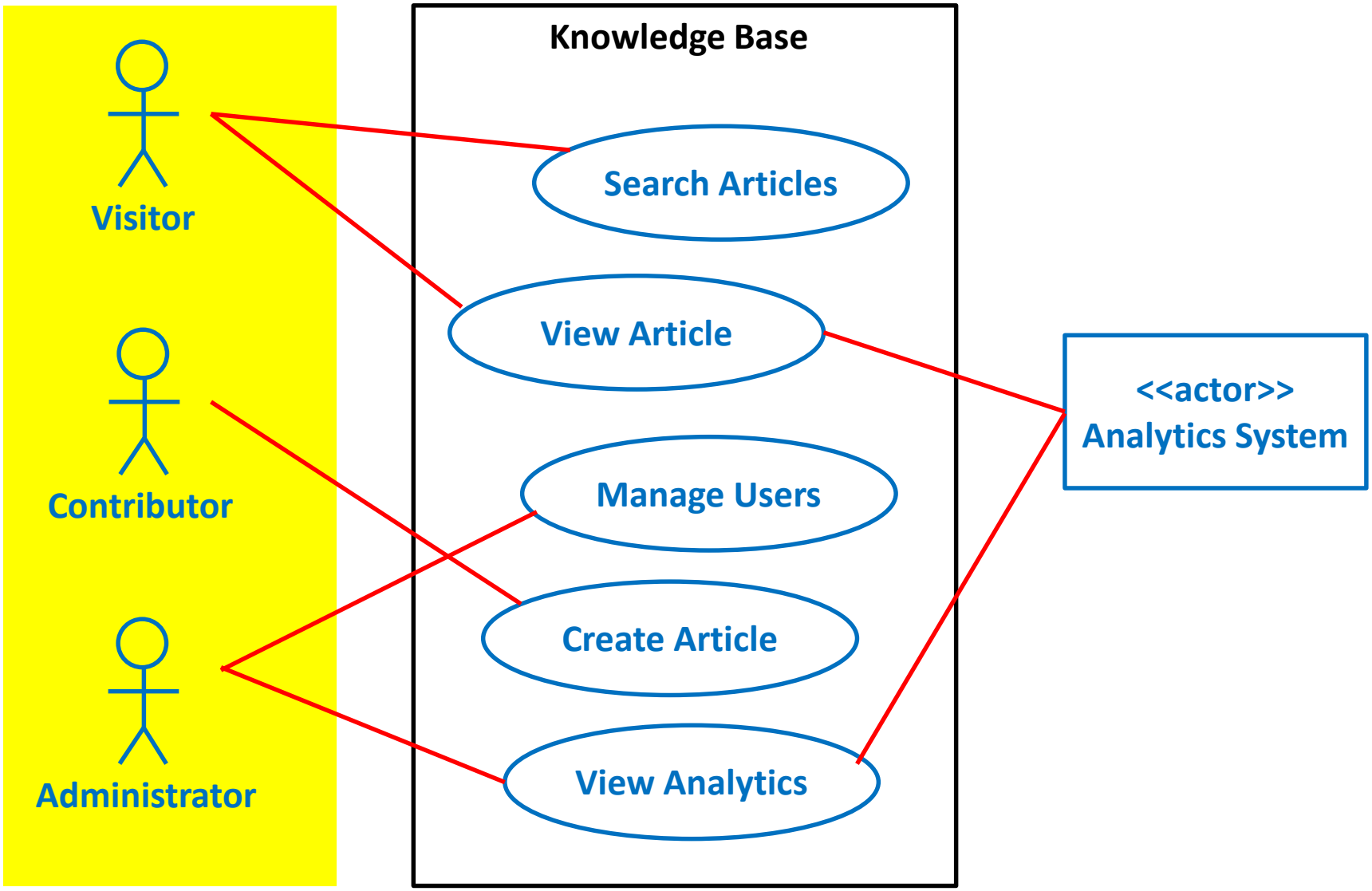
View Article: Save Data to Analytics System



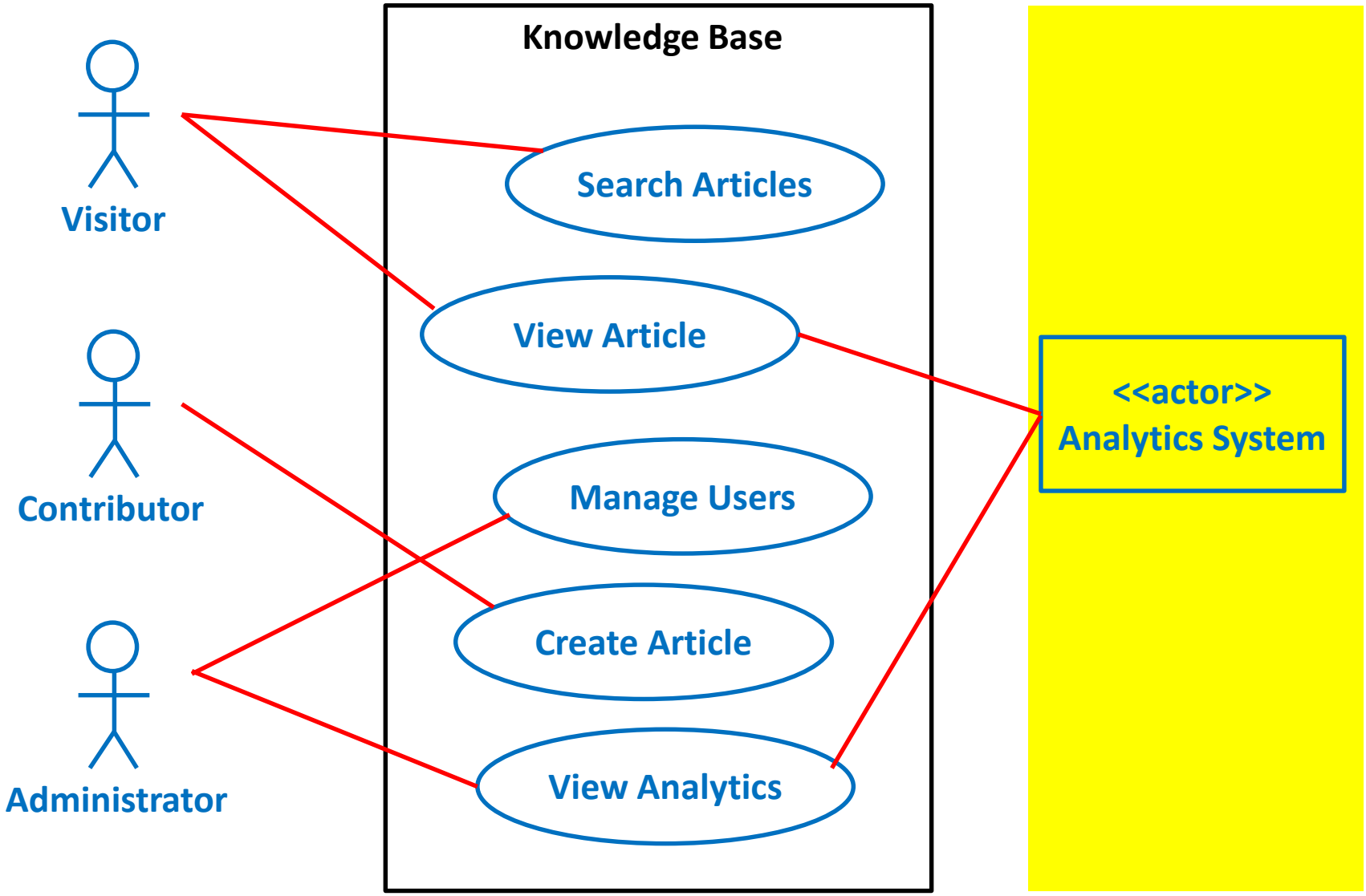
View Analytics: Administrator Reads Details



Primary Actors (LHS): Initiate Use Cases



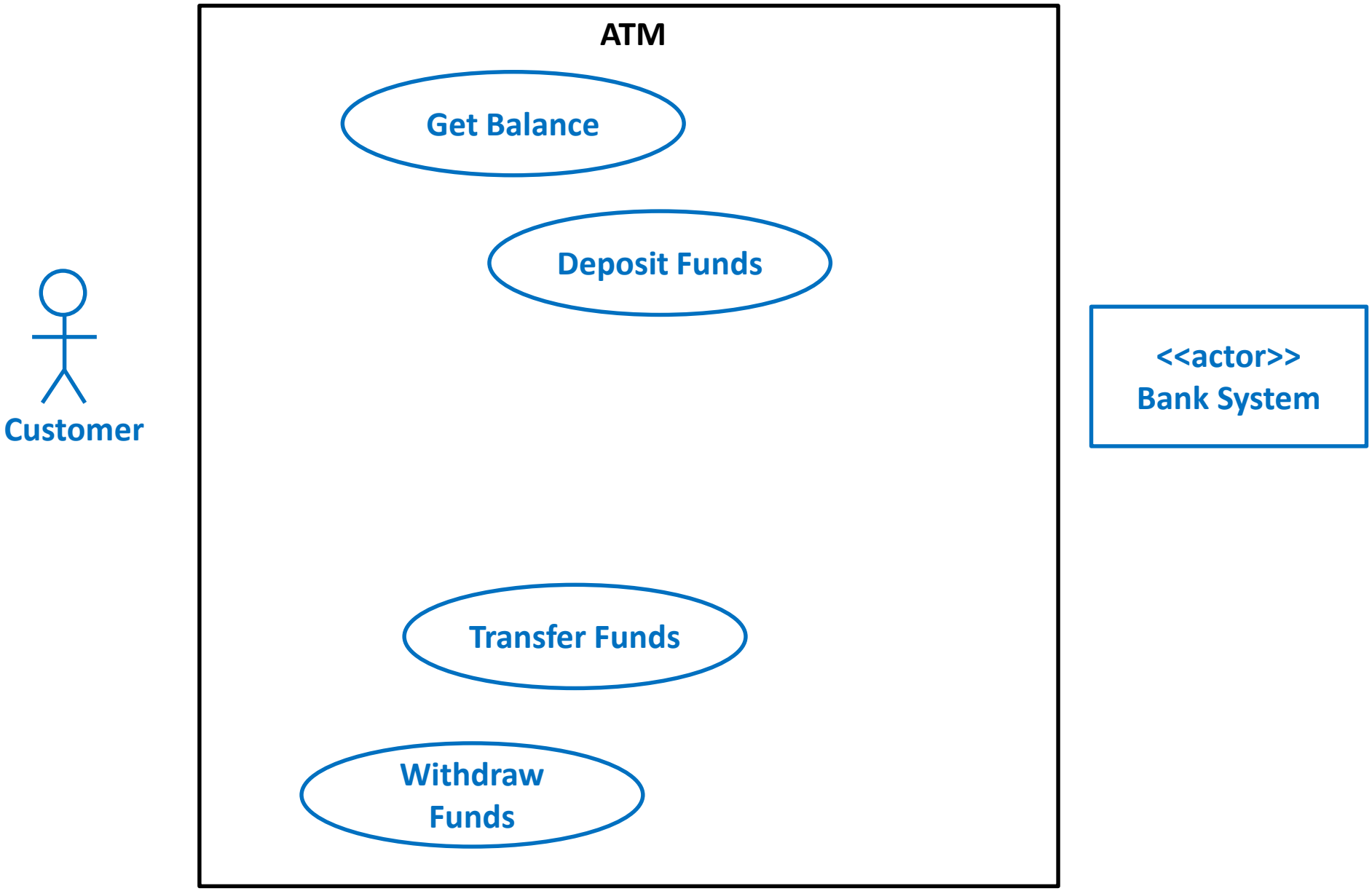
Secondary Actors (RHS): More Reactive Role



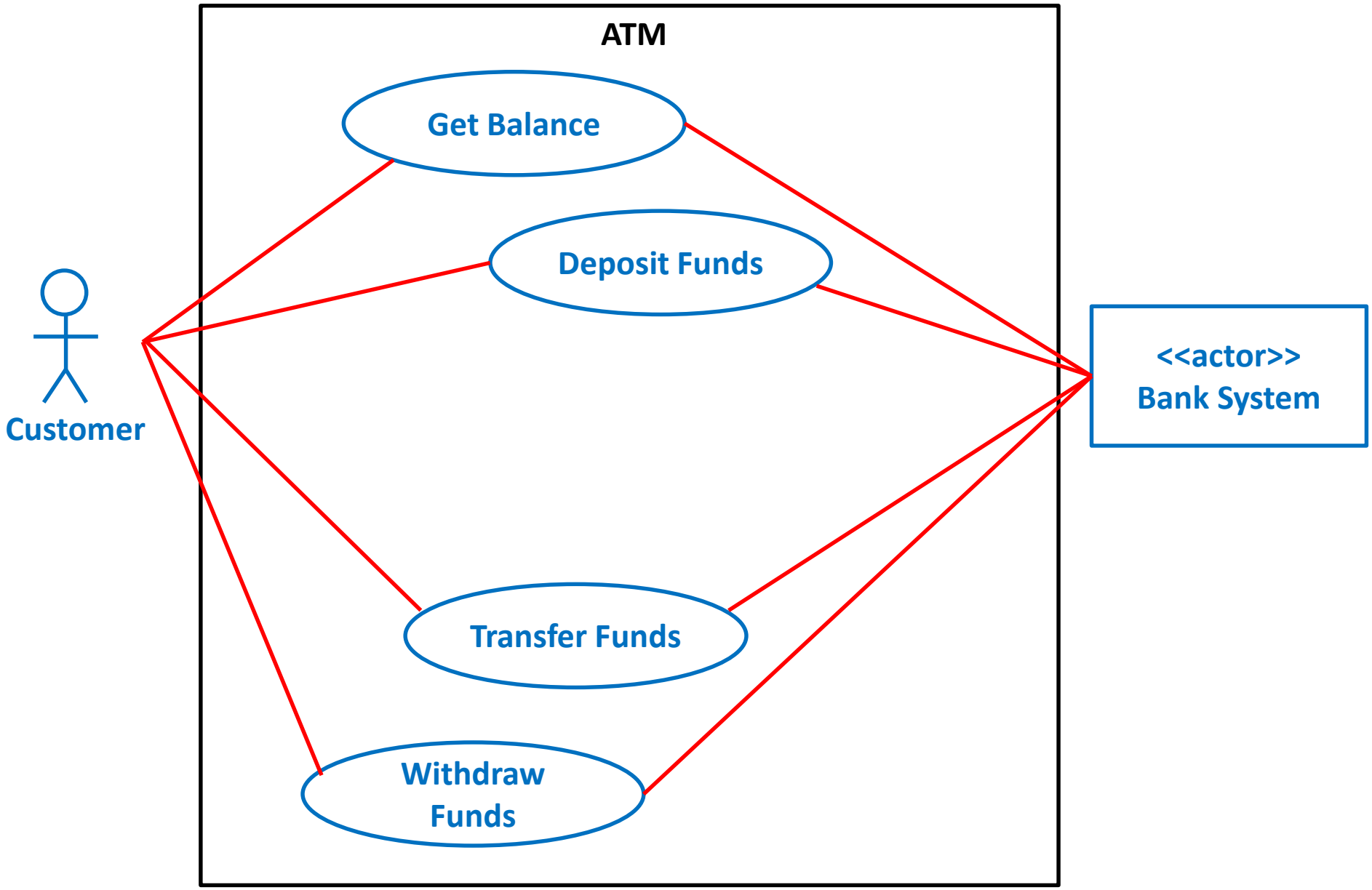
Use Case Diagram

- Not necessarily any sequence or order to the use cases
- Simply an overview of multiple use cases and multiple actors at the same time, without the details of the written use cases
- Useful as a communication tool
  - For business and technical audiences
  - Can help in working out if something is missing from the picture

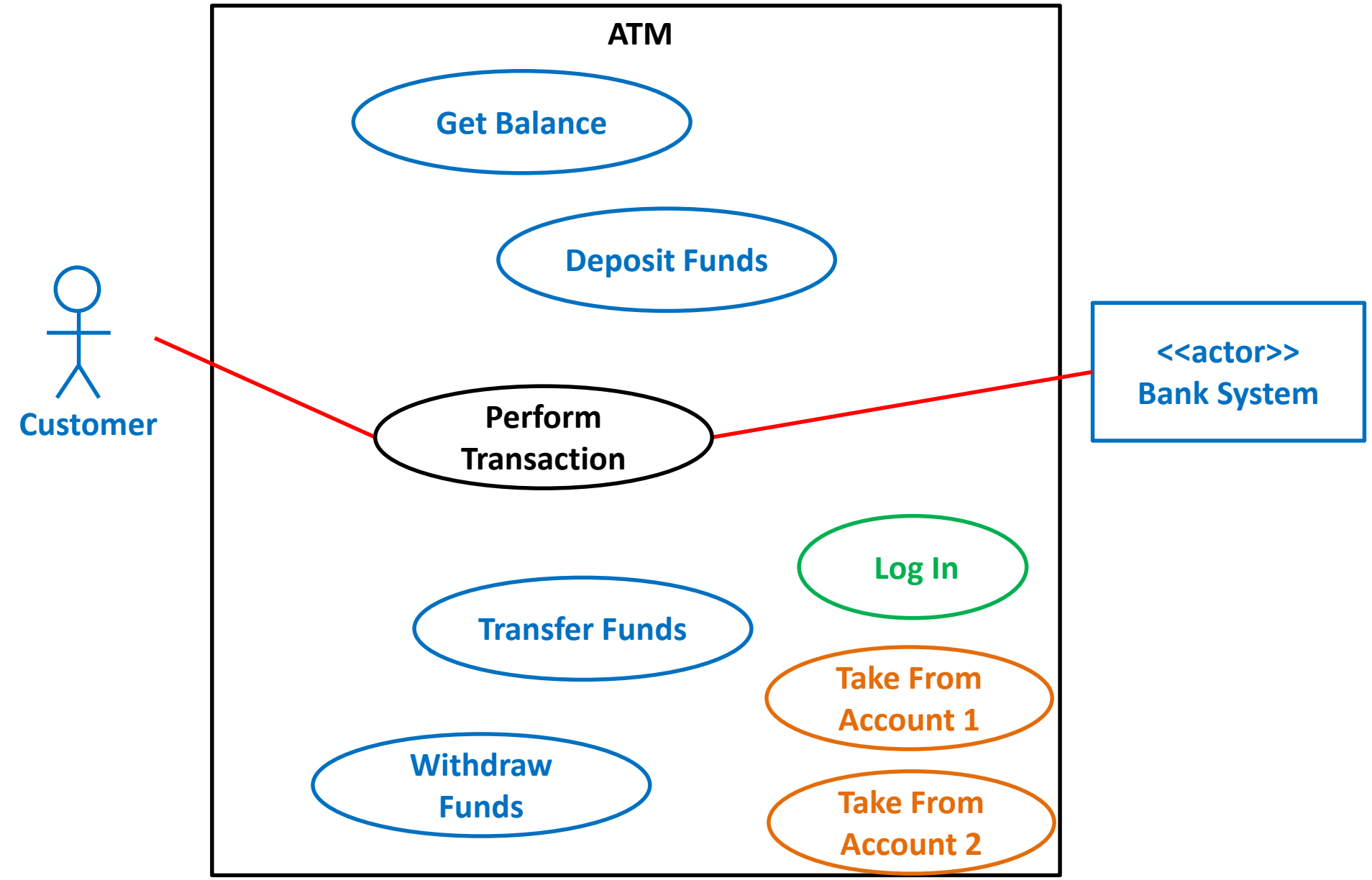
Automated Teller Machine (ATM) - 1



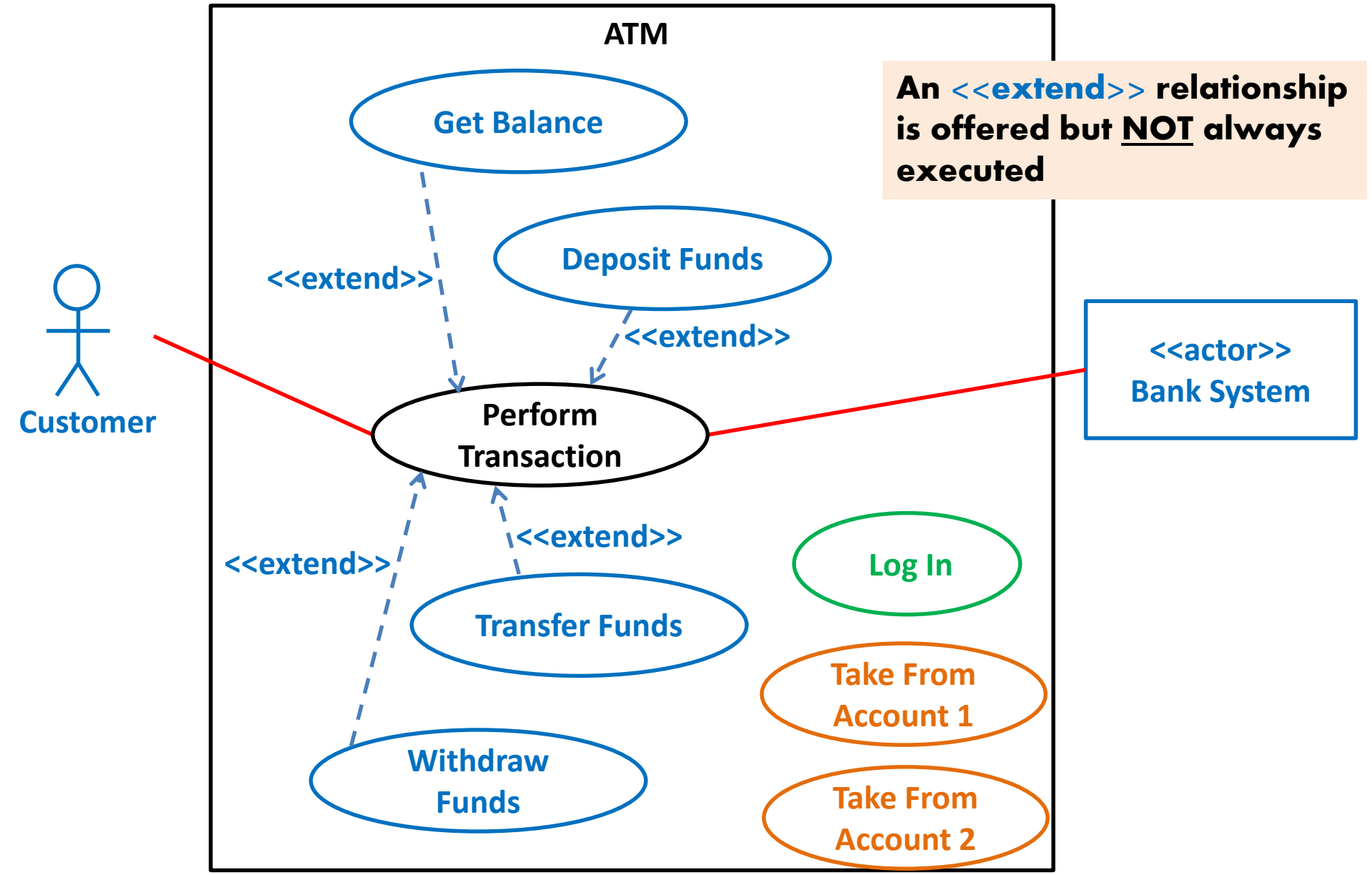
Automated Teller Machine (ATM) - 1



Automated Teller Machine (ATM) - 2

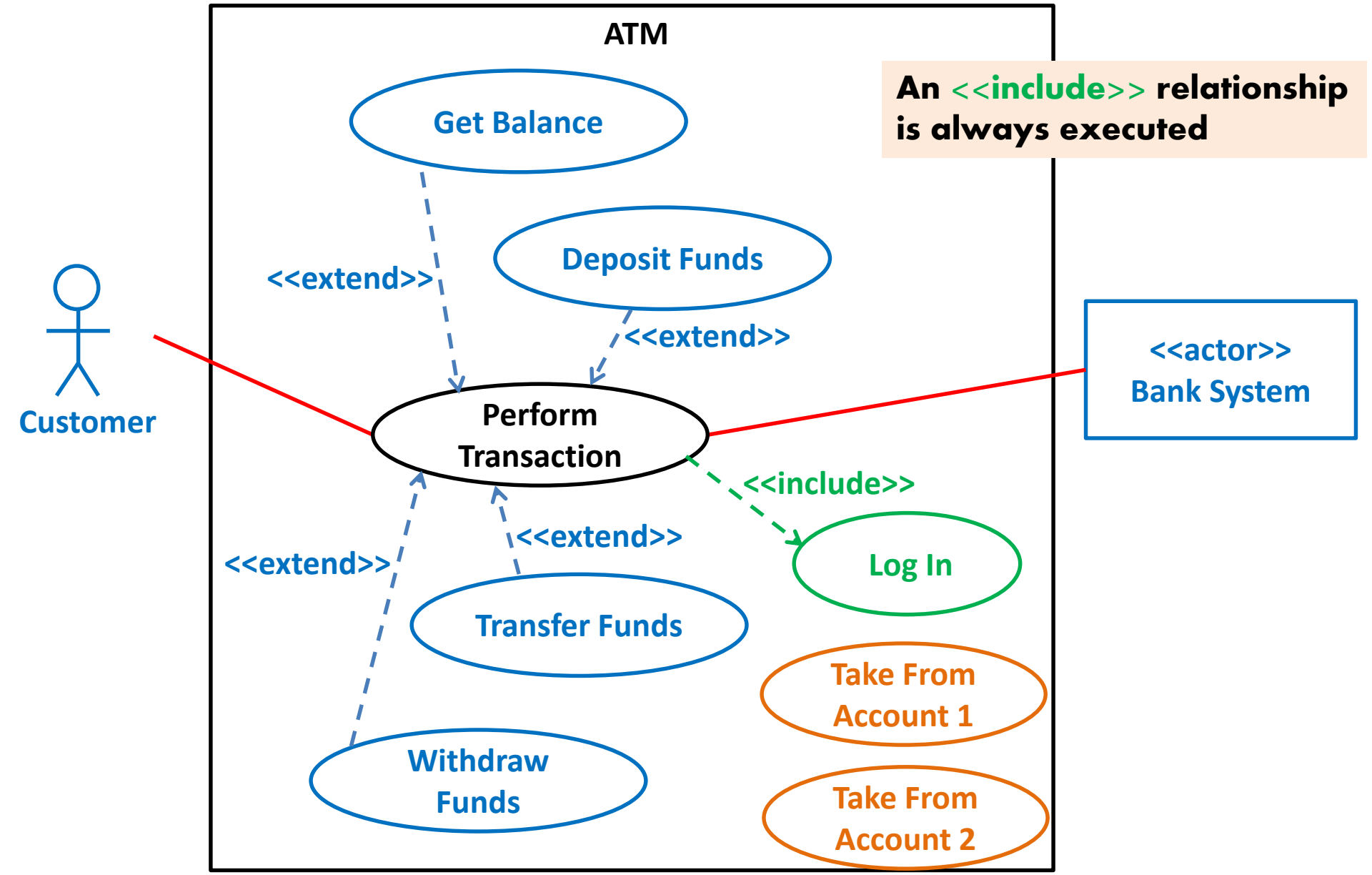


Automated Teller Machine (ATM) - 2

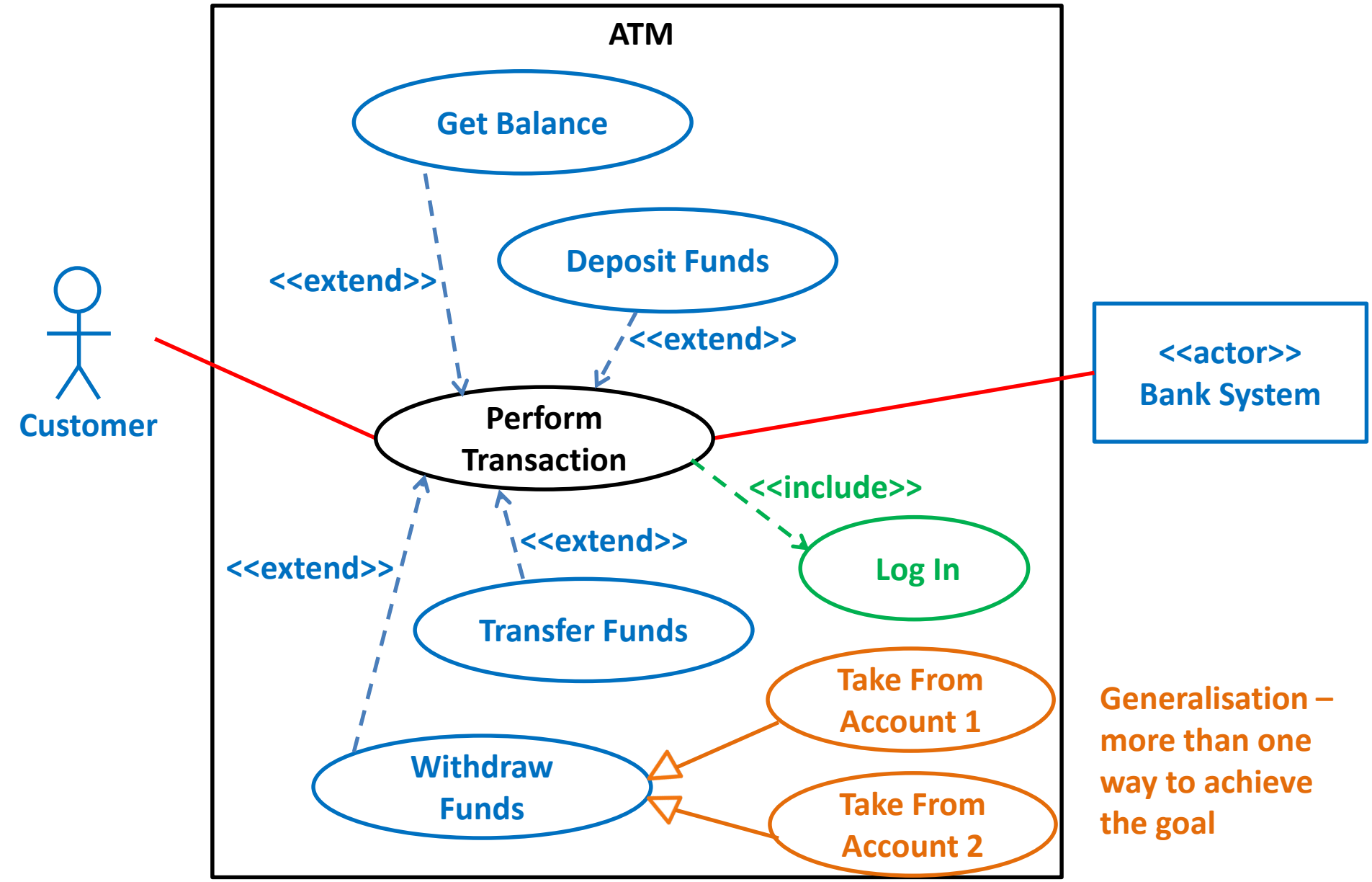


An <<extend>> relationship is offered but NOT always executed

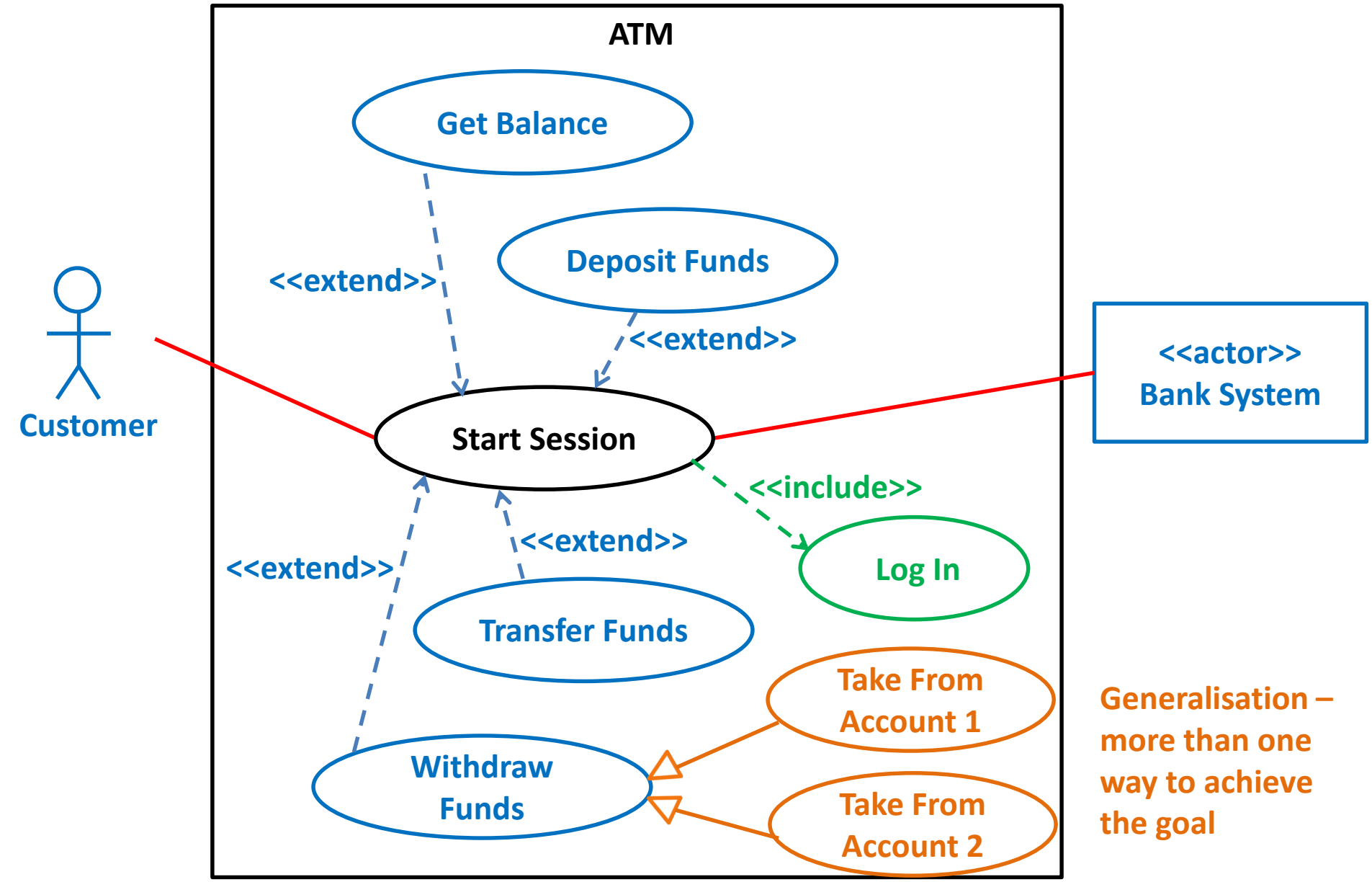
Automated Teller Machine (ATM) - 2



Automated Teller Machine (ATM) - 2



Richard Jones Alternative



Generalisation – more than one way to achieve the goal

Example: <<include>> Relationship

Withdraw Funds

1. Customer enters card
2. System asks customer to choose a language and enter a PIN
3. Customer selects a language and enters a PIN
4. System validates the PIN
5. System asks for a transaction type
6. Customer selects "Withdraw Funds"
7. System asks for account type
8. Customer selects account
9. System asks for amount
10. Customer selects amount
11. System validates amount
12. Machine dispenses amount
13. System records transaction, dispenses card and receipt, and closes door

Transfer Funds

1. Customer enters card
2. System asks customer to choose a language and enter a PIN
3. Customer selects a language and enters a PIN
4. System validates the PIN
5. System asks for a transaction type
6. Customer selects "Transfer Funds"
7. System asks customer to choose "accounts from" and "accounts to"
8. Customer selects "accounts from" and "accounts to"
9. System asks for amount
10. Customer selects amount
11. System validates amount
12. System transfers funds
13. System records transaction, dispenses card and receipt, and closes door

Example: <<include>> Relationship

Withdraw Funds

1. Customer enters card
2. System asks customer to choose a language and enter a PIN
3. Customer selects a language and enters a PIN
4. System validates the PIN

*

5. System asks for a transaction type
6. Customer selects "Withdraw Funds"
7. System asks for account type
8. Customer selects account
9. System asks for amount
10. Customer selects amount
11. System validates amount
12. Machine dispenses amount
13. System records transaction, dispenses card and receipt, and closes door

Transfer Funds

1. Customer enters card
2. System asks customer to choose a language and enter a PIN
3. Customer selects a language and enters a PIN
4. System validates the PIN

*

5. System asks for a transaction type
6. Customer selects "Transfer Funds"
7. System asks customer to choose "accounts from" and "accounts to"
8. Customer selects "accounts from" and "accounts to"
9. System asks for amount
10. Customer selects amount
11. System validates amount
12. System transfers funds
13. System records transaction, dispenses card and receipt, and closes door

Example: <<include>> Relationship

Withdraw Funds

1. Customer enters card
2. System asks customer to choose a language and enter a PIN
3. Customer selects a language and enters a PIN
4. System validates the PIN

*

5. System asks for a transaction type
6. Customer selects "Withdraw Funds"
7. System asks for account type
8. Customer selects account
9. System asks for amount
10. Customer selects amount
11. System validates amount
12. Machine dispenses amount
13. System records transaction, dispenses card and receipt, and closes door

Transfer Funds

1. Customer enters card
2. System asks customer to choose a language and enter a PIN
3. Customer selects a language and enters a PIN
4. System validates the PIN

*

5. System asks for a transaction type
6. Customer selects "Transfer Funds"
7. System asks customer to choose "accounts from" and "accounts to"
8. Customer selects "accounts from" and "accounts to"
9. System asks for amount
10. Customer selects amount
11. System validates amount
12. System transfers funds
13. System records transaction, dispenses card and receipt, and closes door

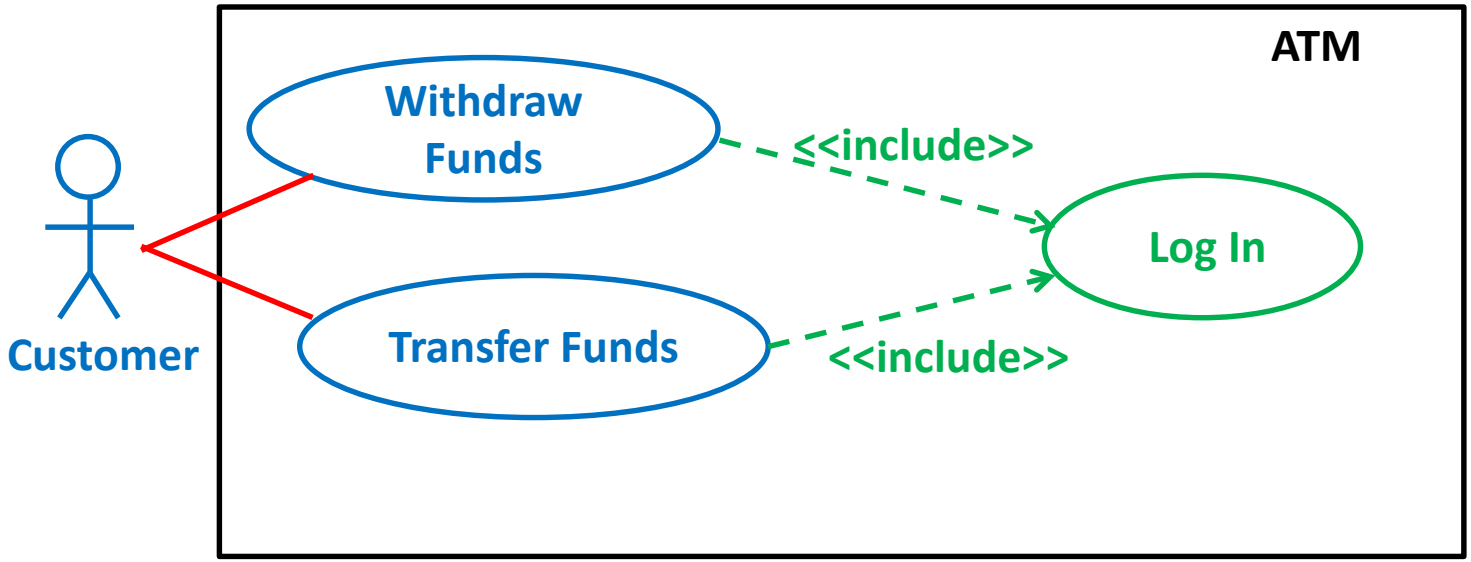
* **Log In** Use Case

Example: <<include>> Relationship

Withdraw Funds
1. Log in use case
2. System asks for a transaction type
3. Customer selects "Withdraw Funds"
4. …

Transfer Funds
1. Log in use case
2. System asks for a transaction type
3. Customer selects "Transfer Funds"
4. …

Log in *
1. Customer enters card
2. System asks customer to choose a language and enter a PIN
3. Customer selects a language and enters a PIN
4. System validates the PIN



Example: <<extend>> Relationship

Perform ATM Transaction

1. **Log In** use case
2. System asks for a transaction type
3. Customer selects transaction type
   - 3.1 **Withdraw Funds** use case
   - 3.2 **Transfer Funds** use case
   - 3.3 **Get Balance** use case
4. System records transaction, dispenses card and receipt, and closes door

Transfer Funds

Extension point: 3

Conditions: Customer selects "Transfer Funds"

1. System asks customer to choose "from account" and "to account"
2. Customer selects "from account" and "to account"
3. System asks for amount
4. Customer selects amount
5. System validates amount
6. System transfers funds

Remerge at: 2

Withdraw Funds

Extension point: 3

Conditions: Customer selects "Withdraw Funds"

1. System asks for account type
2. Customer selects account
3. System asks for amount
4. Customer selects amount
5. System validates amount
6. Machine dispenses amount

Remerge at: 2

Get Balance

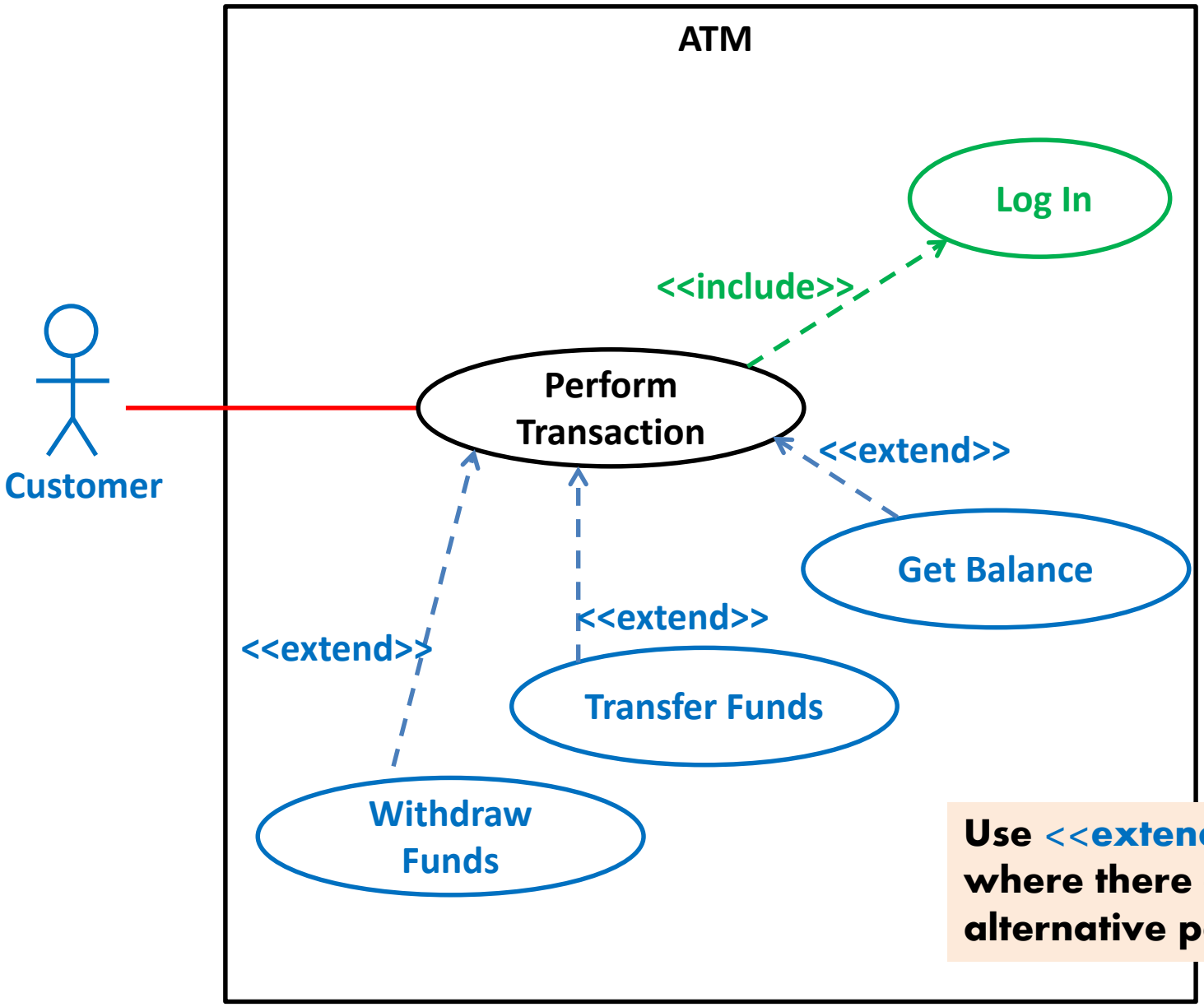
Extension point: 3

Conditions: Customer selects "Get Balance"

1. System asks customer to choose account
2. Customer selects account
3. System provides balance

Remerge at: 2

Example: <<extend>> Relationship



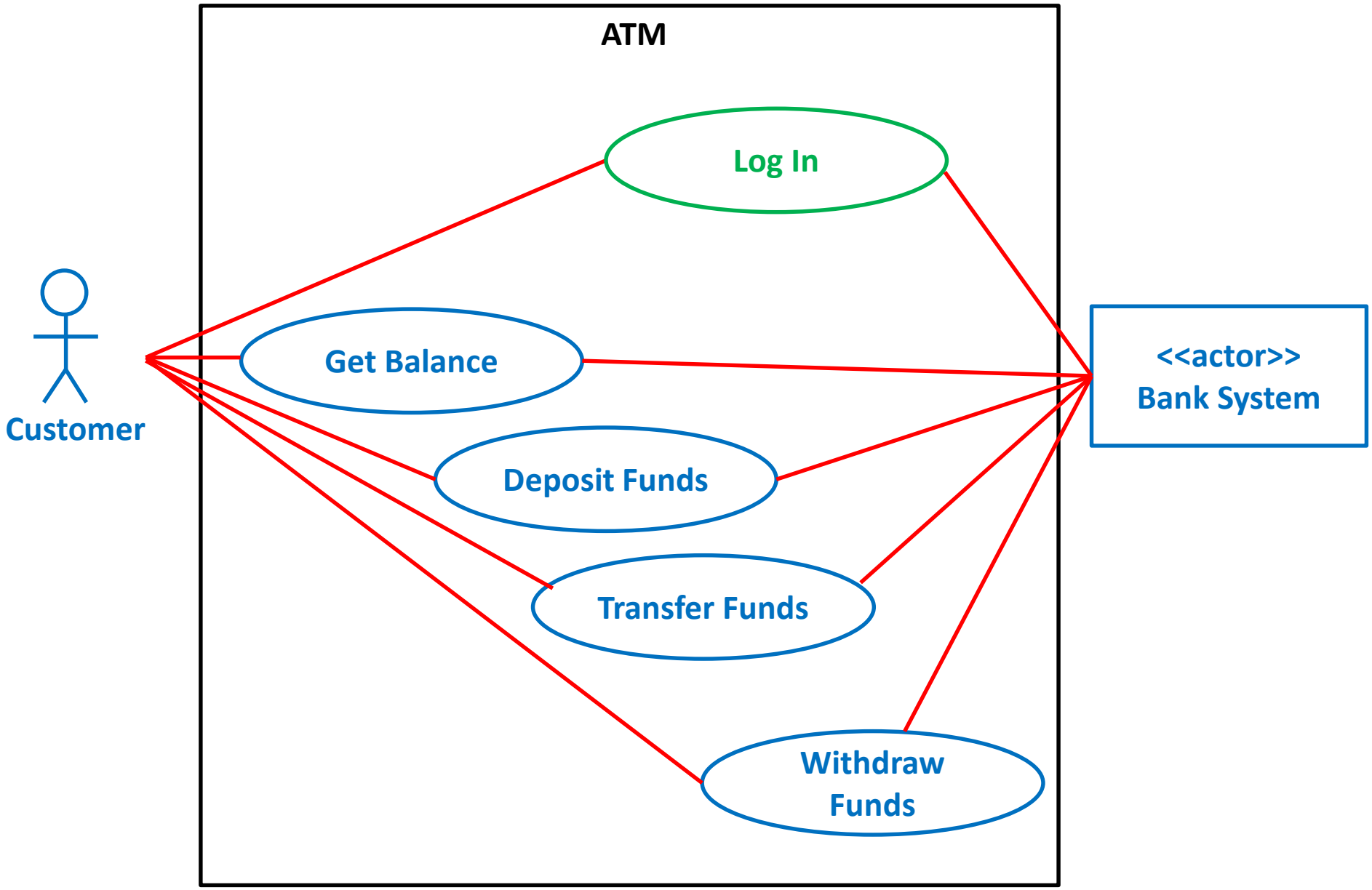
Use <<extend>> relationships where there are meaningful alternative paths

Some Words of Advice

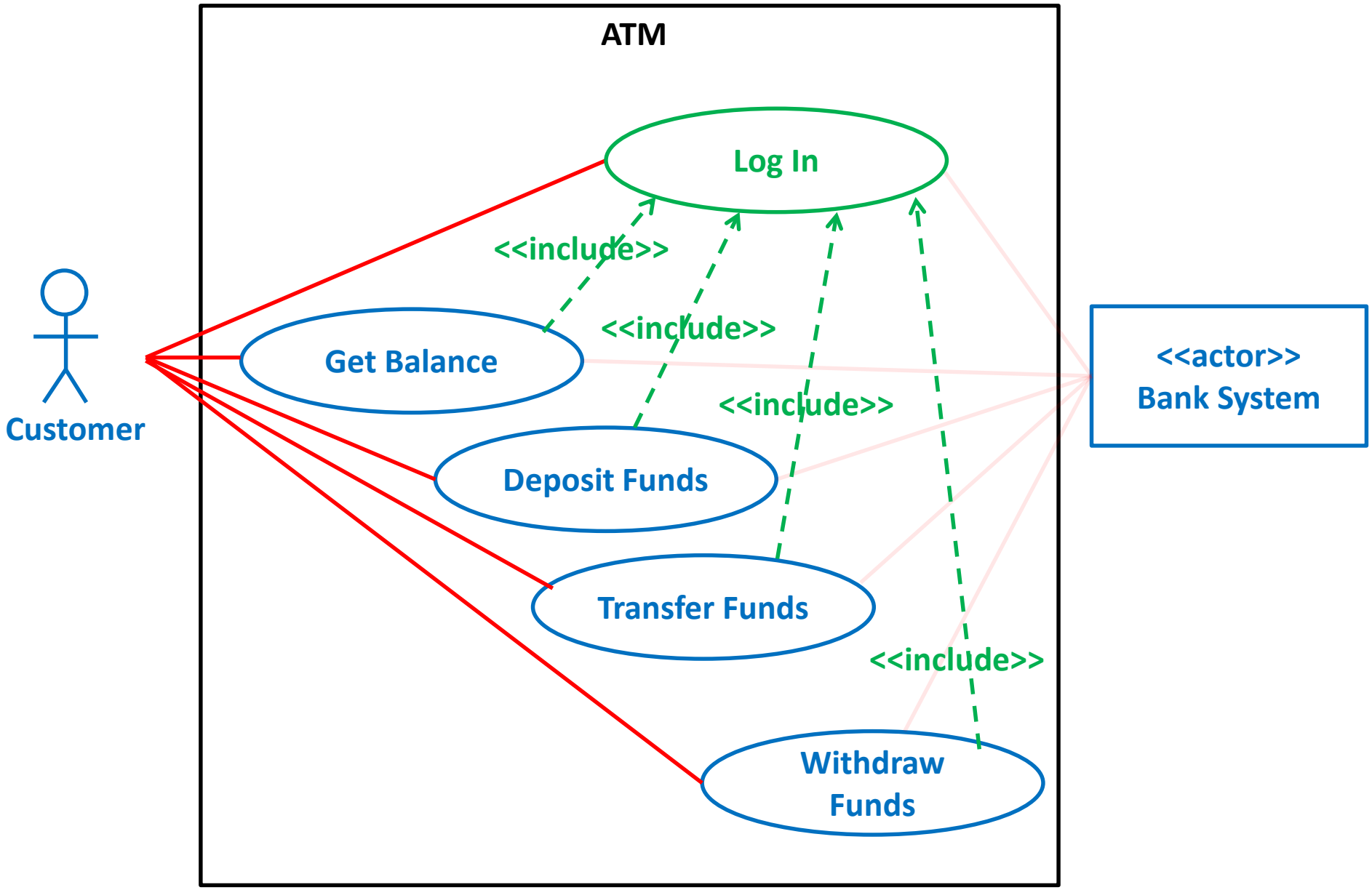
- There is not always a model answer
- Different analysts may have different perspectives and different ways of adapting the diagrams to describe the system and the various interactions
- As long as it makes sense to the stakeholders (but must be tested and validated)

- For example….

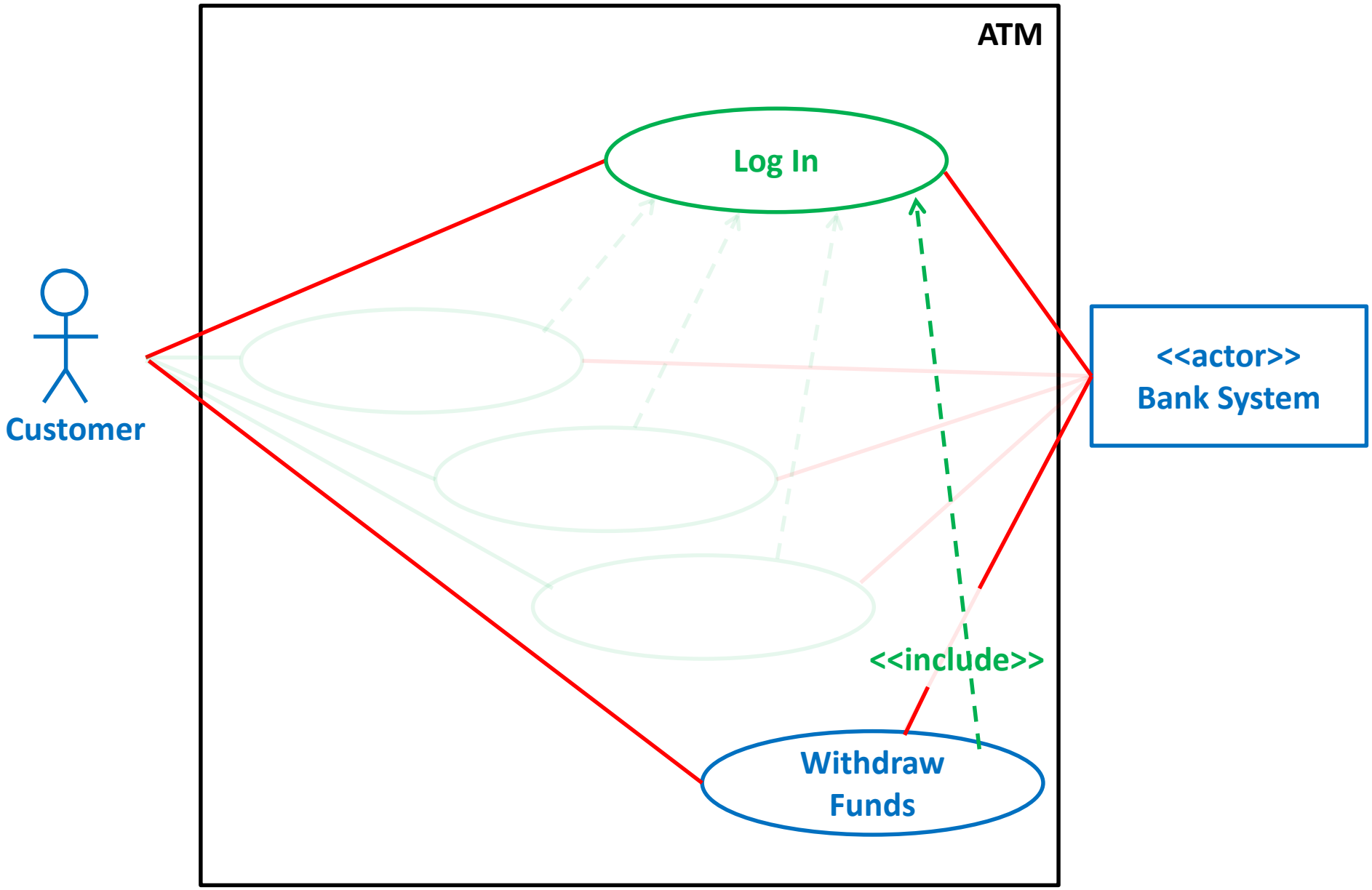
Automated Teller Machine (ATM) - 3



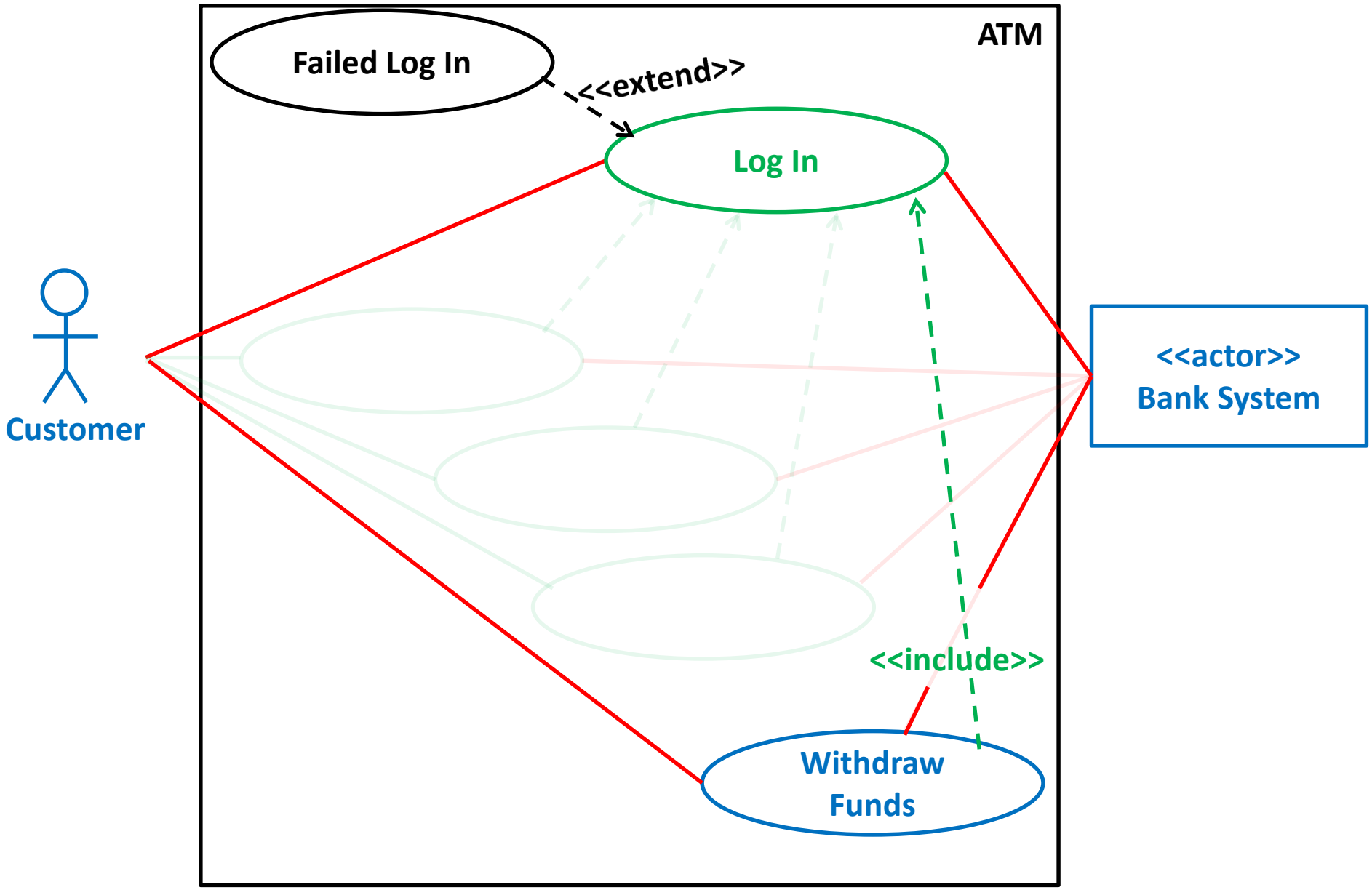
Automated Teller Machine (ATM) - 3



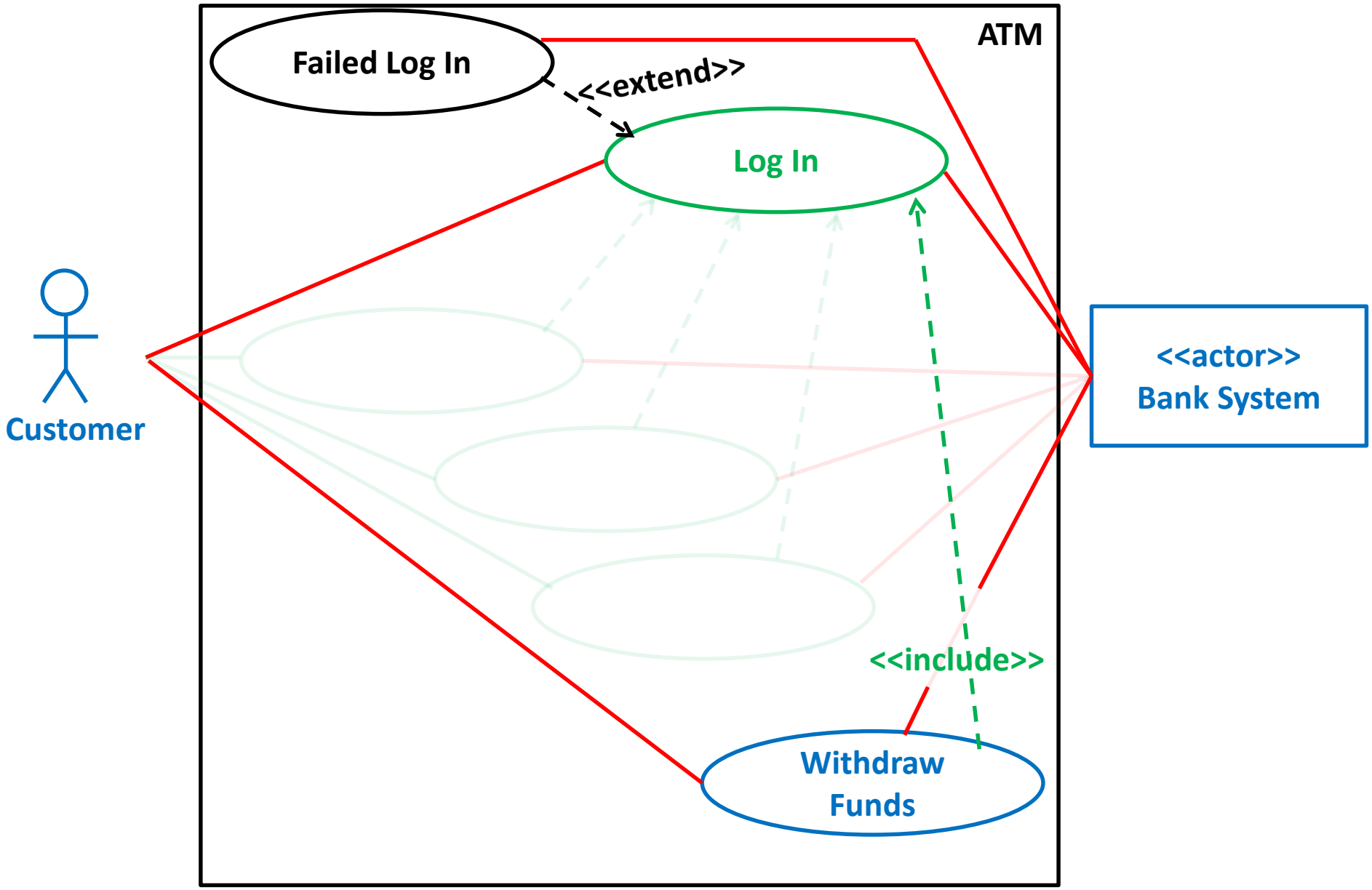
Automated Teller Machine (ATM) - 3



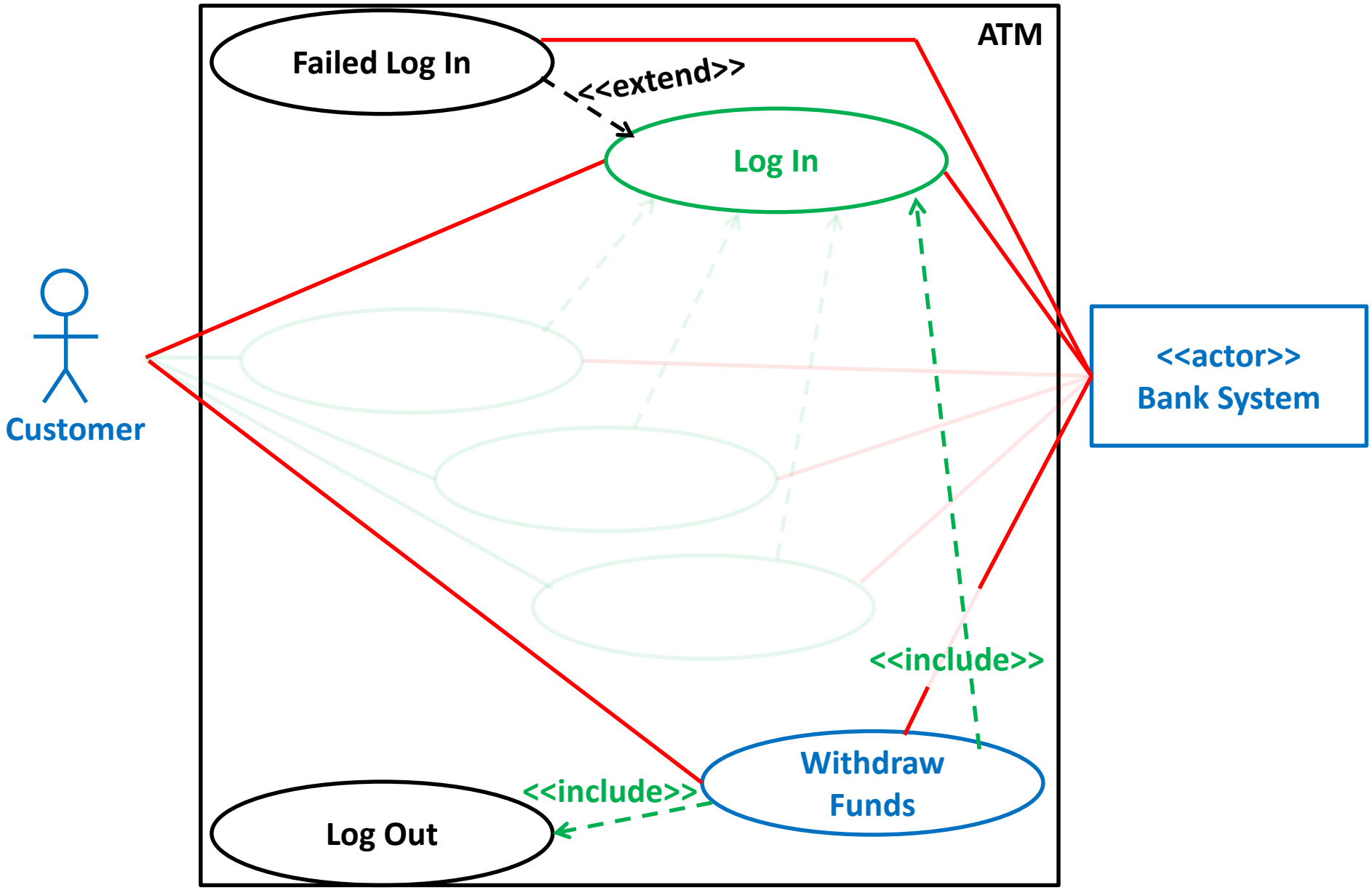
Automated Teller Machine (ATM) - 3



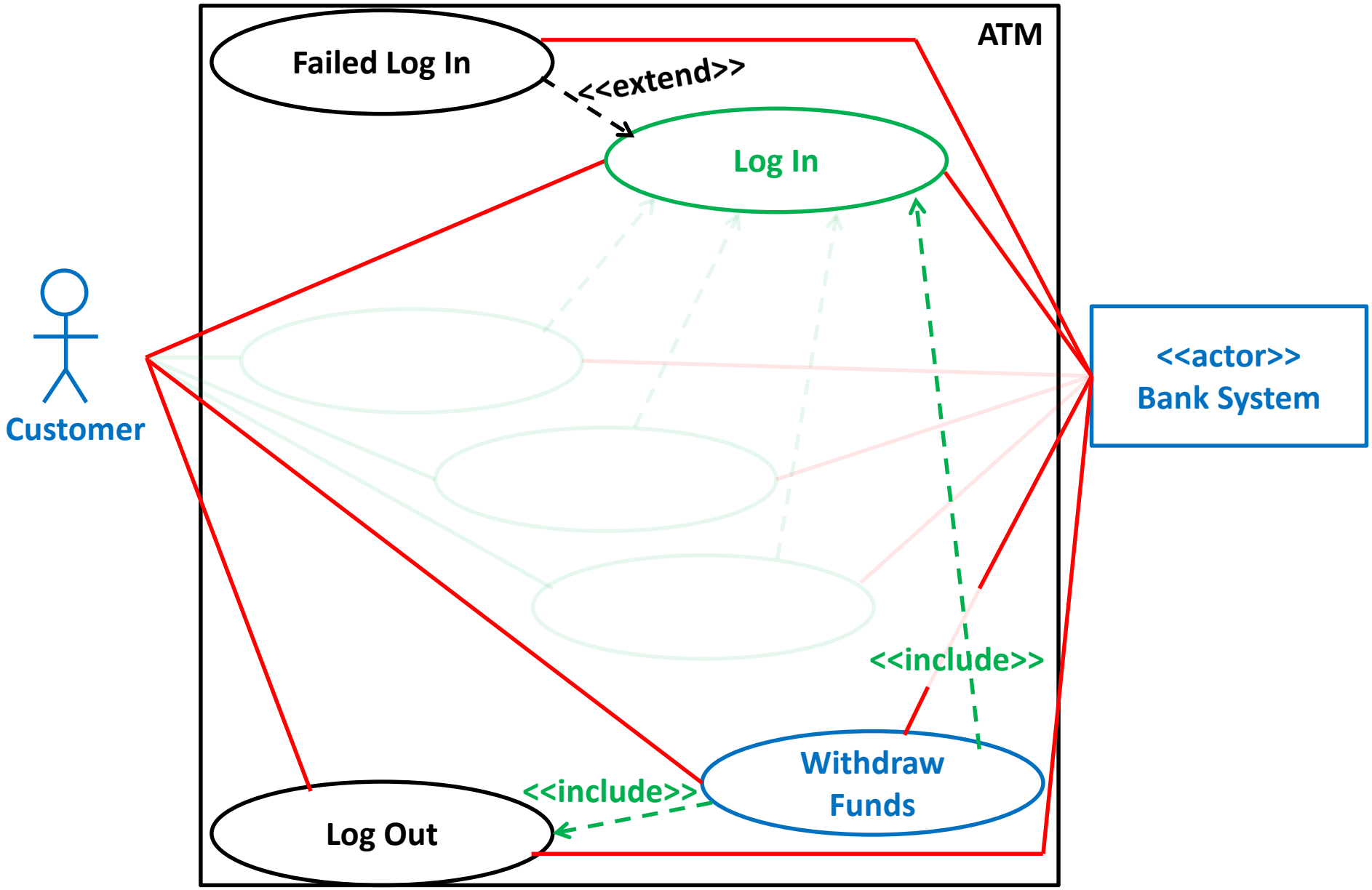
Automated Teller Machine (ATM) - 3



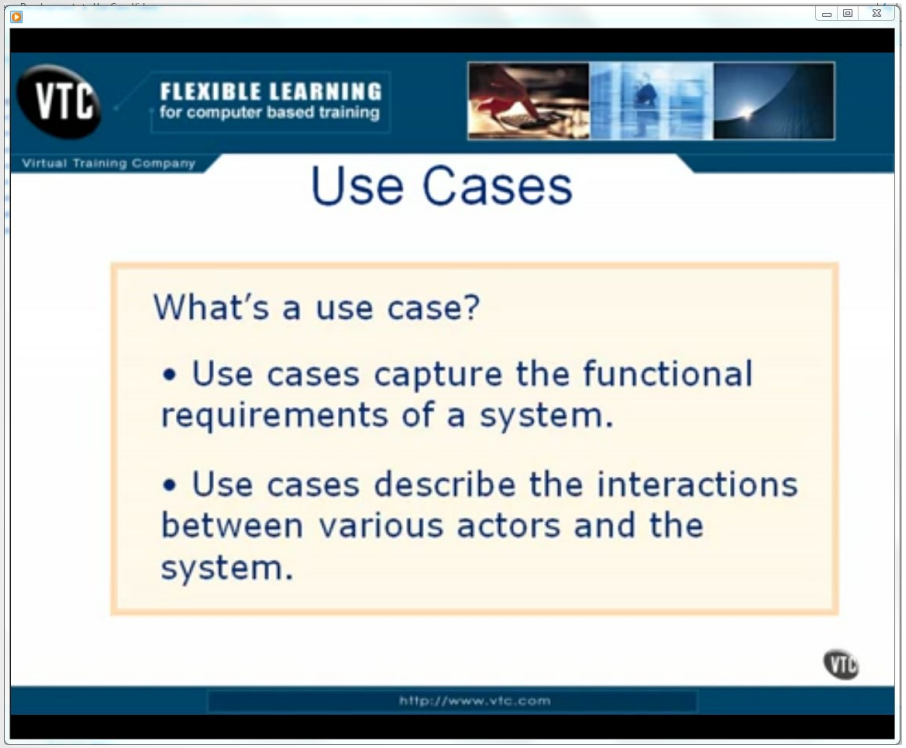
Automated Teller Machine (ATM) - 3

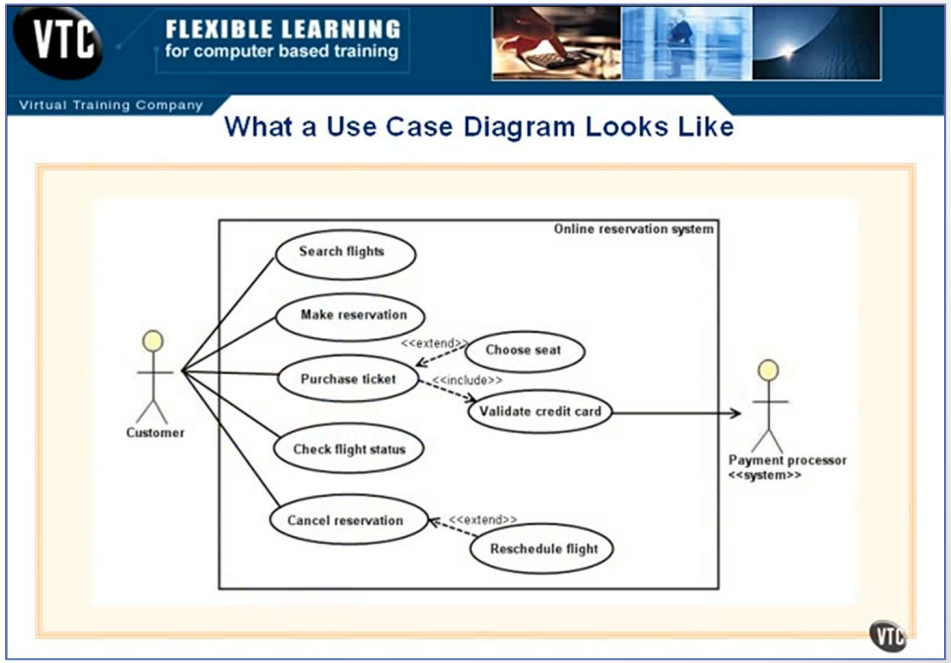


Automated Teller Machine (ATM) - 3



Video Training





Video Training

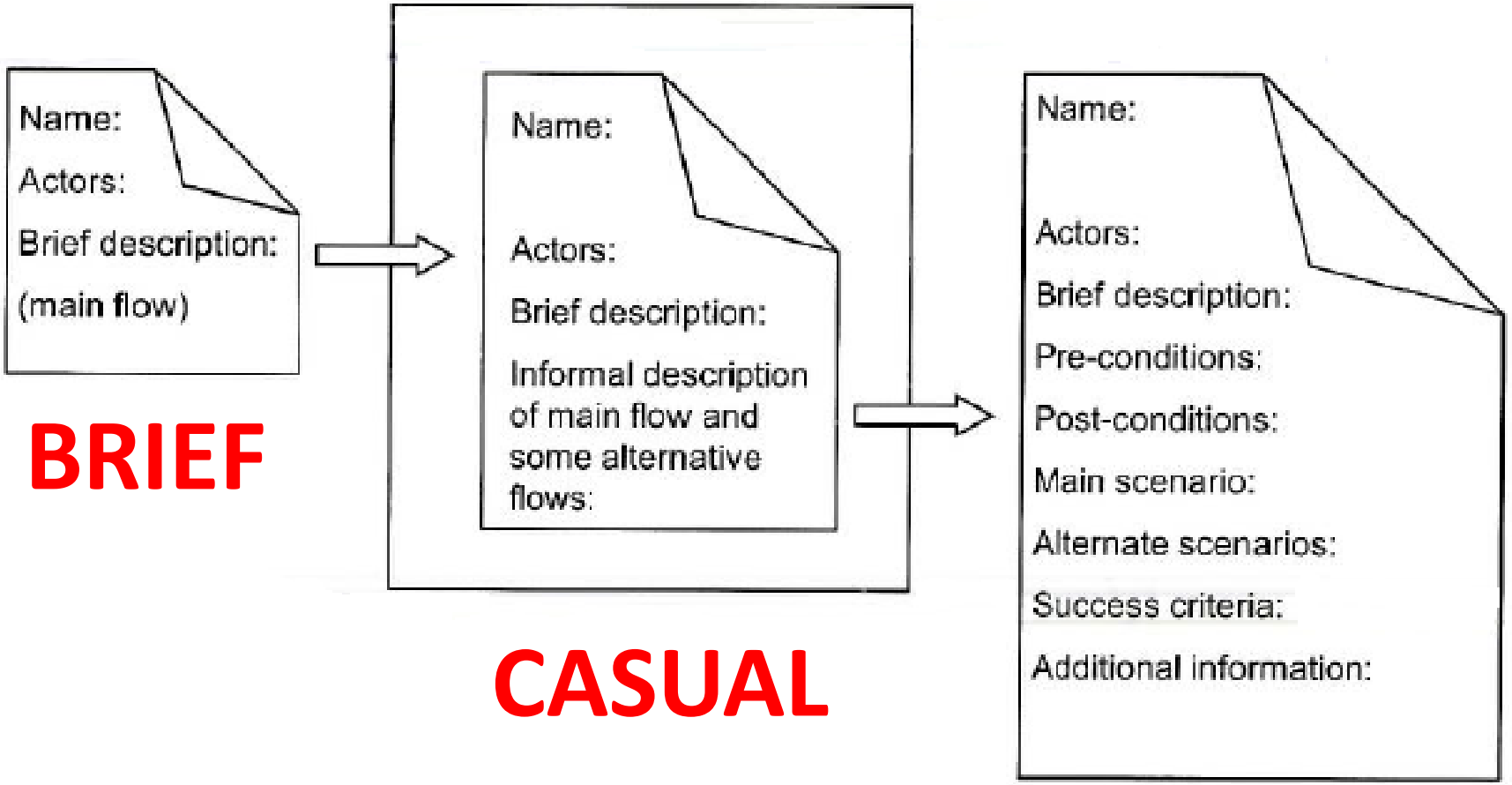
- 3.01 Use Case Basics
- 3.02 Modeling Use Case Elements
- 3.03 A Use Case Diagram for an ATM
- 3.04 The “include” Dependency
- 3.05 The “extend” Dependency
- 3.06 Generalization
- 3.07 Putting It All Together

- Available from You Tube or via Blackboard

Video Training

- Make notes (including diagrams) as you work through the videos

Use Cases Go Through Iterations



Different Styles of Presentation

The Customer (Actor)	The System (ATM)
1. Customer enters card	
	2. System asks customer to choose a language and enter a PIN
3. Customer selects a language and enters a PIN	
	4. System validates the PIN
	5. System asks for a transaction type
6. Customer selects “Withdraw Funds”	
	7. System askes for account type
8. Customer selects account	
	etc…….

Two Column Style showing “interplay” between ACTOR and SYSTEM: A good approach

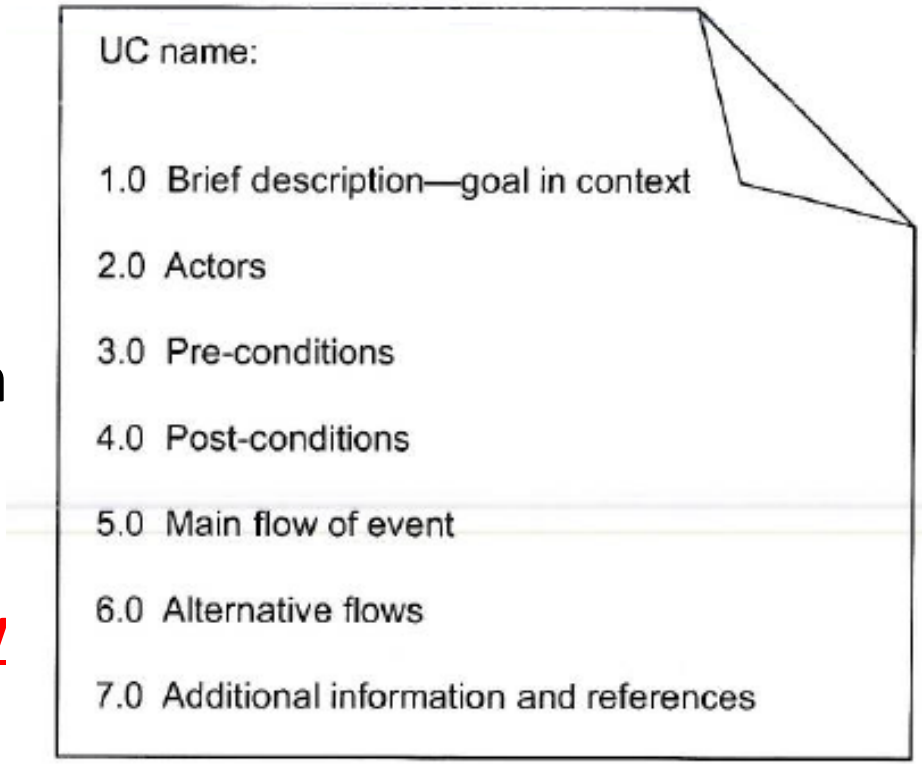
Next Steps

- *Consider the different users of the system*
- *Consider the different use cases that are relevant*

- *Work in groups to discuss and analyse the situation*

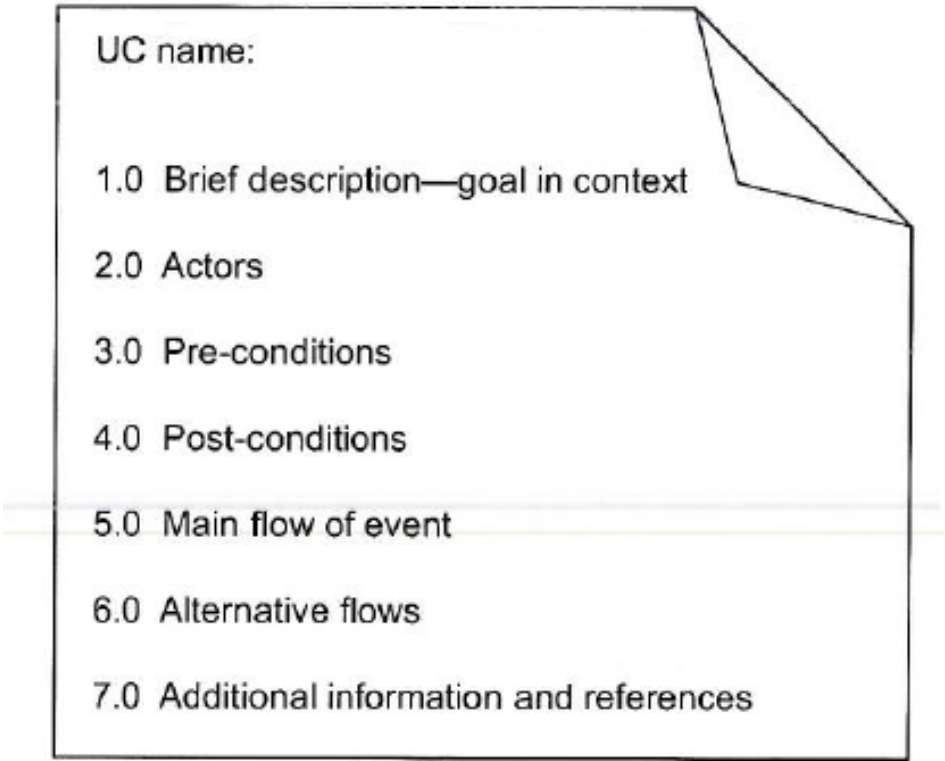
Then…..

- (A) For the **WHALER PROJECT**, draw a **Use Case Diagram** that includes relevant use cases, actors and relationships. Describe and explain your diagrammatic representation of the situation.
- (B) For a Use Case called **Choose Boat**, develop a **Fully Dressed Use Case**. Again, include any necessary description and supporting explanation.



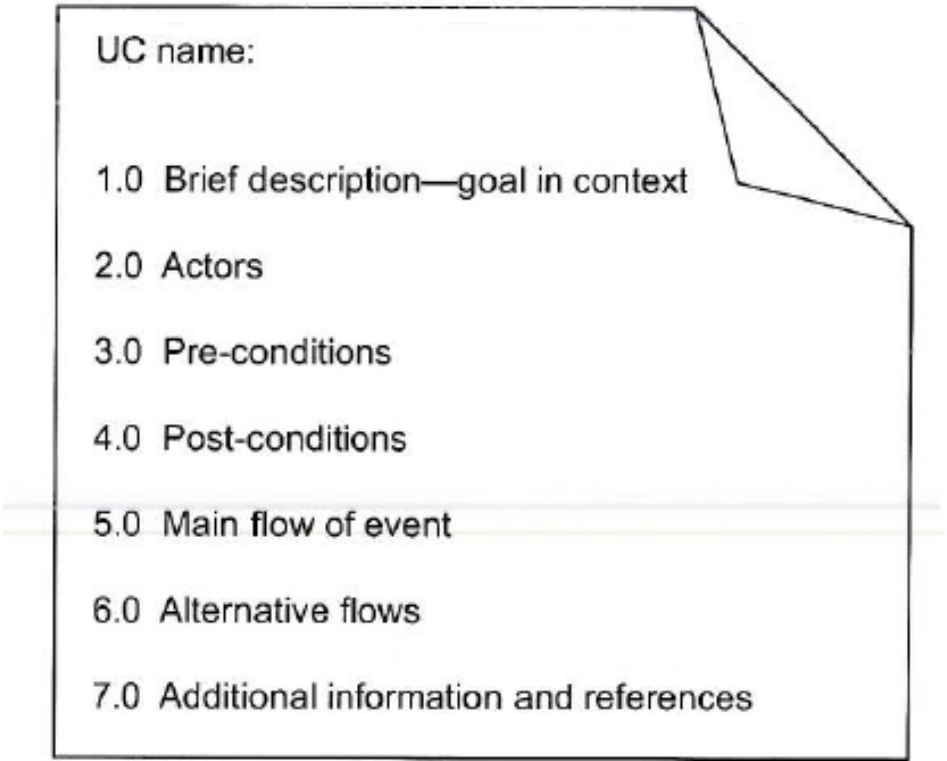
Introduction to Next Steps

- (A) Use Case Diagram
- (B) Fully Dressed Use Case for 'Choose Boat'



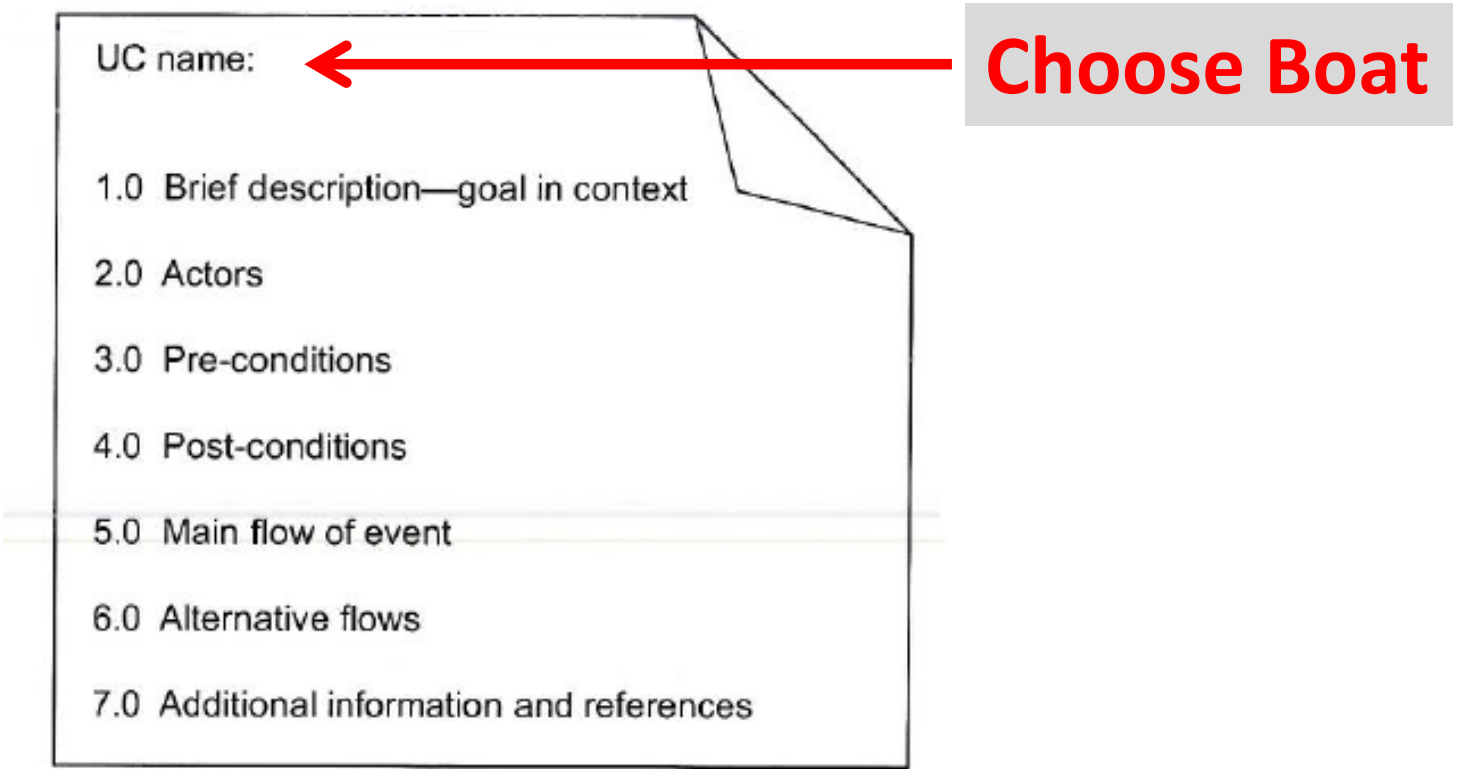
Introduction to Next Steps

- (A) Use Case Diagram
- (B) Fully Dressed Use Case for ‘Choose Boat’



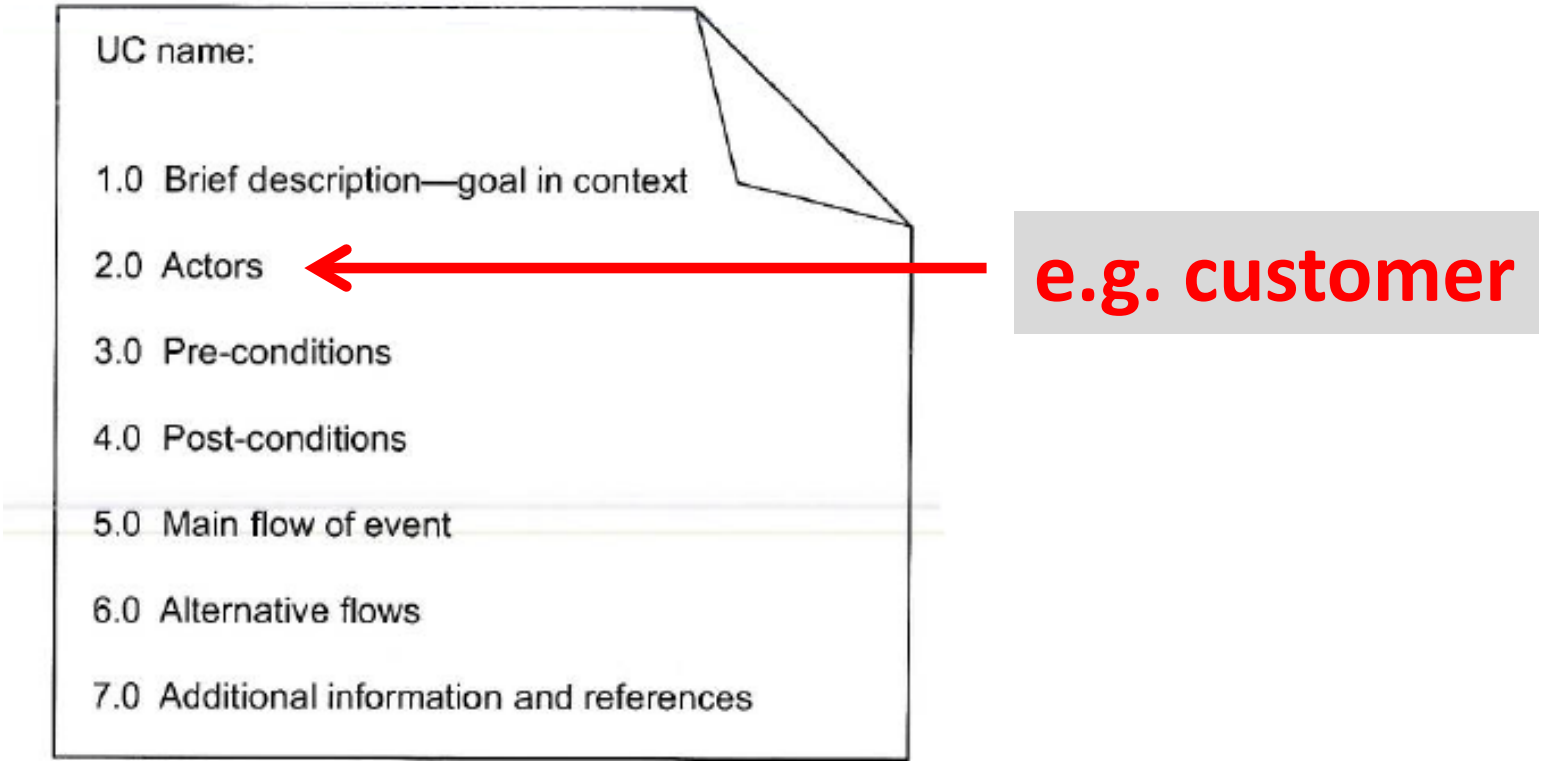
Introduction to Next Steps

- (A) Use Case Diagram
- (B) Fully Dressed Use Case for 'Choose Boat'



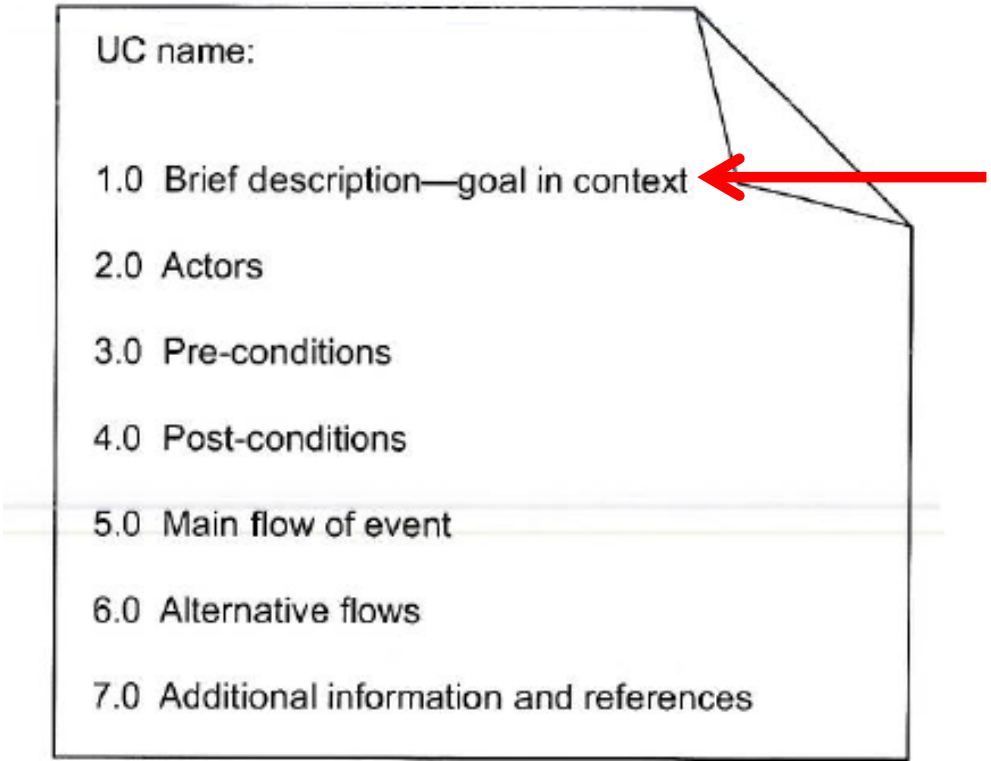
Introduction to Next Steps

- (A) Use Case Diagram
- (B) Fully Dressed Use Case for ‘Choose Boat’



Introduction to Next Steps

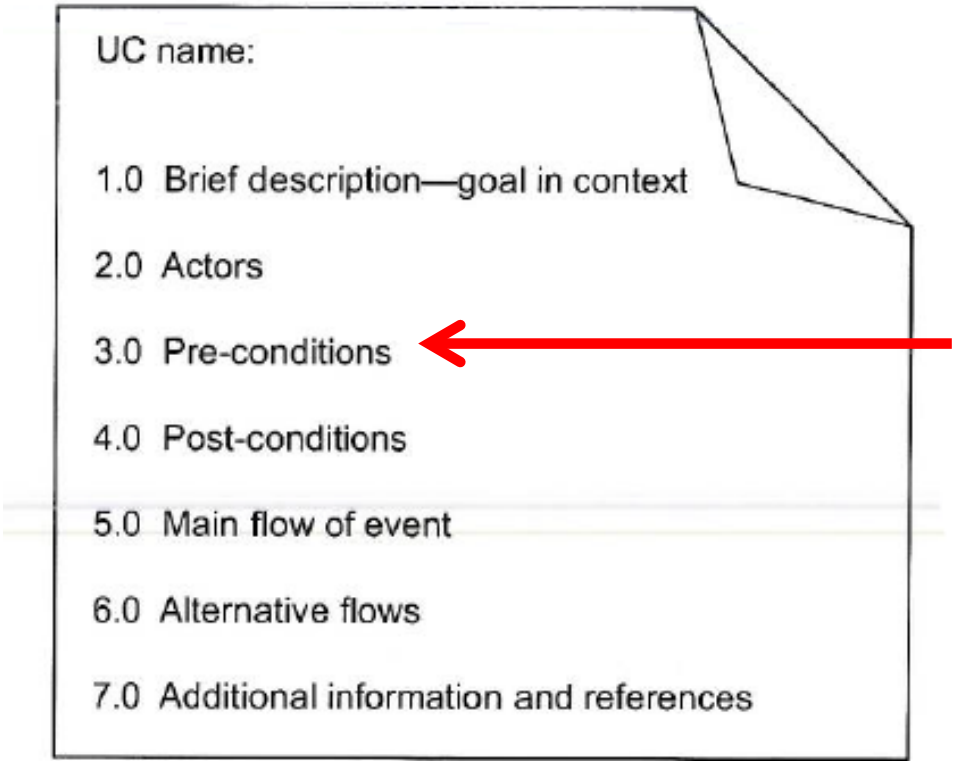
- (A) Use Case Diagram
- (B) Fully Dressed Use Case for 'Choose Boat'



Main goal for customer

Introduction to Next Steps

- (A) Use Case Diagram
- (B) Fully Dressed Use Case for 'Choose Boat'



Conditions assumed to be true at the start of this use case

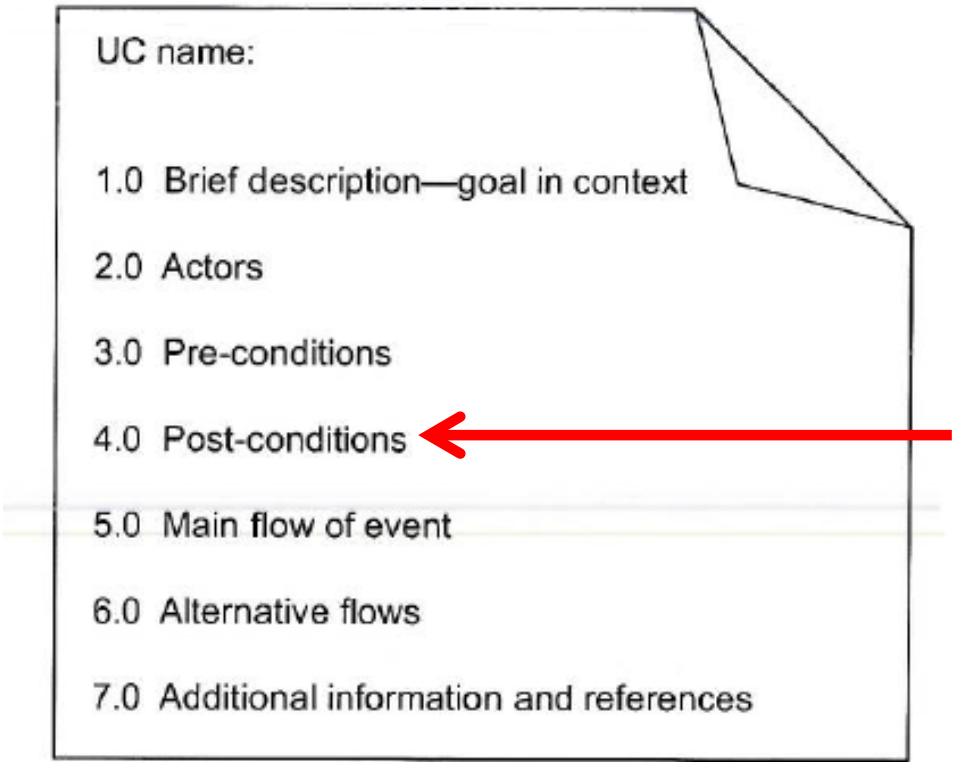
ATM: Withdraw Funds

Pre-Conditions:

- **ATM is operational**
- **ATM has been filled**
- **with cash**

Introduction to Next Steps

- (A) Use Case Diagram
- (B) Fully Dressed Use Case for 'Choose Boat'



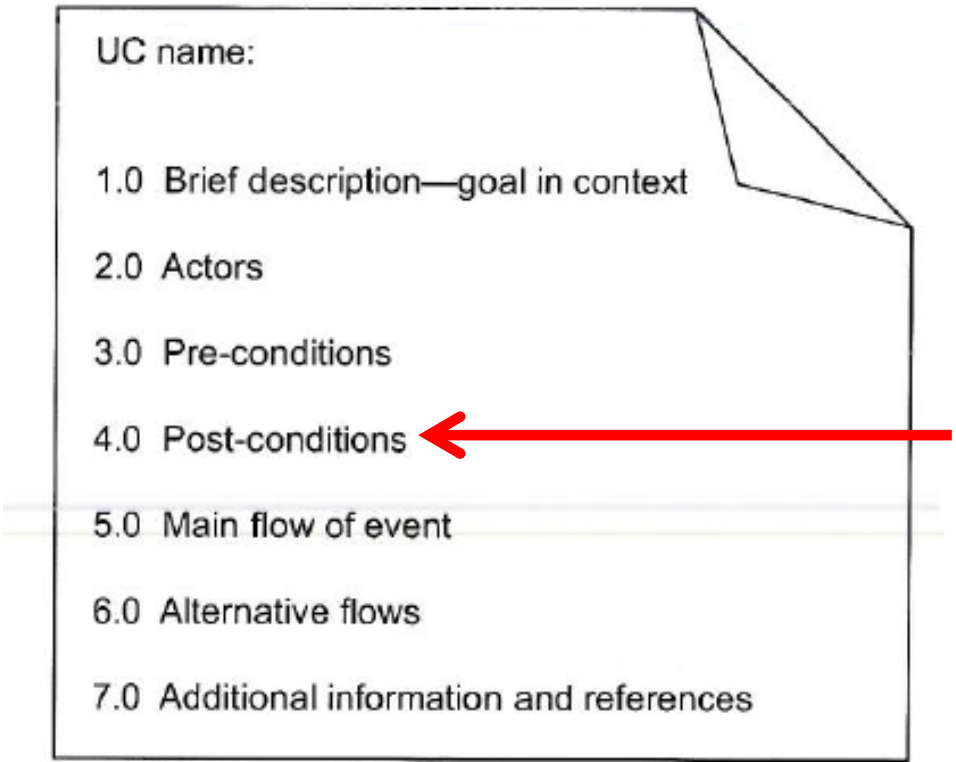
All the potential states of the system at the end of the use case

Successful conclusions = *"success guarantees"*

Unsuccessful conclusions = *"minimal guarantees"*

Introduction to Next Steps

- (A) Use Case Diagram
- (B) Fully Dressed Use Case for 'Choose Boat'



ATM: Withdraw Funds

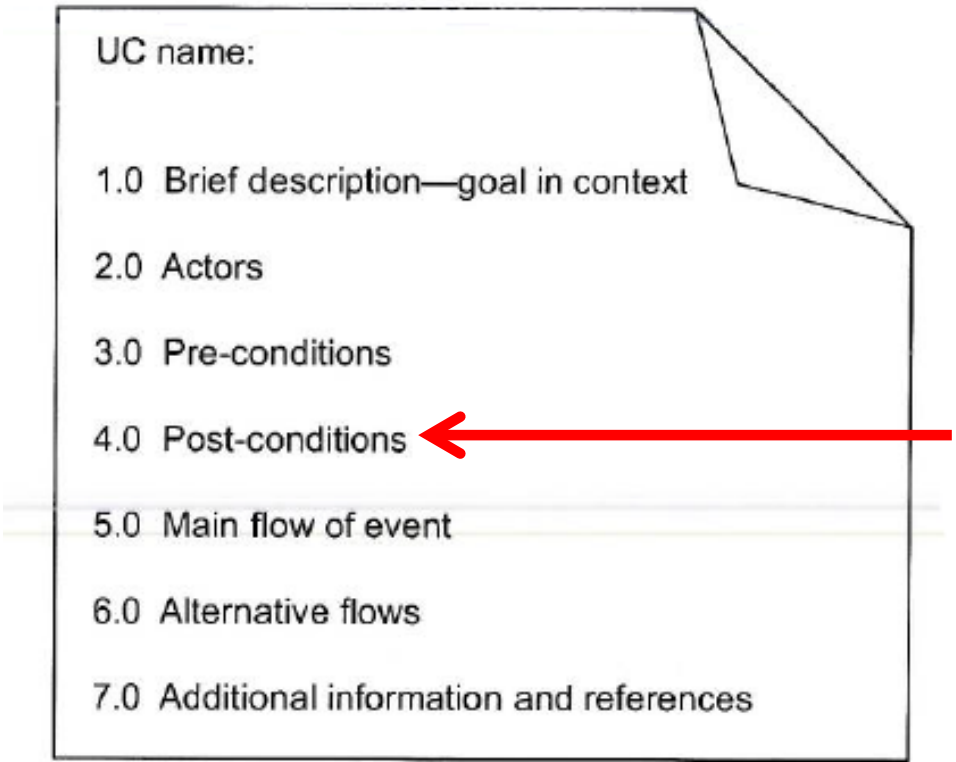
Post-Conditions:

Success guarantees

Requested funds have been dispensed; customer account has been debited by the same amount; the transaction is recorded; machine dispenses card and receipt to customer; door closes

Introduction to Next Steps

- (A) Use Case Diagram
- (B) Fully Dressed Use Case for 'Choose Boat'

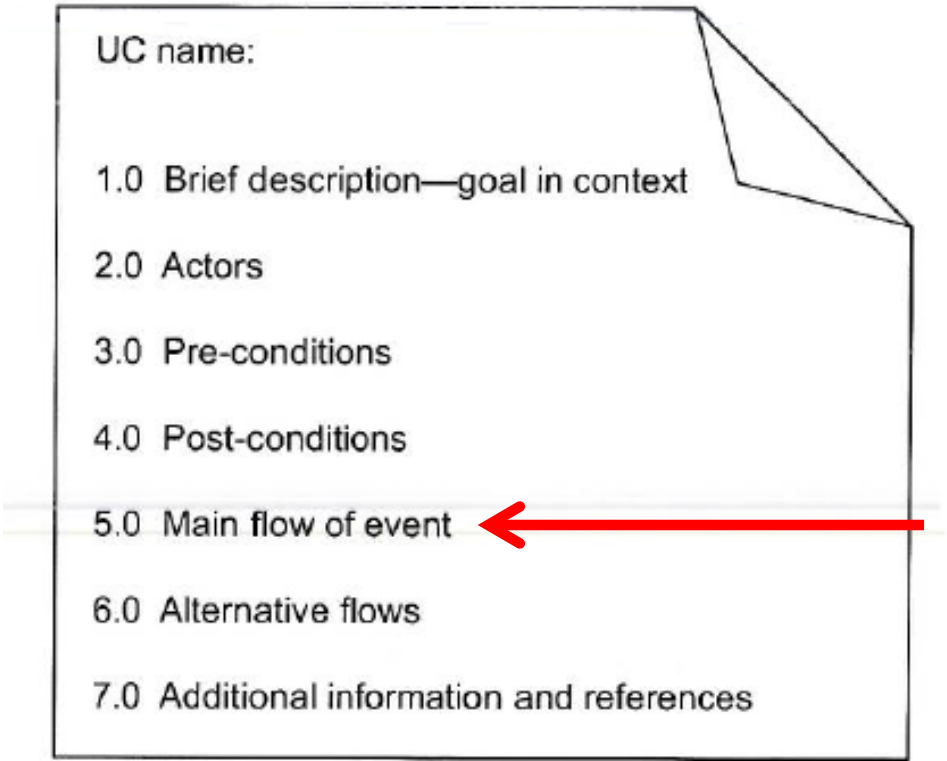


ATM: Withdraw Funds

Post-Conditions:
Minimal guarantees
Machine records attempted transaction; total amount of funds in account is unchanged

Introduction to Next Steps

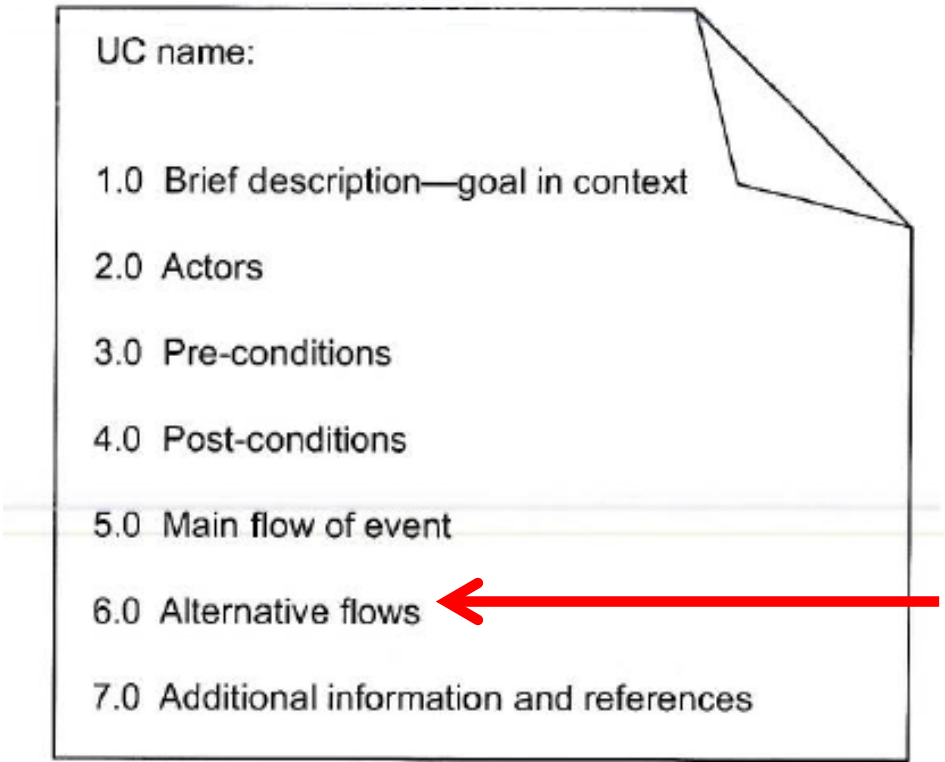
- (A) Use Case Diagram
- (B) Fully Dressed Use Case for 'Choose Boat'



The Customer (Actor)	The System (ATM)
1. Customer enters card	
	2. System asks customer to choose a language and enter a PIN
3. Customer selects a language and enters a PIN	
	4. System validates the PIN
	5. System asks for a transaction type
6. Customer selects "Withdraw Funds"	
	7. System askes for account type
8. Customer selects account	
	etc…….

Introduction to Next Steps

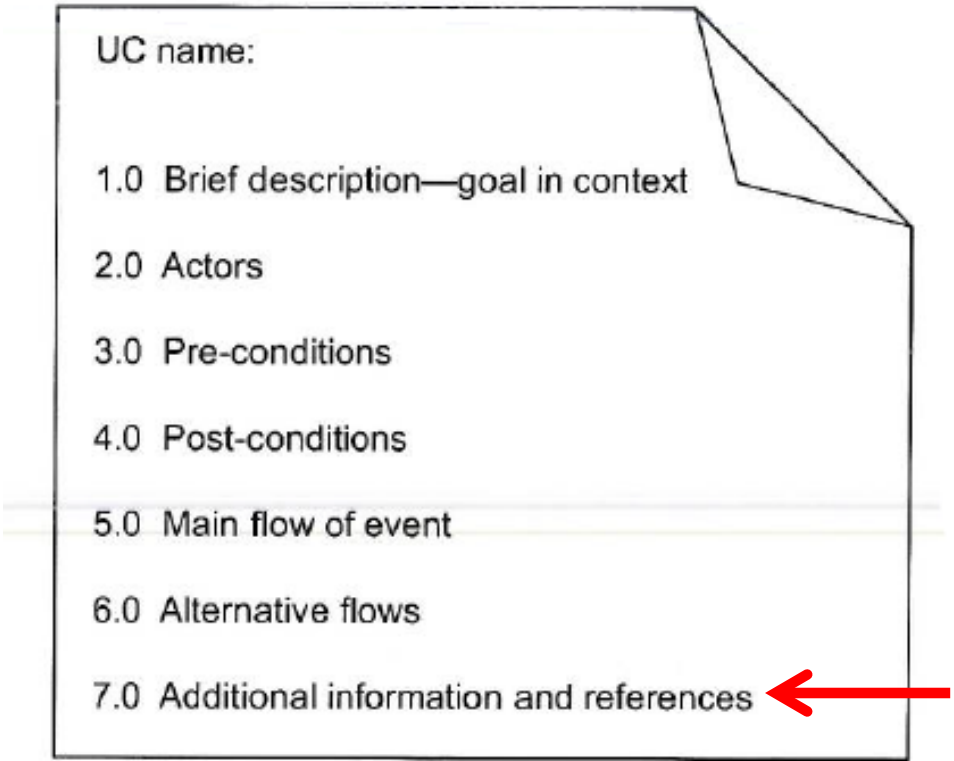
- (A) Use Case Diagram
- (B) Fully Dressed Use Case for 'Choose Boat'



After identifying and describing the main (primary) flow, identify alternative flows including points of departure from and return to the main flow.

Introduction to Next Steps

- (A) Use Case Diagram
- (B) Fully Dressed Use Case for 'Choose Boat'



Any other relevant information and explanation, for example definitions of terms used, any assumptions made, etc.

Conclusion

- Examined the utilisation of use cases and use case diagrams to model the interactions between users and the system
- Extended our understanding of the situation
- Better informed as we start planning the design and then development of the system