Requirement Engineering

Use Case Modeling

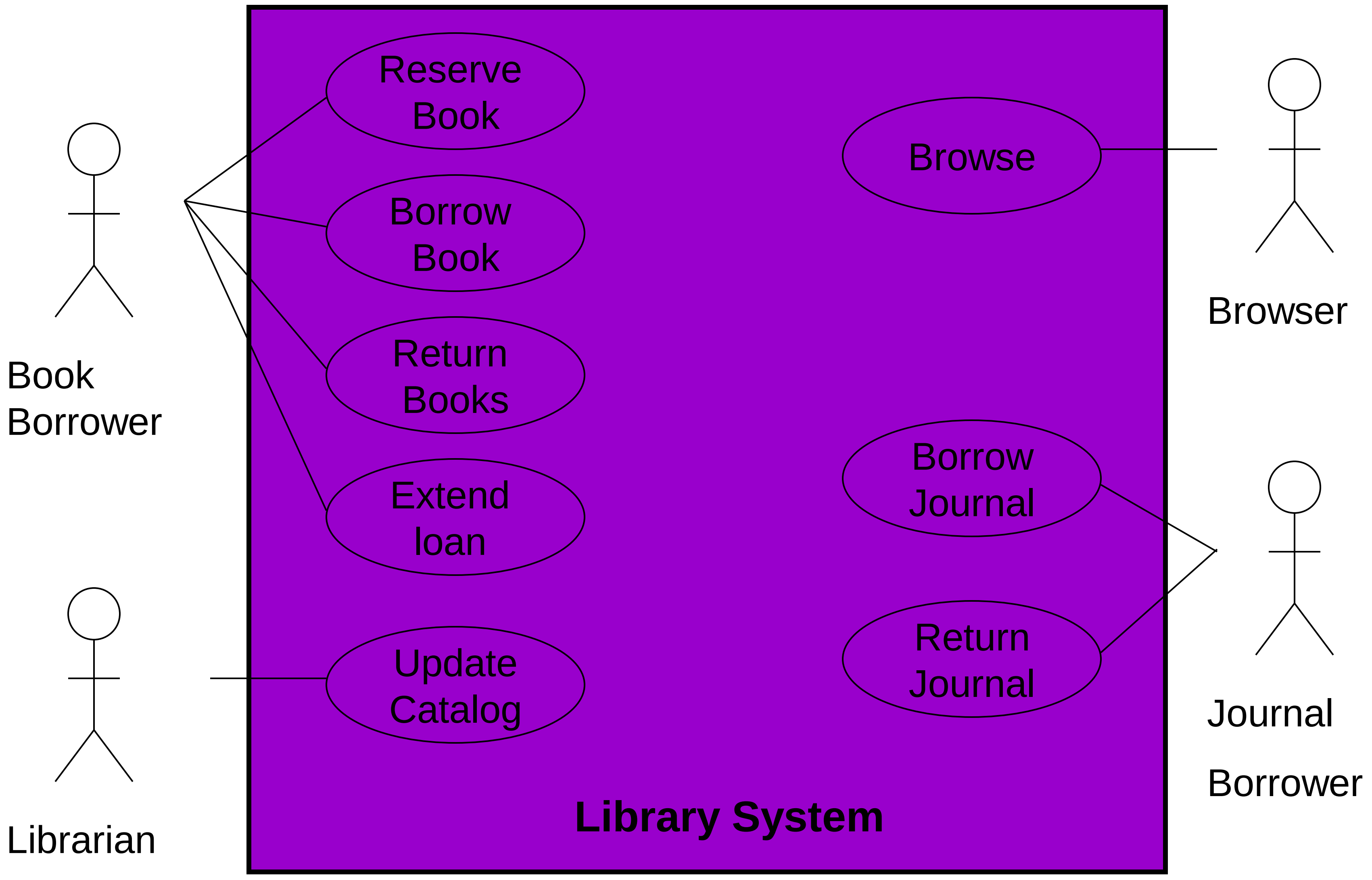
Use Case – Ivar Jacobson, 1994

- Modeling technique used to describe what a new system should do or what an existing system already does.
- Captures a discussion process between the system developer and the customer
- Comparatively easy to understand intuitively – even without knowing the notation.
- Can be easily discussed with the customer who may not be familiar with UML.
- Leads to a requirement specification on which all agree

Use Case Model Components

- Use cases
  - Boundaries of the system are defined by functionality that is handled by the system
  - Each use case specifies a complete functionality – from its initiation by an actor until it has performed the requested functionality
  - A use must always deliver some value to the actor
- Actors
  - An entity that has an interest in interacting with the system – a human or some other device or system

Use Diagram for a Library System

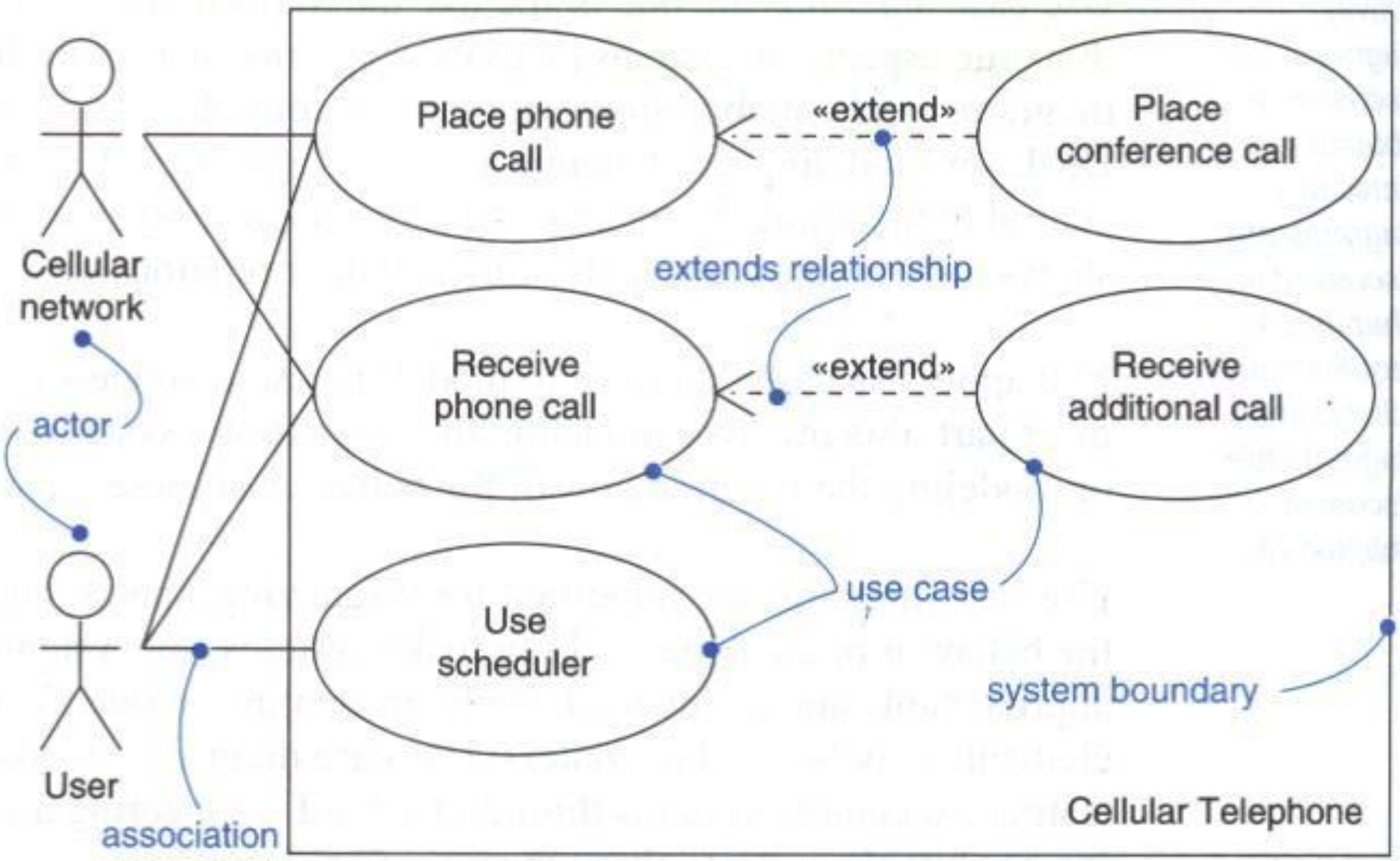


Use Case Model

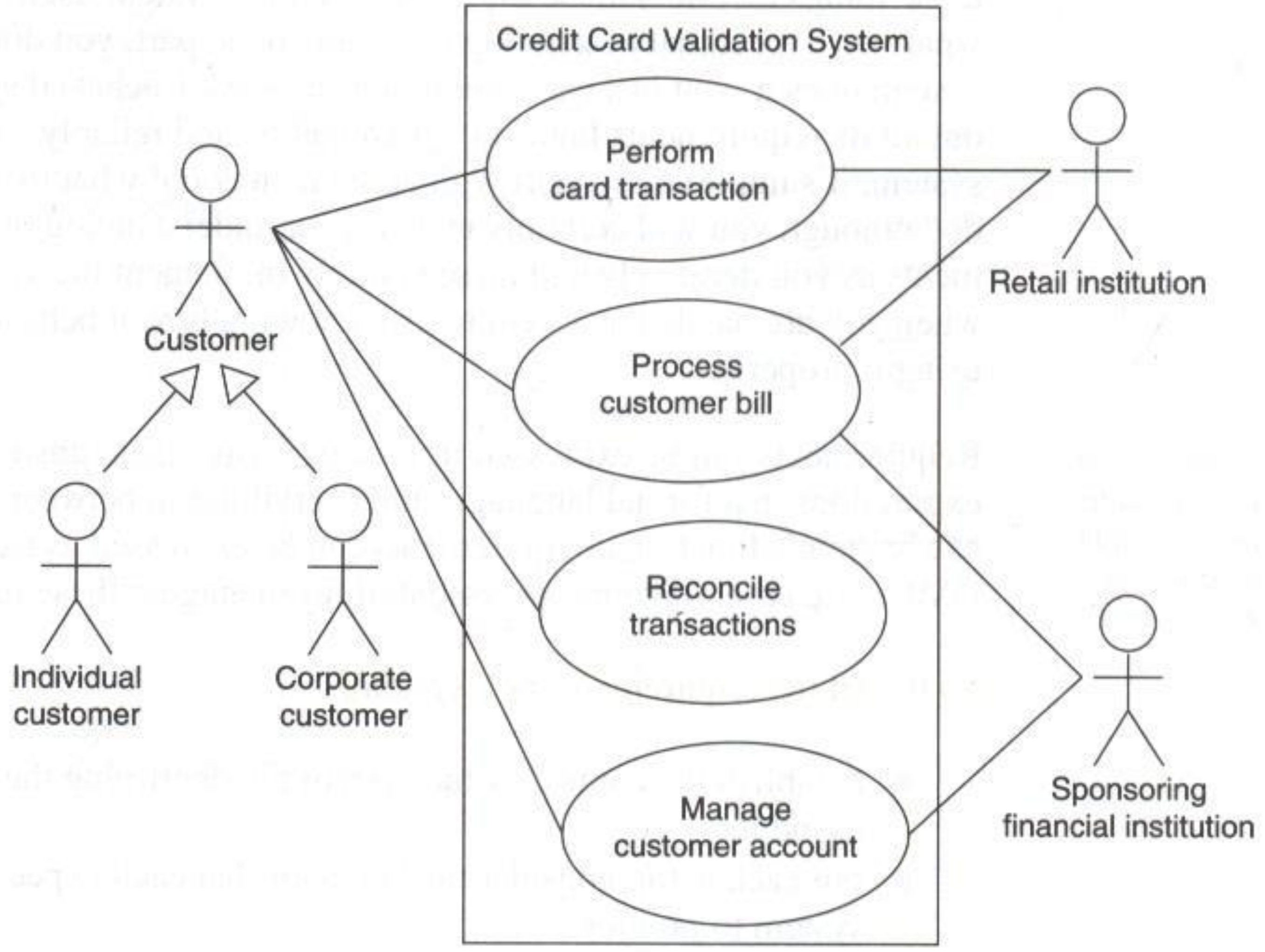
- Represents a use case view of the system – how the system is going to be used
- System is treated as a black box
- Aid to the user
  - decide and describe the functional requirements of the system
- Aid to the developer
  - give a clear and consistent description of what the system should do – model is used throughout the development process.
- Aid to the tester
  - Provide a basis for performing system tests that verify the system
- Traceability
  - Provide the ability to trace functional requirements into actual classes and operations in the system

Creating the Use Case Model

- In an iterative cycle
  - Finding the actors
  - Finding the use cases
  - Defining the system
  - Defining the relationship between use cases
- Validating the model



A Use Case Diagram



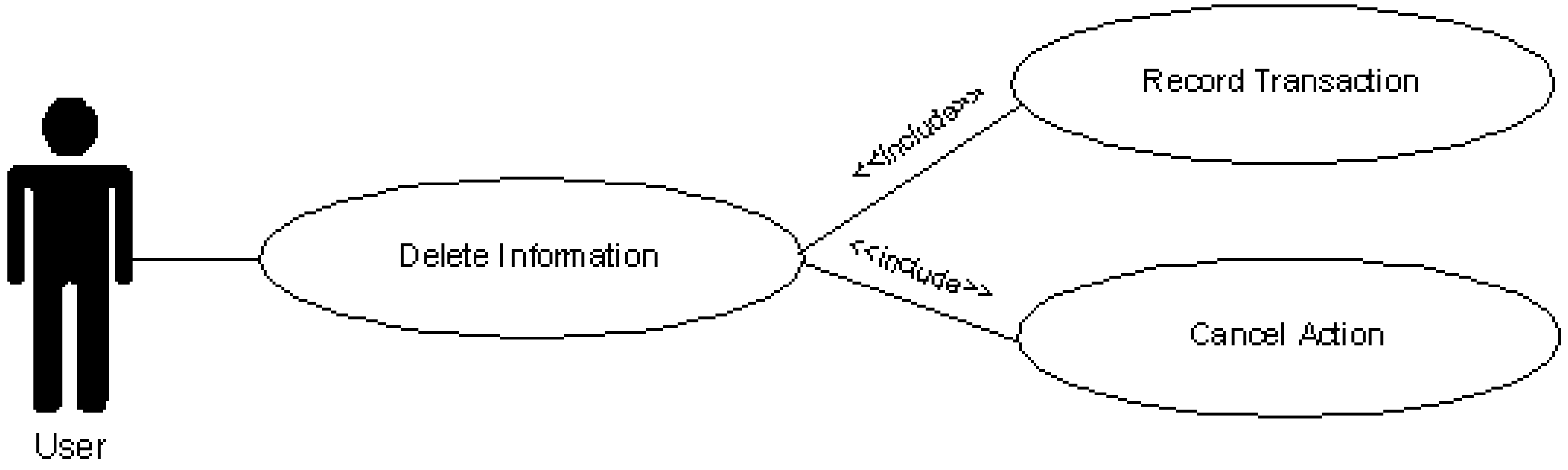
Modeling the Context of a System

Figure shows the context of a credit card validation system,

Components Of A Use Case

1. Priority
2. Actor
3. Summary
4. Precondition
5. Post- Condition
6. Extend
7. Normal Course of Events
8. Alternative Path
9. Exception
10. Assumption

Delete Information



Delete Information – Include

Use Case Relationships

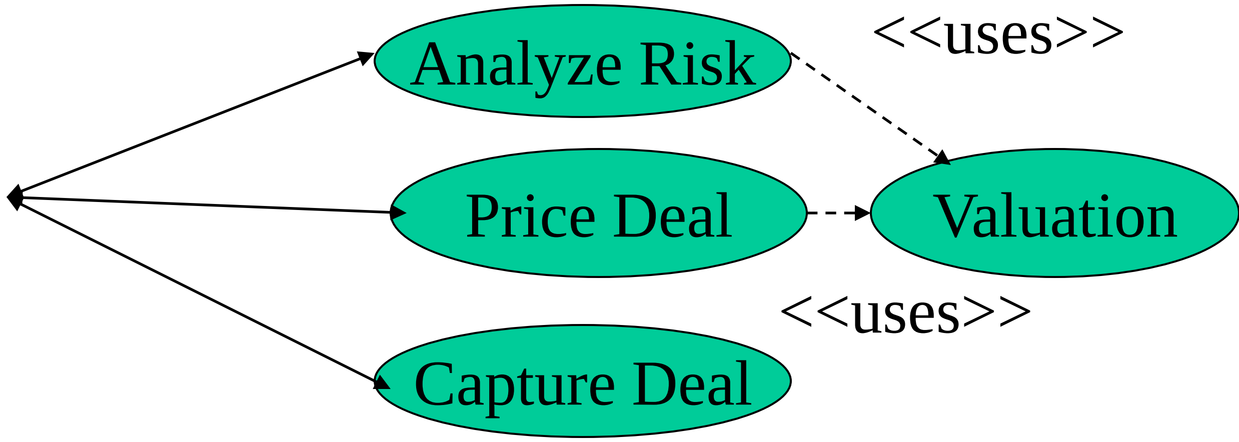
- Include or Uses
- Extend

Use Case Relationships: Includes

- One use case invokes the steps defined in another use case during the course of its own execution.
- The first use case often depends on the outcome of the included use case
- Behavior common to many use cases is used by one use case
- Denoted by **<<includes>>**
- Include use cases must be **used** by the use cases that use them

Include or Uses

– Use "include" when you are repeating yourself in two or more separate use cases and you want to avoid repetition

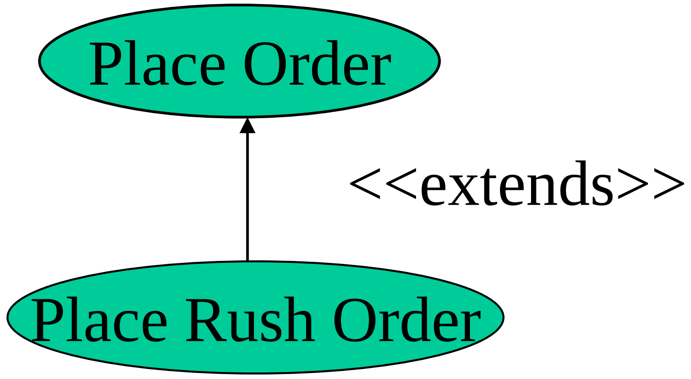


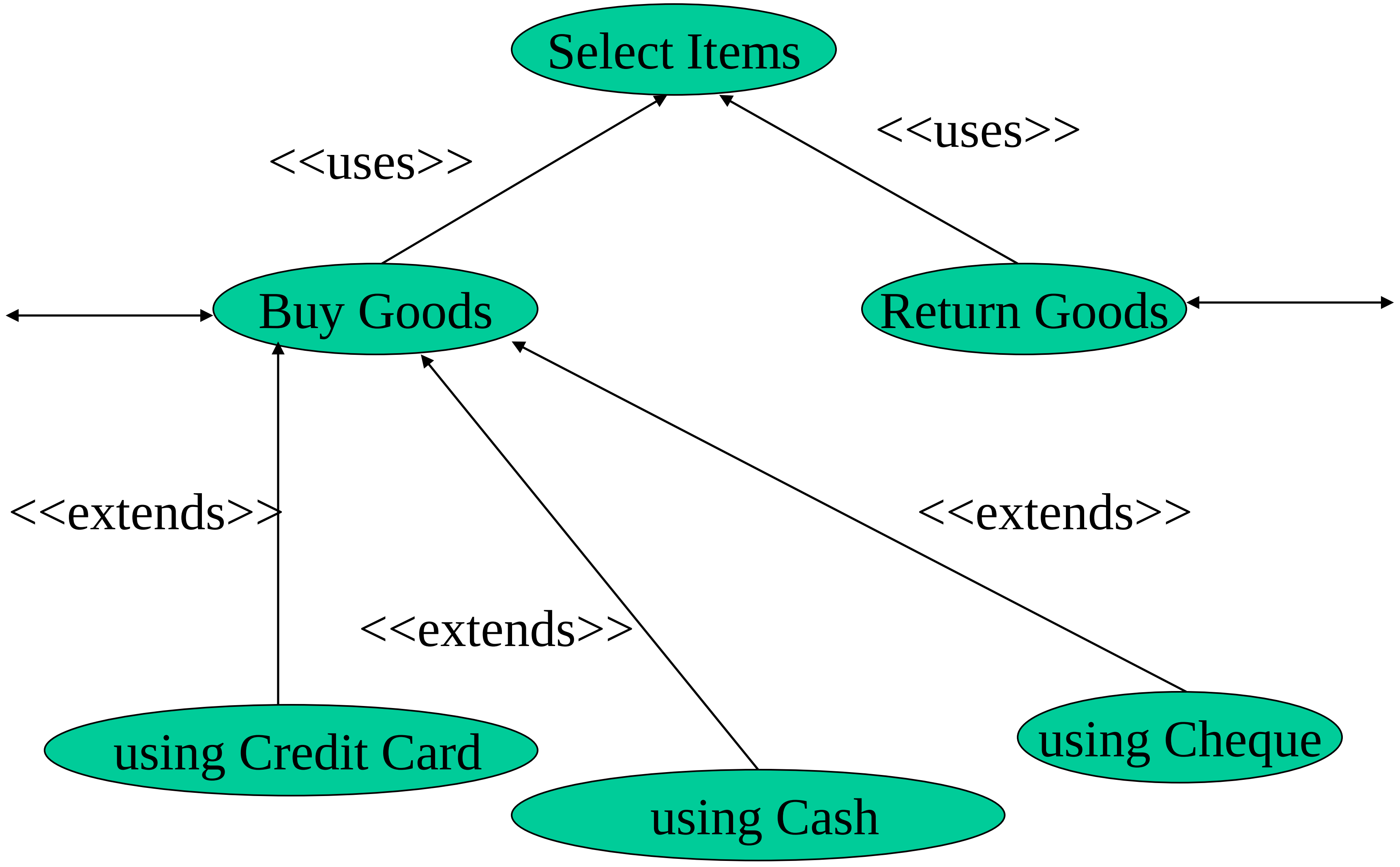
Use Case Relationships: Extend

- A use case **extends** another use case to do more than the latter
- Adds more information to allow the already complete use case to extend its use
- Denoted by <<extend>>
- Eg. "Place Conference Call" use case extends "Place Phone Call" use case

Extend

- Use “extend” when you are describing a variation on normal behavior and you wish to use the more controlled form, declaring your extension points in your base use case.




Select Items
<<uses>>
<<uses>>
Buy Goods
Return Goods
<<extends>>
<<extends>>
<<extends>>
using Credit Card
using Cash
using Cheque

Difference between the two..

- Includes
  - "X *includes* Y" indicates that the task "X" **has a** subtask "Y"; that is, in the process of completing task "X", task "Y" will be completed at least once.
- Extends

  - "X extends Y" indicates that "X" is a task for the same type as "Y", but "X" is a special, more specific case of doing "Y". That is, doing X is a lot like doing Y, but X has a few extra processes to it that go above and beyond the things that must be done in order to complete Y.