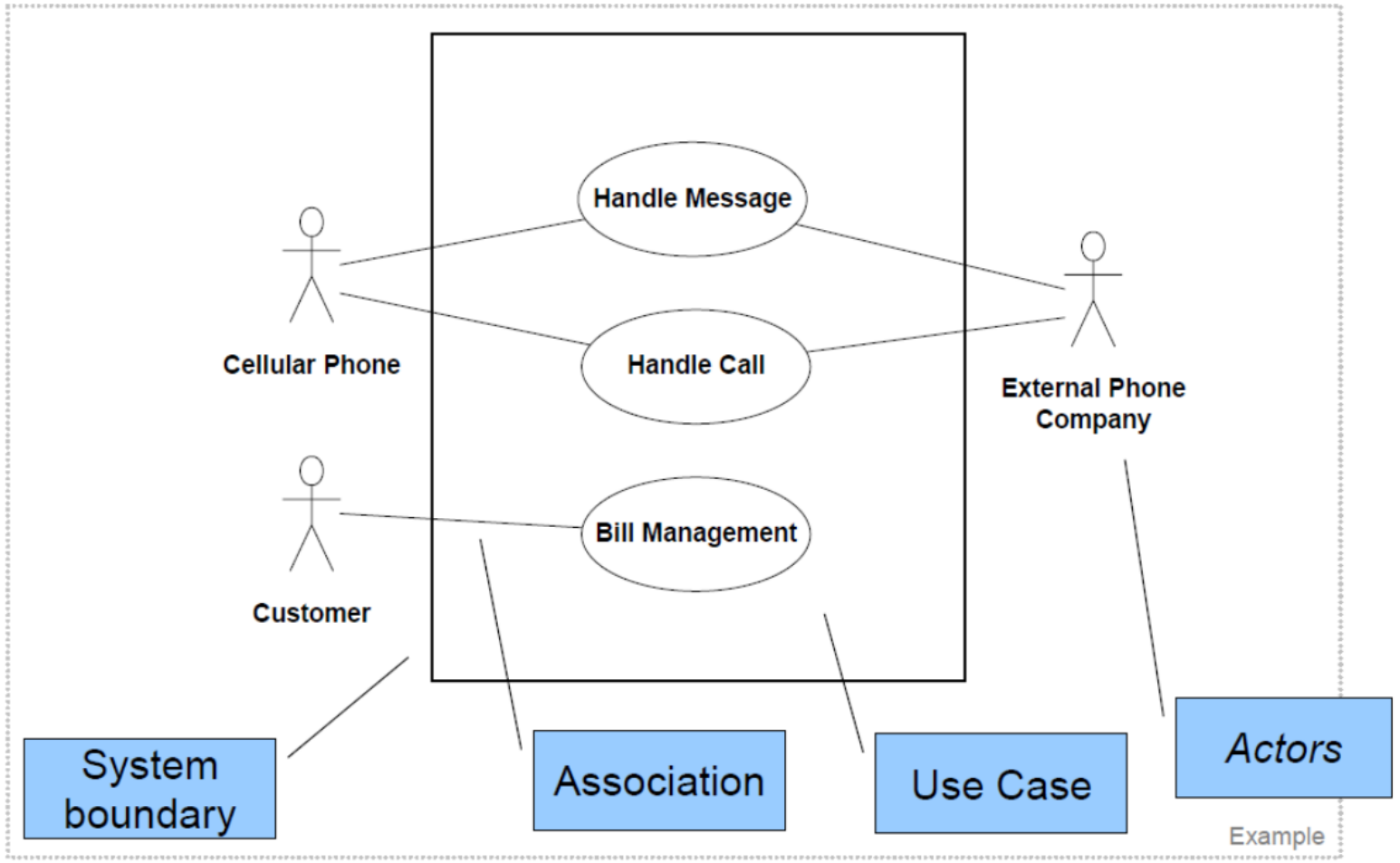
Use Case Diagrams

- A picture
  - describes how actors relate to use cases
  - and use cases relate to one another
- Diagrams are not essential
- They are helpful in giving an overview, but only secondary in importance to the textual description
- They do not capture the full information of the actual use cases
- In contrast, text ***is*** essential

Use Case Diagram Objective

- Built in early stages of development
- Purpose
  - Specify the context of a system
  - Capture the requirements of a system
  - Validate a systems architecture
  - Drive implementation and generate test cases
  - Developed by analysts and domain experts

Example Use-Case Diagram



A standard form of use case diagram is defined in the Unified Modeling Language.

What is an Actor?

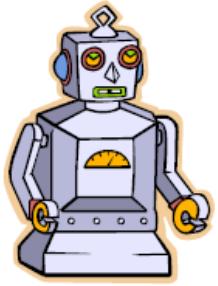
- Include all user roles that interact with the system
- Include system components only if they responsible for initiating/triggering a use case.
  - For example, a timer that triggers sending of an e-mail reminder
- ***primary*** - a user whose goals are fulfilled by the system
  - importance: define user goals
- ***supporting*** - provides a service (e.g., info) to the system
  - importance: clarify external interfaces and protocols
- ***offstage*** - has an interest in the behavior but is not primary or supporting, e.g., government
  - importance: ensure all interests (even subtle) are identified and satisfied

Finding Actors [1]

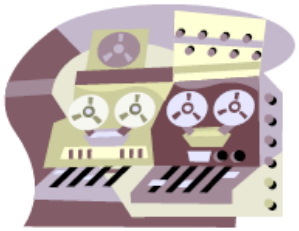
External objects that produce/consume data:

- Must serve as sources and destinations for data
- Must be external to the system

Humans



Machines



External systems

Organizational Units

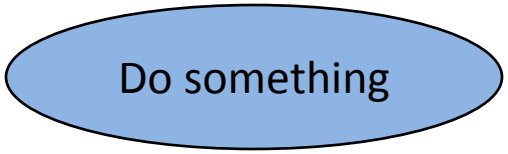
Sensors

Finding Actors [2]

Ask the following questions:

- Who are the system's primary users?
- Who requires system support for daily tasks?
- Who are the system's secondary users?
- What hardware does the system handle?
- Which other (if any) systems interact with the system in question?
- Do any entities interacting with the system perform multiple roles as actors?
- Which other entities (human or otherwise) might have an interest in the system's output?

Elements of use case diagram: Use Case



- System function (process – automated or manual).
- Named by verb.
- Each Actor must be linked to a use case, while some use cases may not be linked to actors.

USER/ACTOR	USER GOAL = Use Case
Order clerk	Look up item availability Create new order Update order
Shipping clerk	Record order fulfillment Record back order
Merchandising manager	Create special promotion Produce catalog activity report

Elements of use case diagram: Other details

Connection between Actor and Use Case

Boundary of system

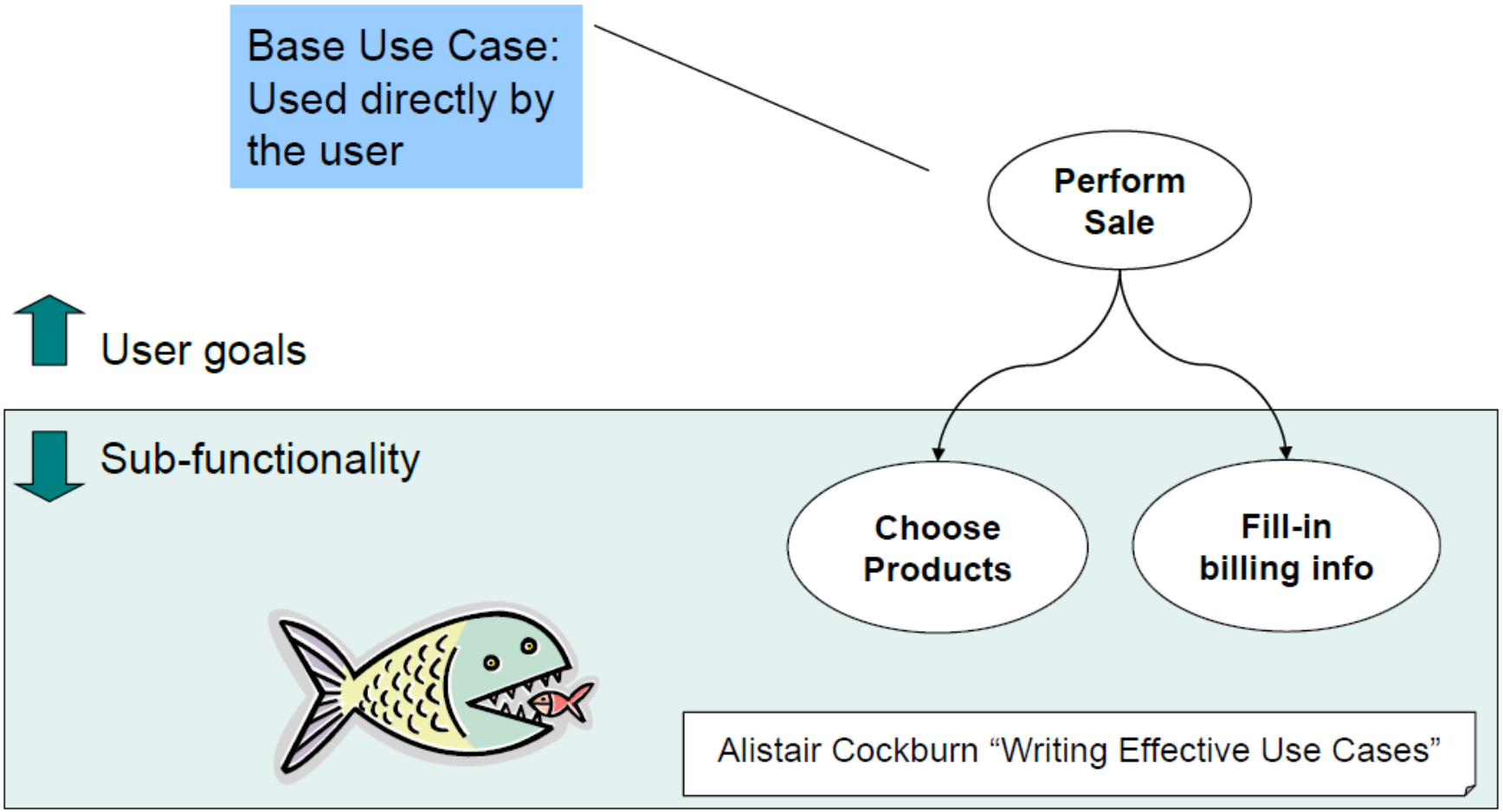
Include relationship between Use Cases (one UC must call another; e.g., Login UC includes User Authentication UC)

Extend relationship between Use Cases (one UC calls Another under certain condition; think of if-then decision points)

Linking Use Cases

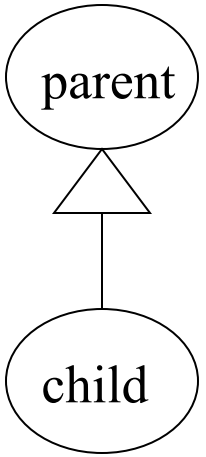
- *Association* relationships
- *Generalization* relationships
  - One element (child) "is based on" another element (parent)
- *Include* relationships
  - One use case (base) includes the functionality of another (inclusion case)
  - Supports re-use of functionality
- *Extend* relationships
  - One use case (extension) extends the behavior of another (base)

Use Case Levels

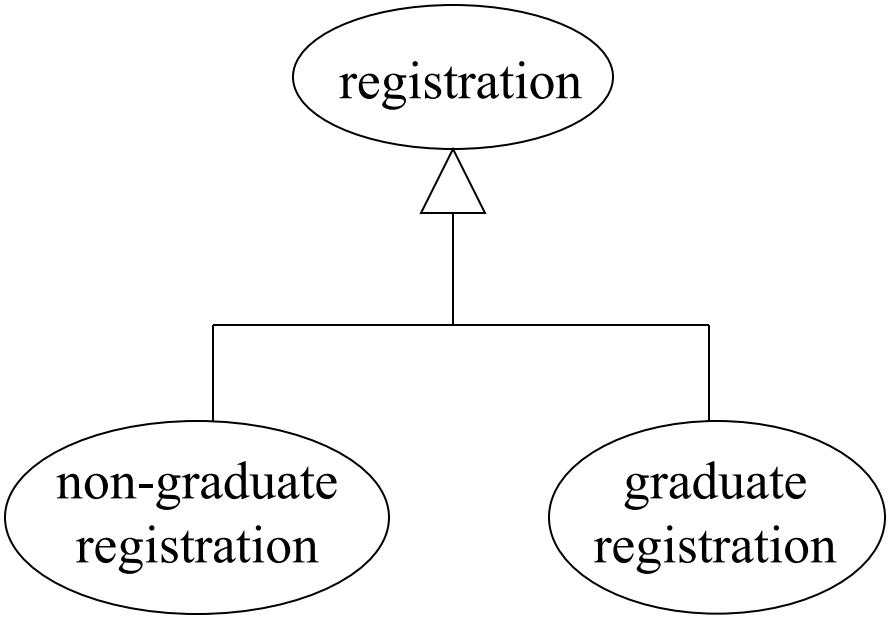


1. Generalization

- The child use case inherits the behavior and meaning of the parent use case.
- The child may add to or override the behavior of its parent.



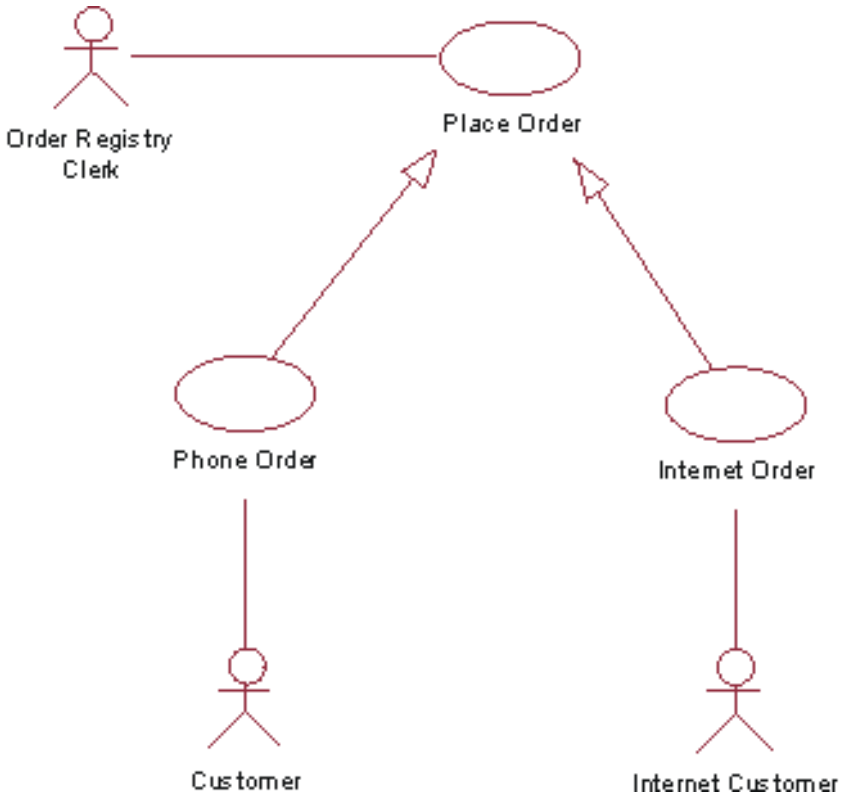
More about Generalization



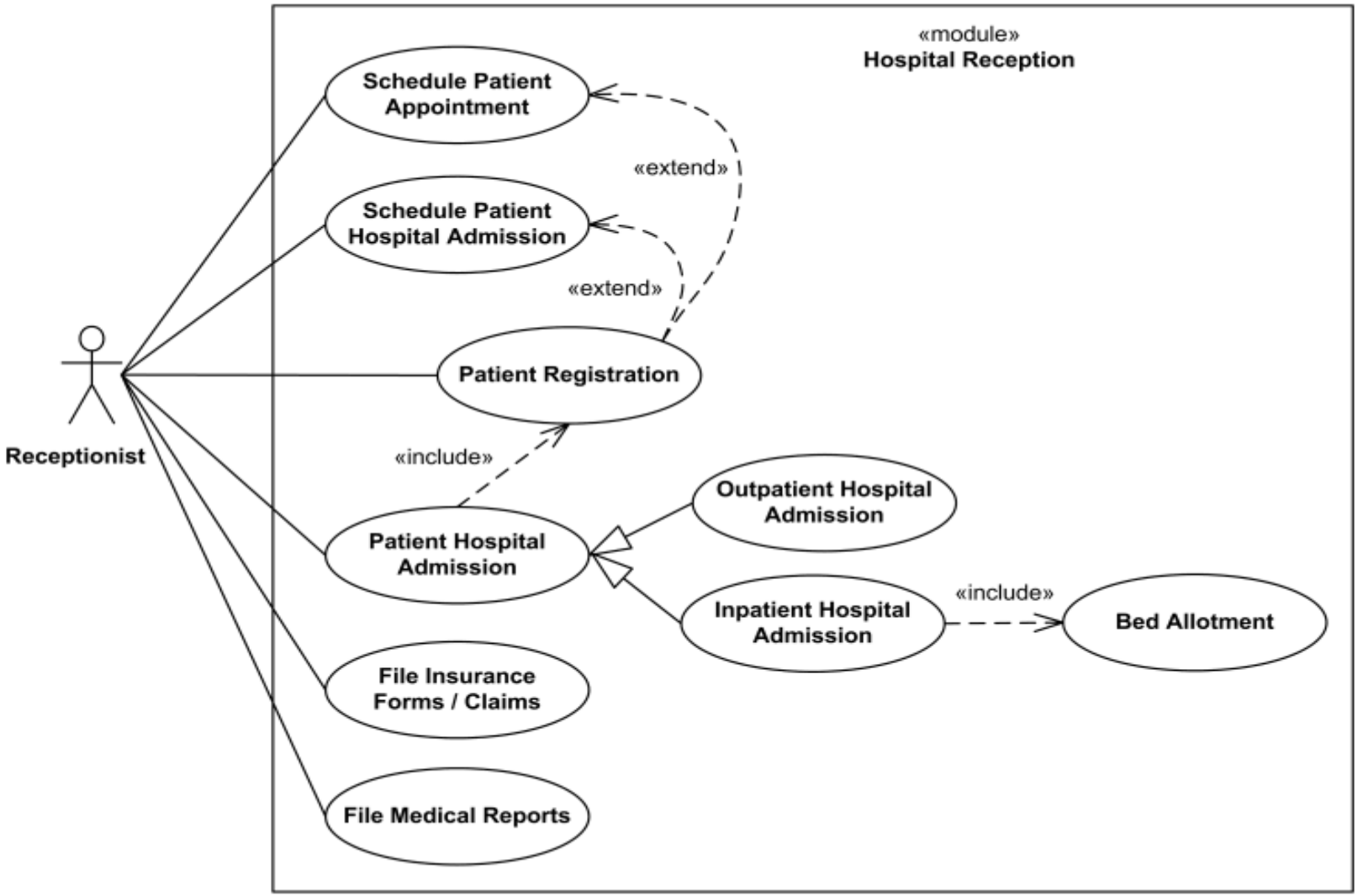
Generalization Example

The actor Order Registry Clerk can instantiate the general use case Place Order.

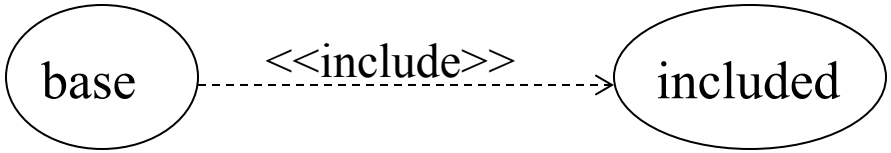
Place Order can also be specialized by the use cases Phone Order or Internet Order.



Generalization Example



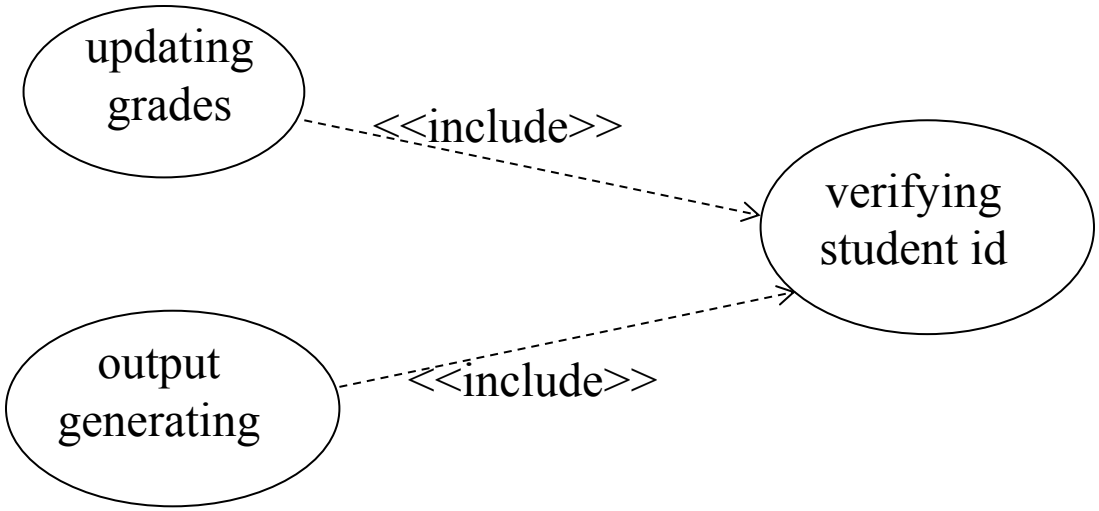
2. Include



- The base use case explicitly incorporates the behavior of another use case at a location specified in the base.
- The included use case never stands alone. It only occurs as a part of some larger base that includes it.

More about Include

Enables us to avoid describing the same flow of events several times by putting the common behavior in a use case of its own.



Include relationship

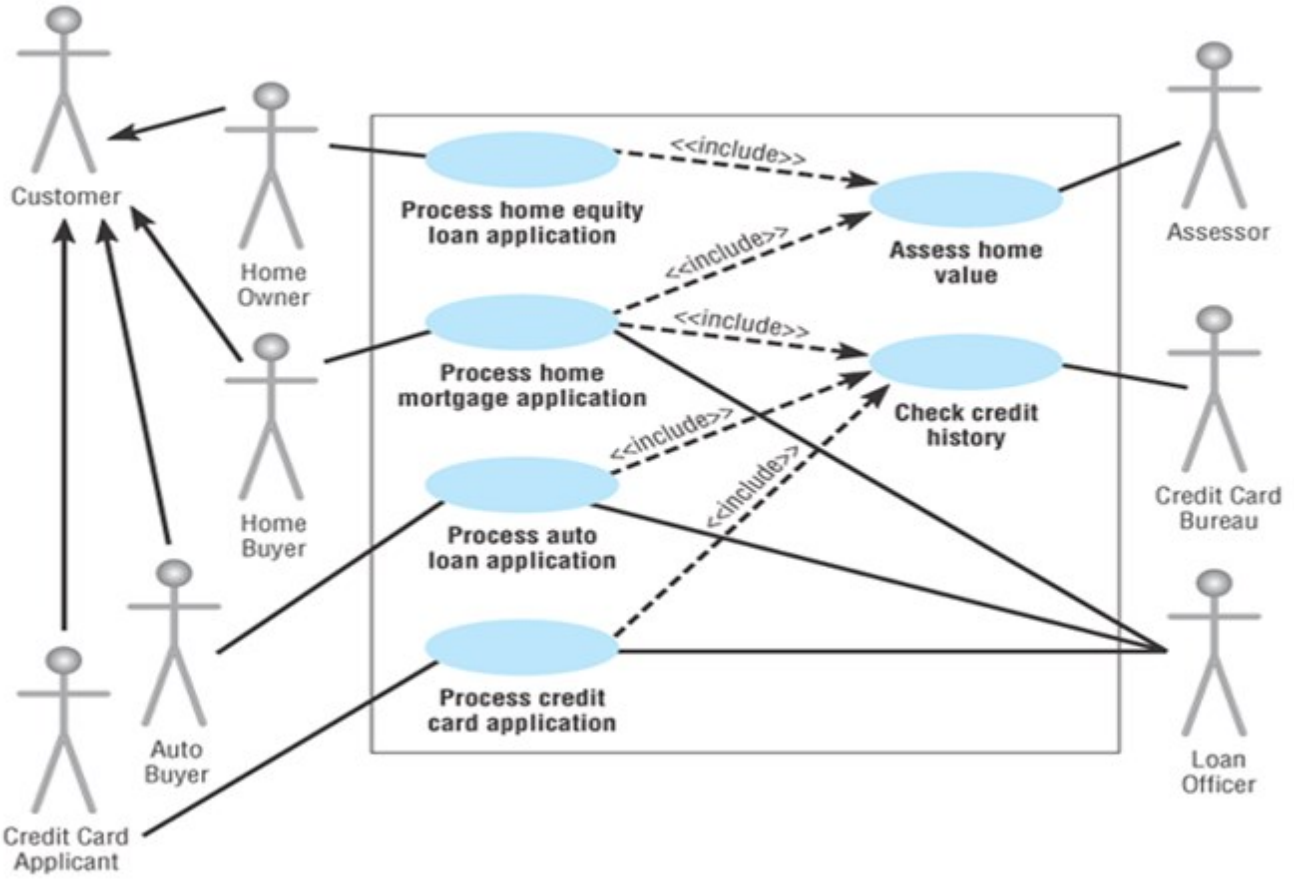
- Include relationship – a standard case linked to a ***mandatory*** use case.

- Example: to *Authorize Car Loan* (standard use case), a clerk must run *Check Client's Credit History* (include use case).

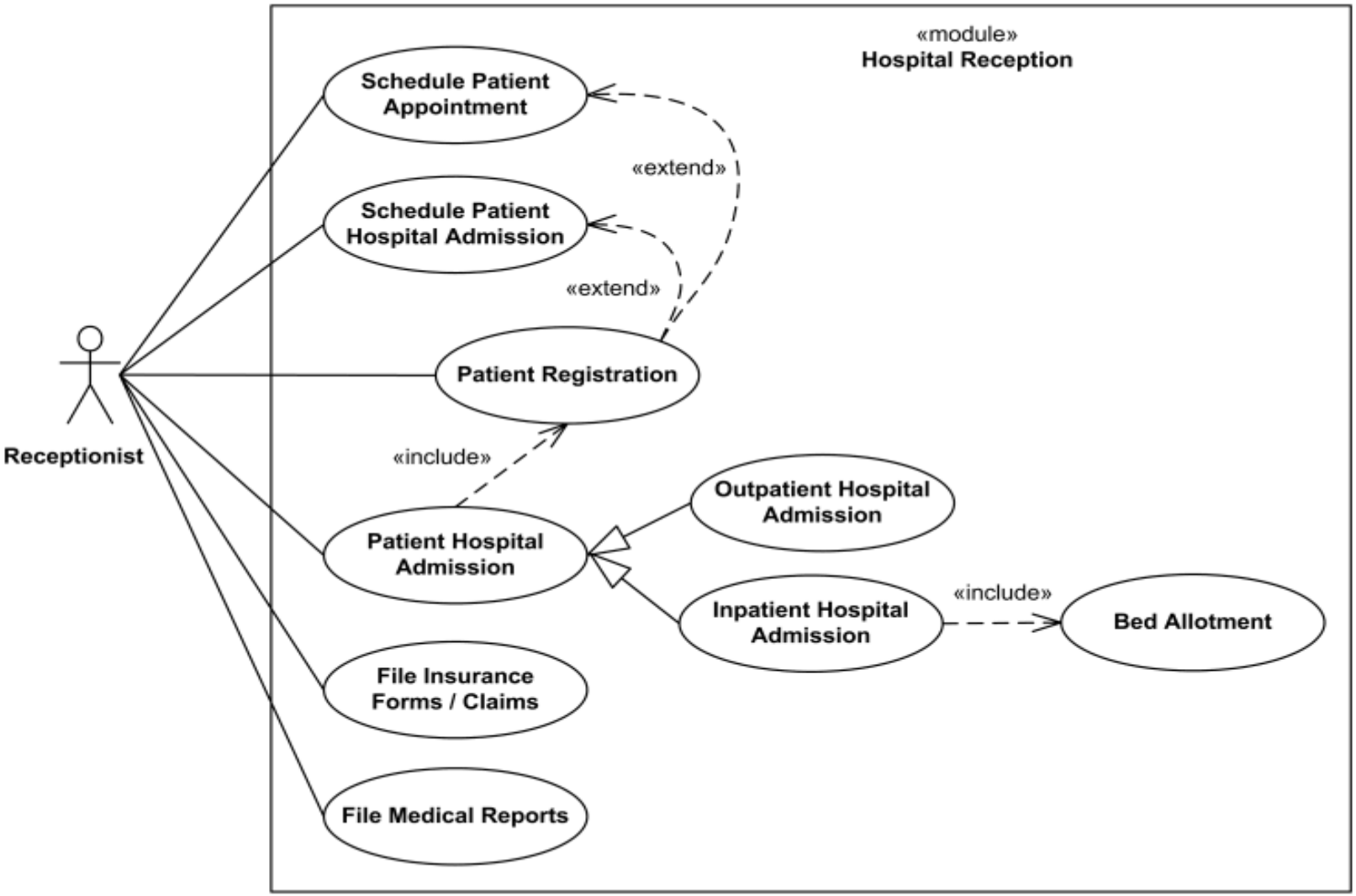
- The standard UC include**s** the mandatory UC (use the verb to figure direction arrow).

- Standard use case can NOT execute without the include case → **tight coupling** .

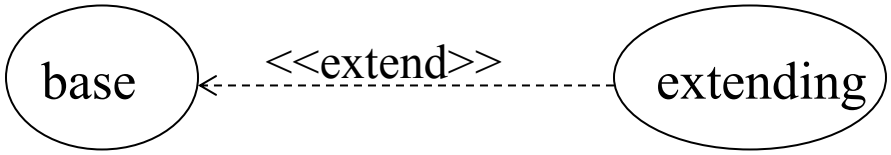
Reading use case diagram with Include relationship



Include Example



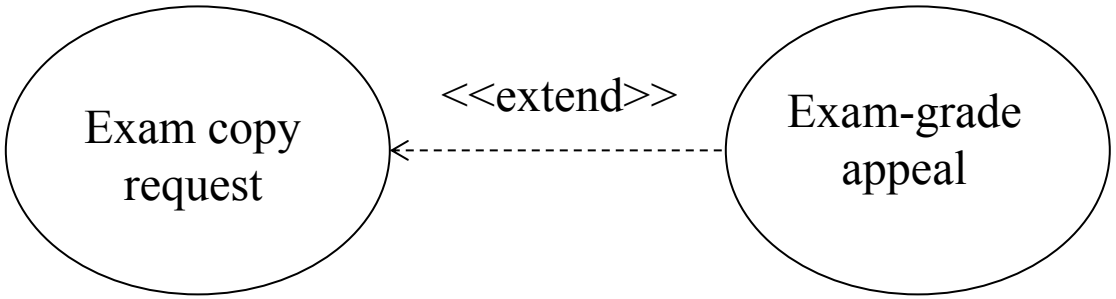
3. Extend



- The base use case implicitly incorporates the behavior of another use case at certain points called extension points.
- The base use case may stand alone, but under certain conditions its behavior may be extended by the behavior of another use case.

More about Extend

- Enables to model optional behavior or branching under conditions.



Extend relationship

- **Extend** relationship – linking an ***optional*** use case to a standard use case.

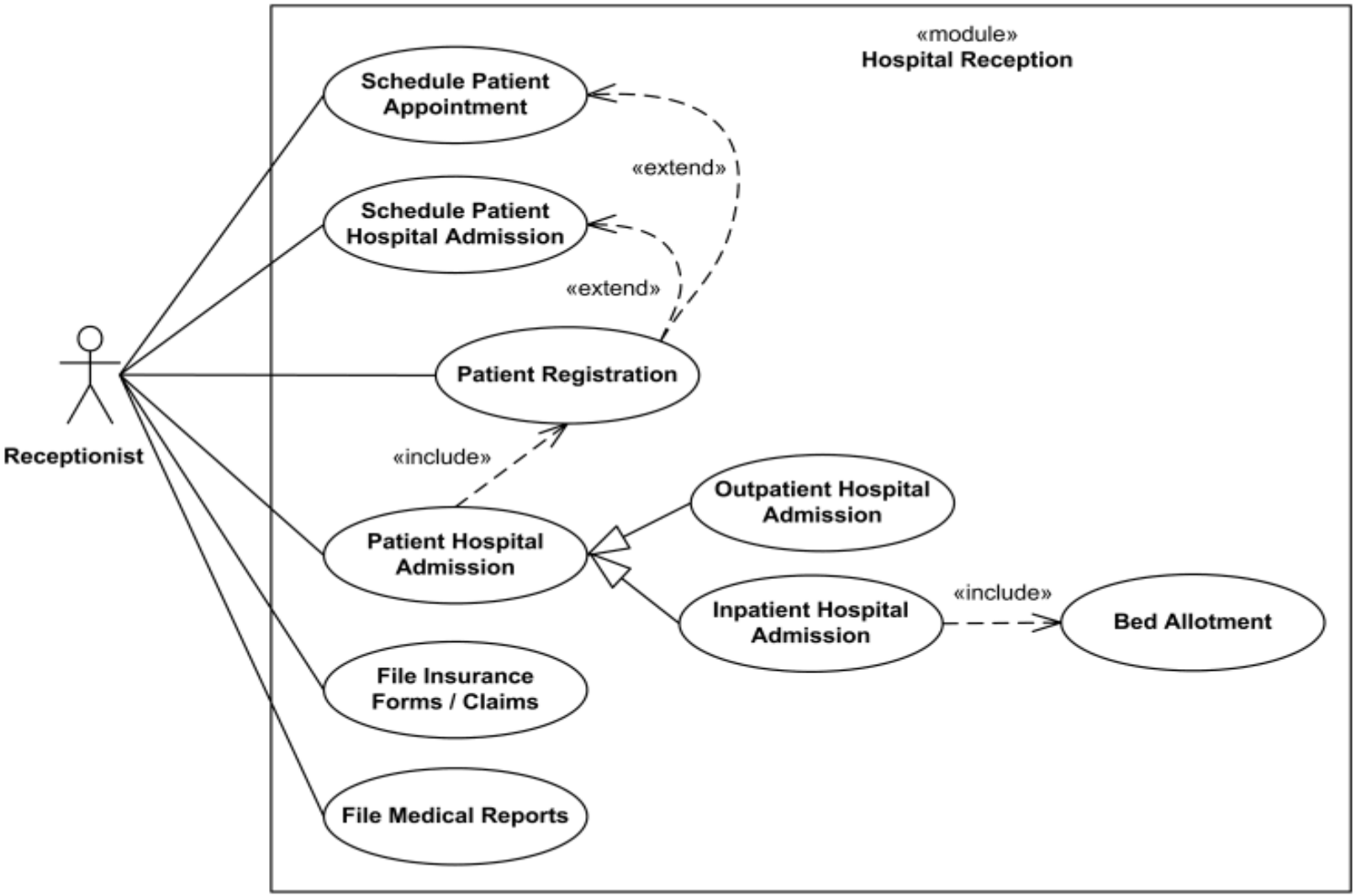
- Example: *Register Course* (standard use case) may have *Register for Special Class* (extend use case) – class for non-standard students, in unusual time, with special topics, requiring extra fees…).

- The optional UC extend**s** the standard UC

- Standard use case can execute without the extend case → loose coupling.

Reading extend relationship

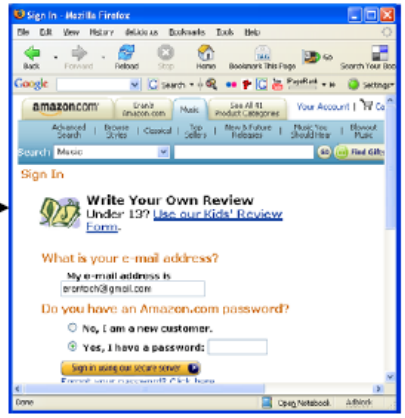
Extend Example #1



Extend Example #2

Product Page

Shopping Cart

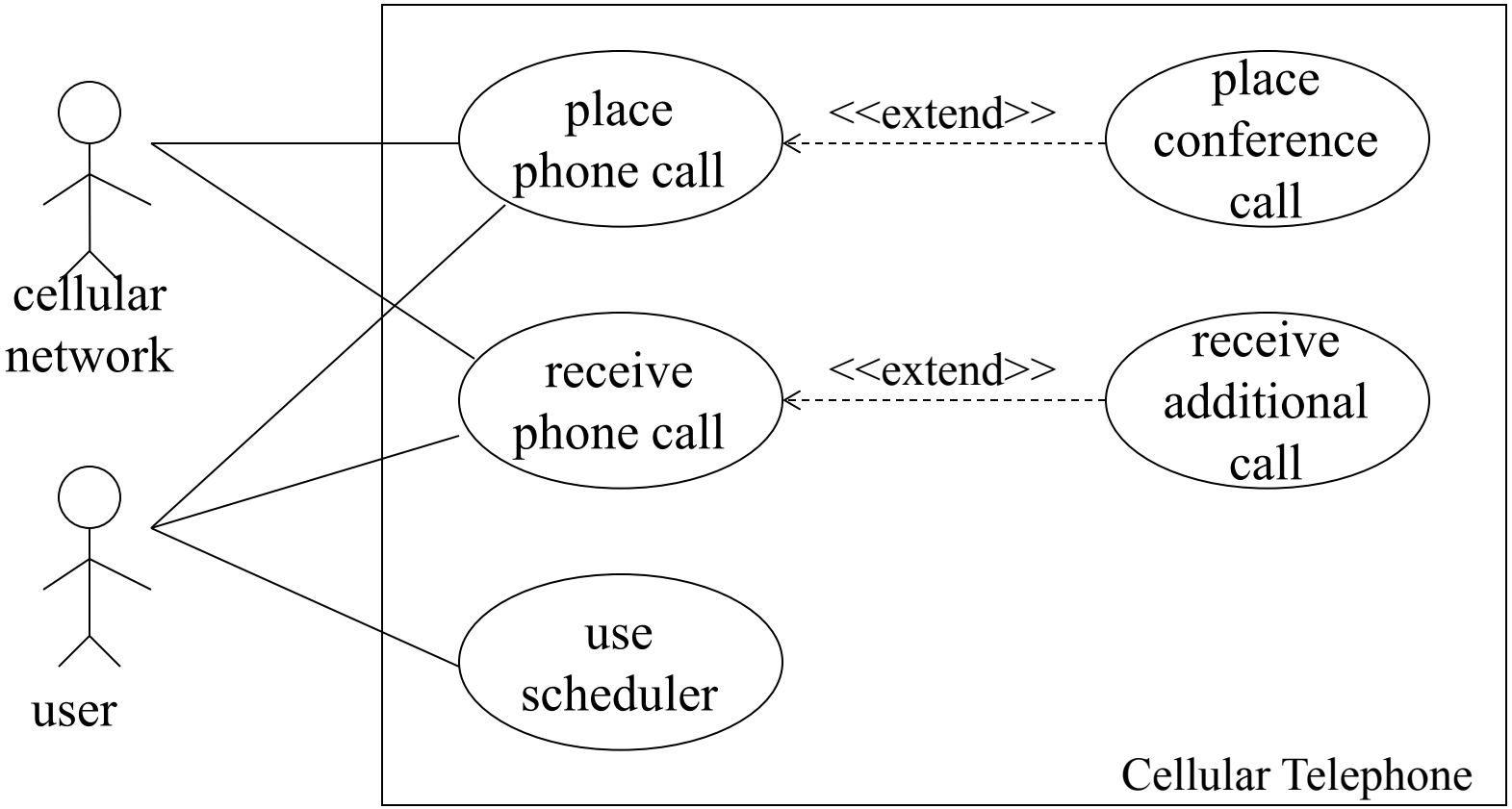


Review Writing

Relationships between Use Cases and Actors

Actors may be connected to use cases by associations, indicating that the actor and the use case communicate with one another using messages.

Example #1


cellular network
user
place phone call
receive phone call
use scheduler
<<extend>>
<<extend>>
place conference call
receive additional call
Cellular Telephone