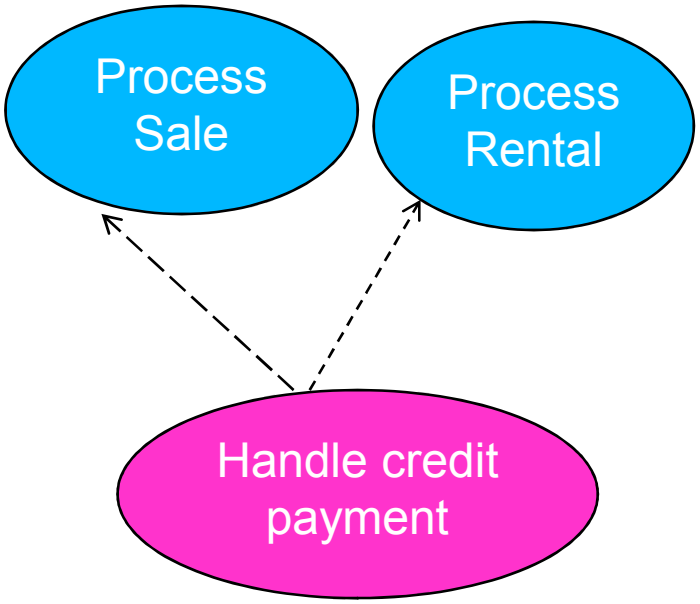
UNIT-III

RELATING USE CASE

MADHESWARI.K
AP/CSE
SSNCE

Relating Use cases

- Use cases can be related to each other
- **Example**



• **Sub-function use case "handle credit payment " may be part of several regular use cases, such as Process sale and Process Rental**
• **Organizing Use case into relationships has no impact on the behavior or requirements of the system**
•**Simply an organization mechanism**
- to improve communication and comprehension of the use cases
- reduce duplication of text
- and improve management of the use cases documents

Relating Use cases

- The use cases are related any other entity by using include, extend and generalization relationship.

1. Includes
2. Extends
3. Generalization

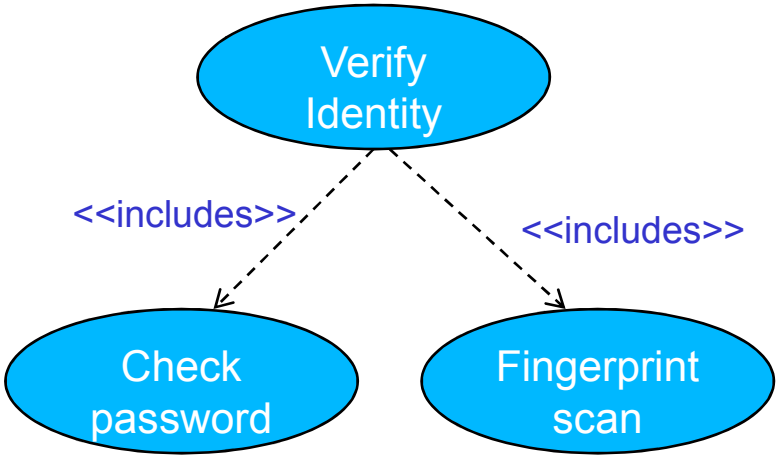
Include Relationship

- Most common and important relationship
- It is common to have some partial behavior that is common across several use cases.
- For example, the description of paying by credit occurs in several use cases, including *Process Sale, Process Rental, Contribute to Lay-away Plan,* and so forth.
- Rather than duplicate this text, it is desirable to separate it into its own sub-function use case, and indicate its inclusion.
- This is simply refactoring and linking text to avoid duplication.
- The include relationship can be used for most use case relationship problems.

Include Relationship

To summarize:

- Factor out sub function use cases and use the *Include* relationship when:
- They are duplicated in other use cases.
- A use case is *very* complex and long, and separating it into subunits aids comprehension.

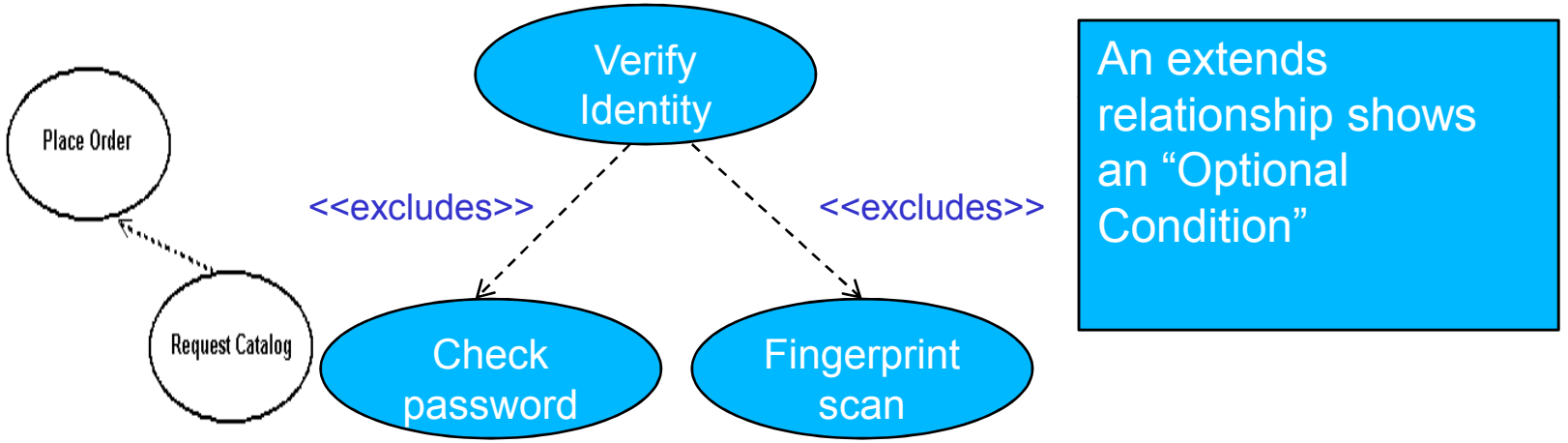


An Includes (or Uses) relationship shows an "AND" condition

Ex. BOTH a Password AND a Fingerprint Scan are required to verify identity

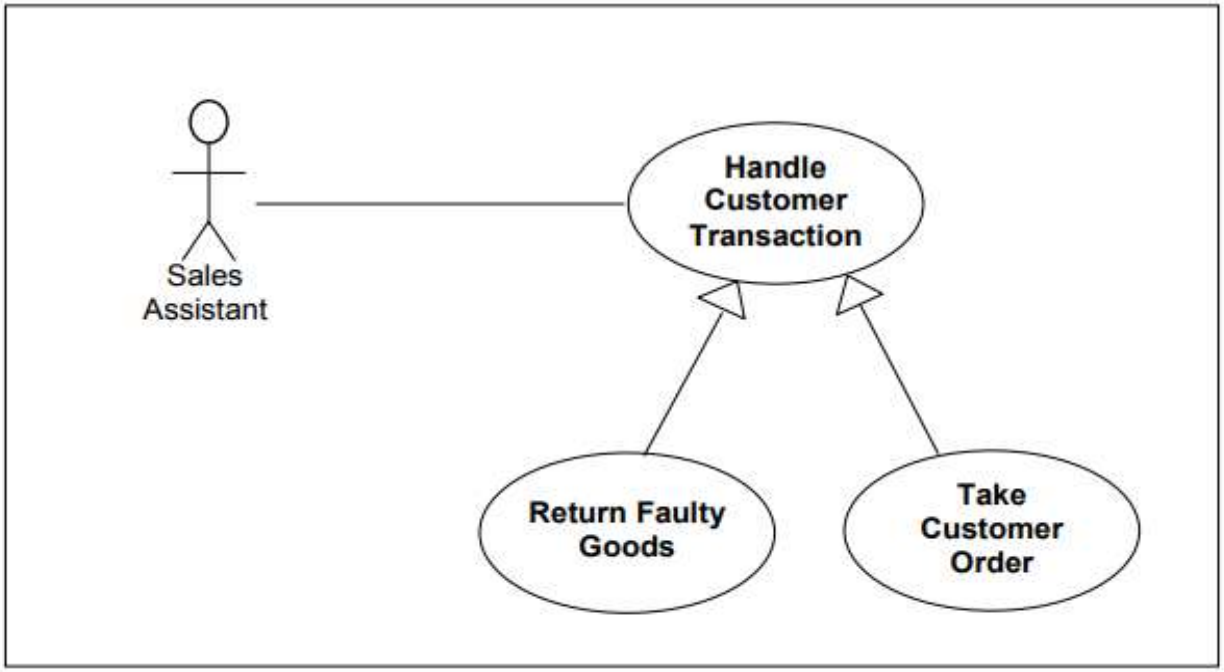
Excludes Relationship

- Extend puts additional behavior in a use case that does not know about it.
- It is shown as a dotted line with an arrow point and labeled <<extend>>
- In this case, a customer can request a catalog when placing an order



generalisation relationship

- A generalisation relationship between Use Cases indicates that the child Use Cases inherit the properties of the parent Use Case.
- Examples where this is a valuable aid to creating a useful Use Case model seem to be hard to find. We suspect that the generalisation relationship arises more from a theoretical consideration than any practical need
- The following example is provided only for completeness.



generalisation relationship

- In this example, we would be indicating that there are some common steps for all Use Cases that handle customer transactions and that the child Use Cases “Return Faulty Goods” and “Take Customer Order” have additional steps that fit into or around them. To be able to specify the additional steps for the child Use Cases you need to have extension points declared against the parent Use Case.

diagram illustrates use case Include relationship

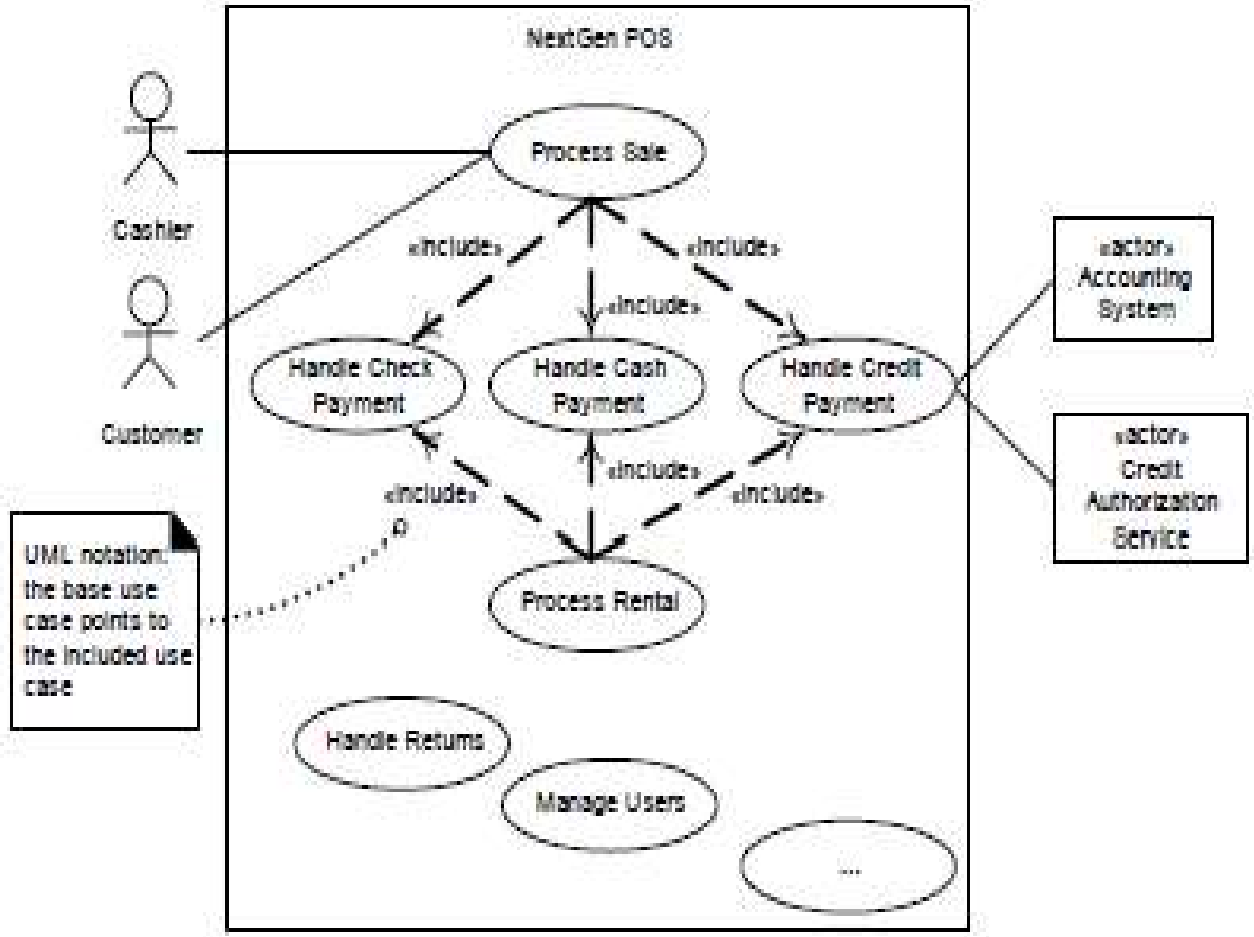


Figure 25.1 Use case include relationship in the Use-Case Model.

Example of Use Case Extend relationship