

EXTEND RELATIONSHIP REVISITED

Yesoda Bhargava



SOURCE OF CONFUSION

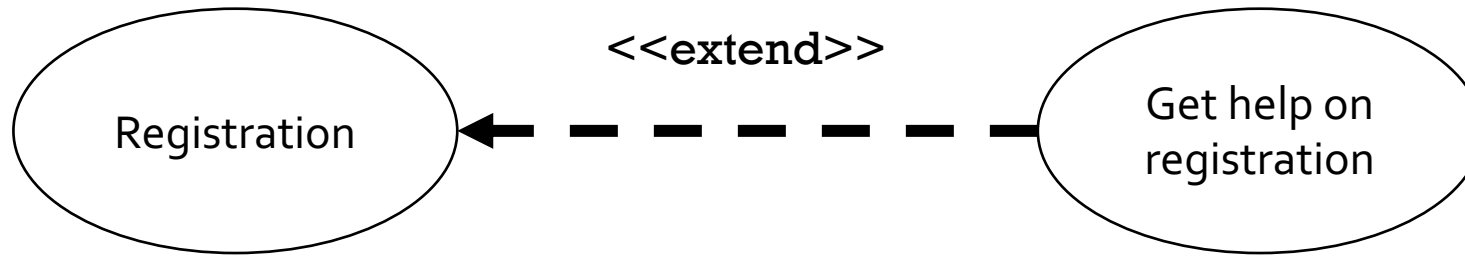
- Poor usage of words and use of extended and base use case, interchangeably across multiple sources on UML use case diagrams. Eg. “An extended use case only executes when the condition is true”. Herein, it should be the “extending use case” executes when the condition is true related to “extended use case”.
- In <<include>> relationship, the arrow head is pointing towards included use case from the including (base) use case.
- Extend relationship adds up behaviour sequence to the base use case. In <<extend>> relationship the direction of adding behaviour sequence to the base use case is opposite.
- In the include relationship, the base use case incorporates the included use case.
- Research has proposed a new construct <<trigger>> to specify how a base use case may call an operation use case.



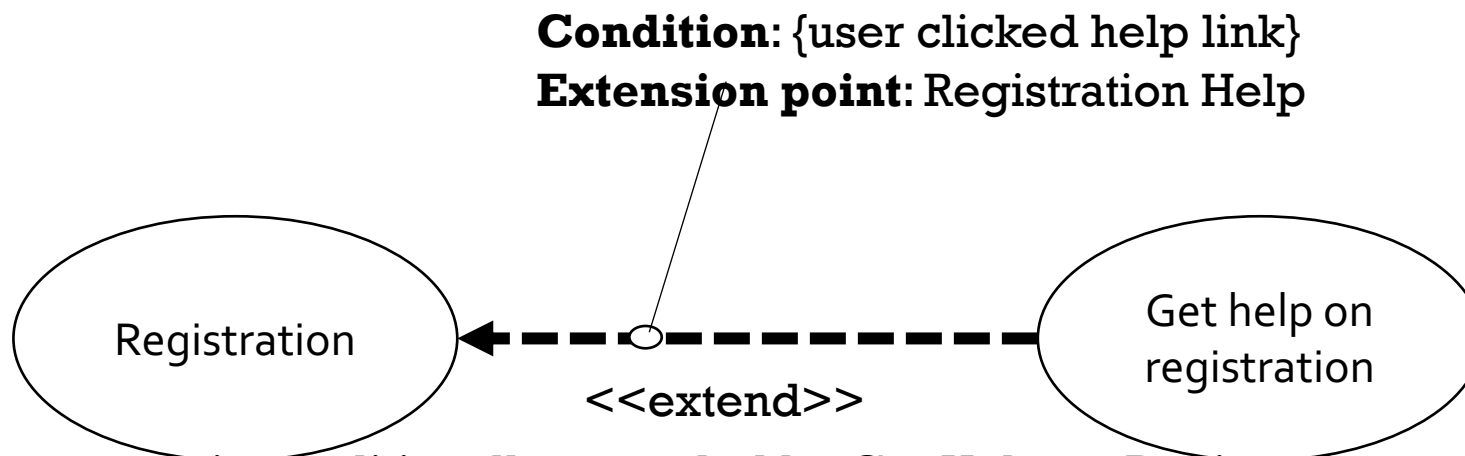
EXTEND RELATIONSHIP

- Indicate optional functionality or system behaviour.
- The <<extend>> relationship is used to include optional behaviour from an extending use case in an extended use case.
- Extend is a **directed relationship** that specifies how and when the behaviour defined in optional **extending use case** can be inserted into the behaviour defined in the **extended use case**.
- The extension takes place at extension points defined in the **extended use case**.





Registration use case could be extended with optional **Get Help On Registration** use case.

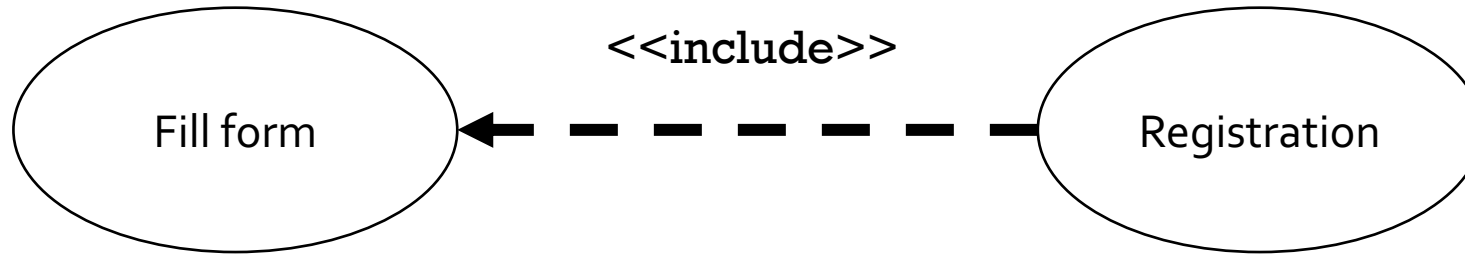


Registration use case is conditionally extended by Get Help on Registration use case in extension point, Registration Help.



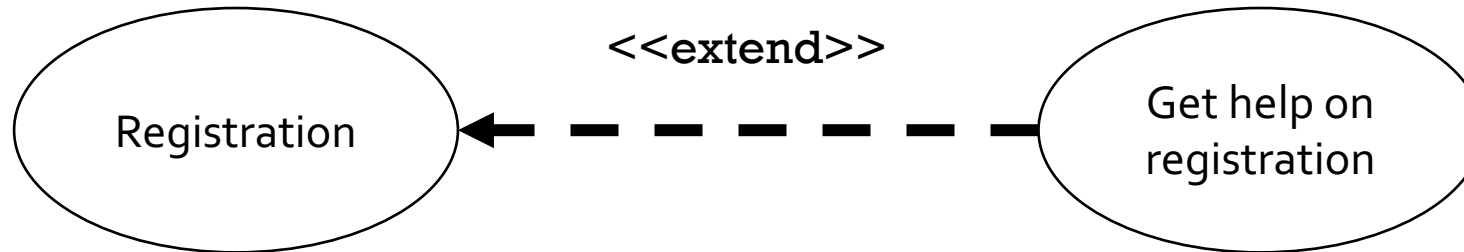
Included use case

Including (Base) use case



Extended (base) use case

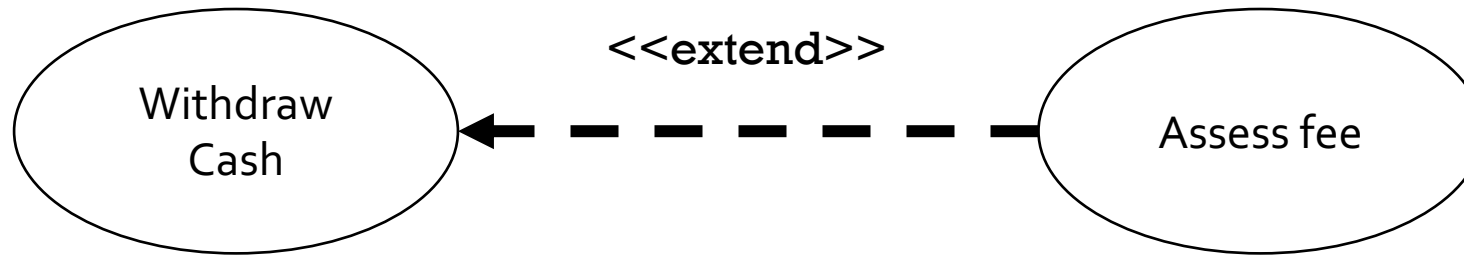
Extending use case



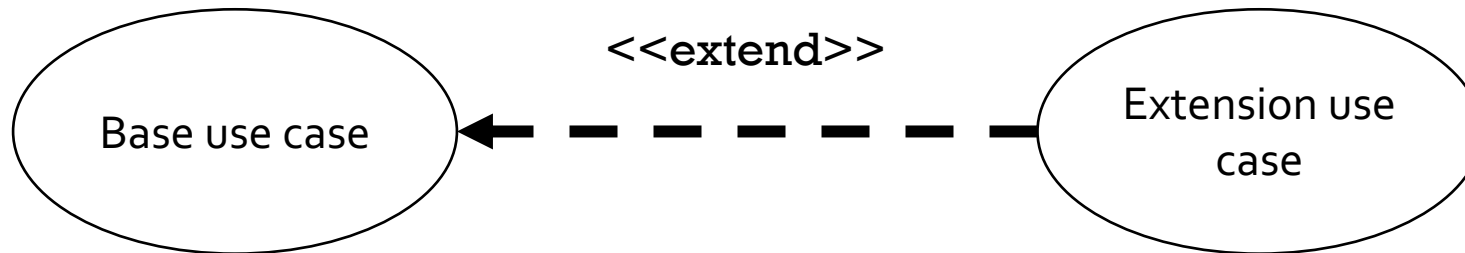
Meaningful on its own

Not meaningful on its own

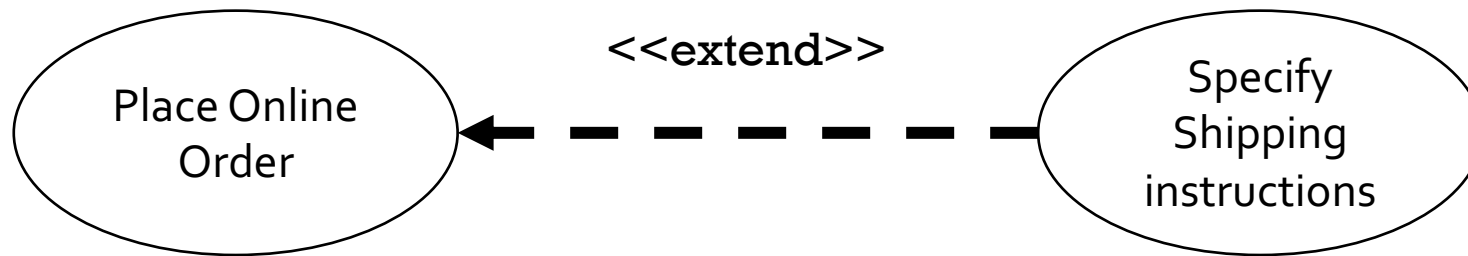




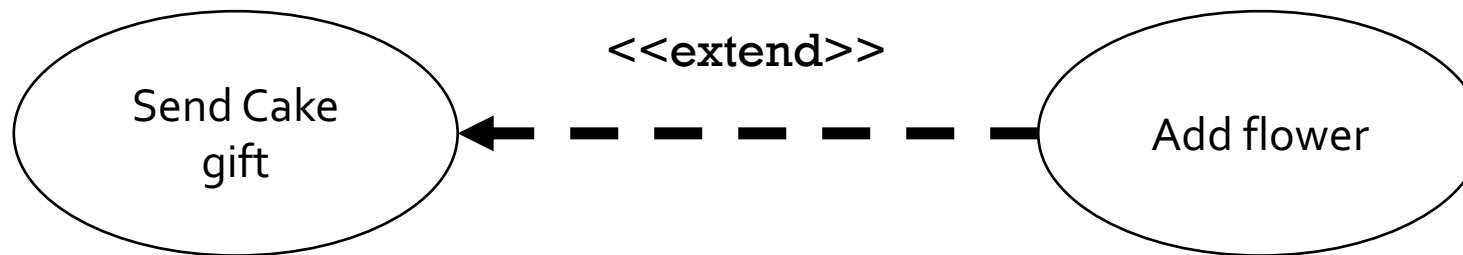
The extension use case consists of one or several behaviour sequences (segments) that can describe additional behaviour that can incrementally augment the behaviour of the base case. Each behaviour can be inserted into the base use case at a different point, called an Extension point.

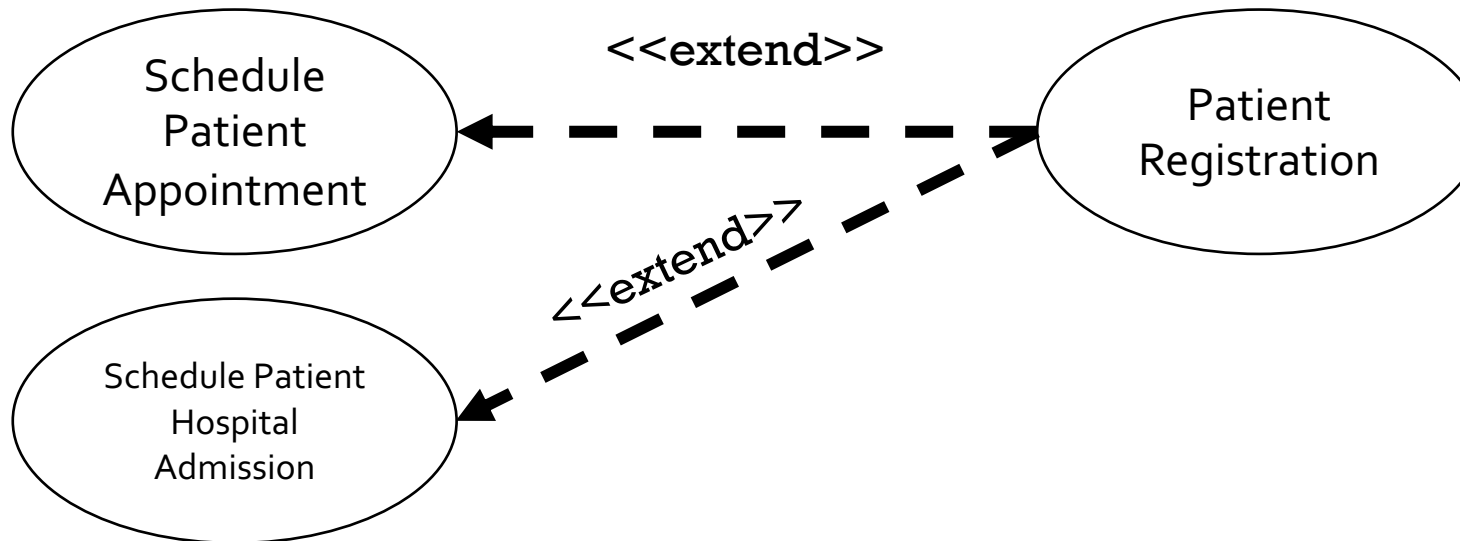
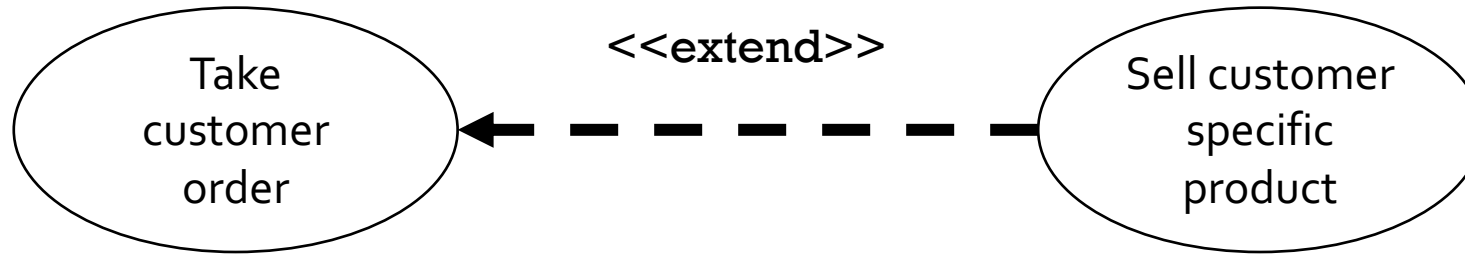
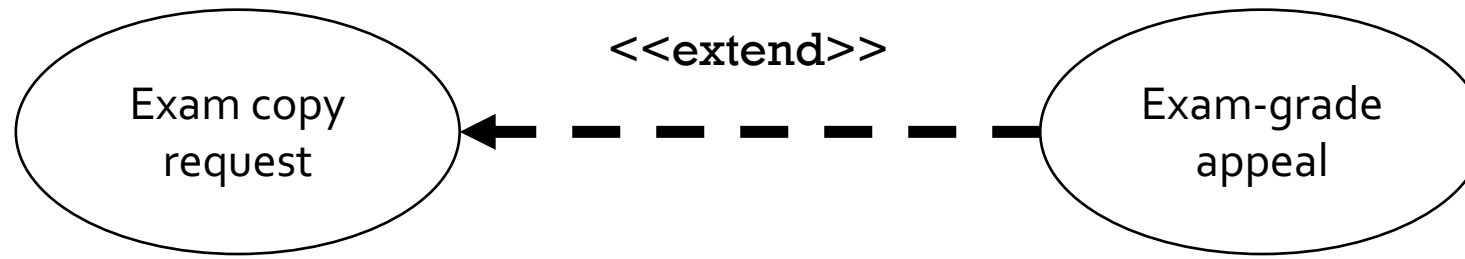


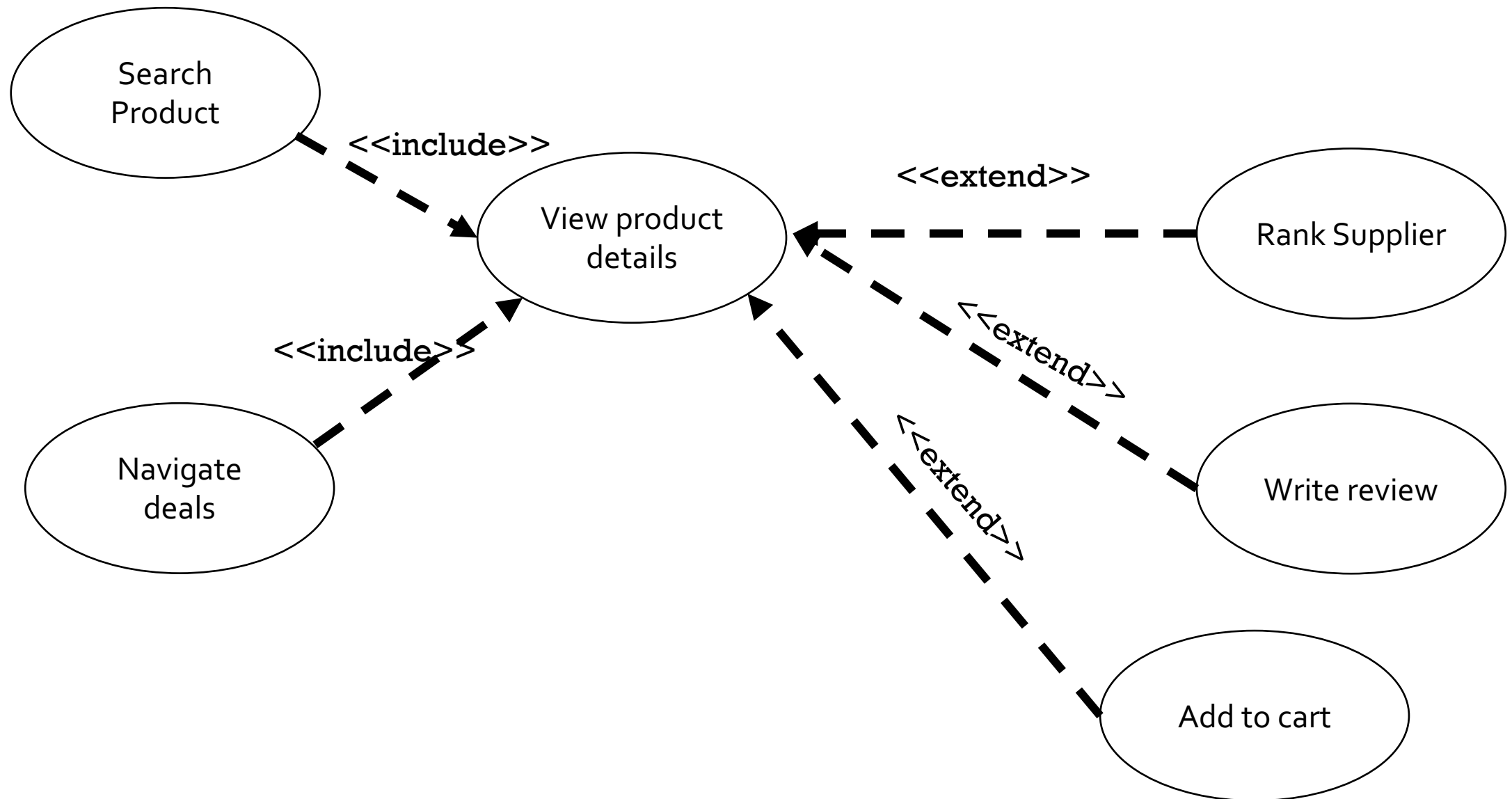
EXAMPLES FOR EXTEND RELATIONSHIP

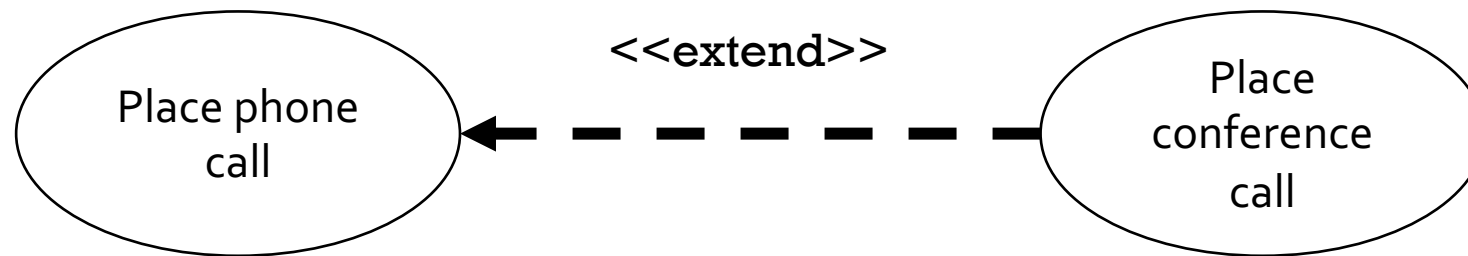
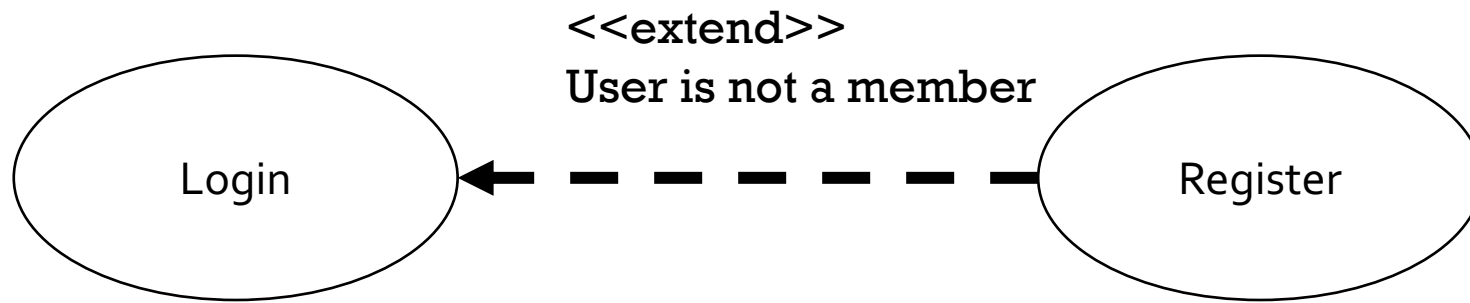


Behaviour in the Specify Shipping Instruction use cases are optional and only occur in certain circumstances.









SUMMARY

- Extend relationship allows us to modify the behaviour of base use case.
- The <<extend>> relationship says that we execute the base Use Case but when we get to a specified point in the flow, if the right conditions are met, we perform some different steps.
- The base use case implicitly incorporates the behaviour of another use case at certain points called **extension points**.
- The base use case may stand alone, but under certain conditions its behaviour may be extended by the behaviour of another use case.

