Jacobson

Would it be possible to express the extend relationship without having to bind to extension points or steps explicitly?

Cockburn

Use case modularization helps in coping with their complexity

Use cases restrict the user interface, but they are not its analogy

A use case diagram is appropriate as an overview, but it is not the main part of the use case model

Can extend be expressed as include and vice versa?

Jacobson

Is everything in order with UC Search Product?
Who is actually an e-shop customer?
Can a merchant be a customer?

A use case is a part of the user mental model with a better or worse support in the software system.

What's the meaning of a use case diagram?

Is there something in it not present in the text?

Lecture 2:

Modularizing Use Cases and Expressing Them in UML

Valentino Vranić

Ústav informatiky, informačných systémov
a softvérového inžinierstva

STU FIIT

vranic@stuba.sk fiit.sk/~vranic

MSOFT 2019/20
October 1, 2019

UC Place an Order

1. Customer selects to place an order.
2. System displays the search options.
3. Customer sets the search options and runs searching.
4. System displays the items that have been found.
5. Customer chooses among the items and confirms the choice.
6. System puts the selected items into the cart.
7. Customer can continue in selecting products – the use case continues with step 2.
8. Customer orders the products in the cart.
9. System requests the data necessary to place the order including the payment method.
10. Customer provides the necessary data.
11. Customer can give up the processes of product ordering at any time.
12. System records the order in a list of orders to be dispatched.
13. For each product in the order, System checks the available quantity.
14. If the quantity is below the limit, System adds the quantity under demand to the restock plan.
15. The use case ends.

Preconditions: Customer is logged on
Postconditions:

- *Minimal:* products that have been a part of the order remain there
- *Success:* products ordered by Customer is a part of the order

UC Place an Order

1. Customer selects to place an order.
2. System displays the search options.
3. Customer sets the search options and runs searching.
4. System displays the items that have been found.
5. Customer chooses among the items and confirms the choice.
6. System puts the selected items into the cart.
7. Customer can continue in selecting products – the use case continues with step 2.
8. Customer orders the products in the cart.
9. System requests the data necessary to place the order including the payment method.
10. Customer provides the necessary data.
11. Customer can give up the processes of product ordering at any time.
12. System records the order in a list of orders to be dispatched.
13. For each product in the order, System checks the available quantity.
14. If the quantity is below the limit, System adds the quantity under demand to the restock plan.
15. The use case ends.

UC Place an Order

Basic Flow: Place an Order

1. Customer selects to place an order.
2. The Search Products auxiliary flow is activated.
3. System puts the selected products into the cart.
4. Customer can continue in selecting products – the use case continues with step 2.
5. Customer orders the products in the cart.
6. System requests the data necessary to place the order including the payment method.
7. Customer provides the necessary data.
8. Customer can give up the processes of product ordering at any time.
9. System records the order in a list of orders to be dispatched.
10. If after dispatching the order the quantity of any product would fall below the prescribed limit, System adds a record about the need to increase the quantity of the corresponding product to the restock plan.
11. The use case ends.

Auxiliary Flow: Search Products

1. System displays the search options.
2. Customer sets the search options and runs searching.
3. System displays the items that have been found.

UC Place an Order

Basic Flow: Place an Order

1. Customer selects to place an order.
2. UC Search Products, its Search Products auxiliary flow, is activated.
3. System puts the selected products into the cart.
4. Customer can continue in selecting products – the use case continues with step 2.
5. Customer orders the products in the cart.
6. System requests the data necessary to place the order including the payment method.
7. Customer provides the necessary data.
8. Customer can give up the processes of product ordering at any time.
9. System records the order in a list of orders to be dispatched.
10. If after dispatching the order the quantity of any product would fall below the prescribed limit, System adds a record about the need to increase the quantity of the corresponding product to the restock plan.
11. The use case ends.

UC Search Products

Auxiliary Flow: Search Products

1. System displays the search options.
2. Customer sets the search options and runs searching.
3. System displays the items that have been found.

Jacobson

UC Place an Order

Basic Flow: Place an Order

1. Customer selects to place an order.
2. UC Search Products, its Search Products auxiliary flow, is activated.
3. System puts the selected products into the cart.
4. Customer can continue in selecting products – the use case continues with step 2.
5. Customer orders the products in the cart.
6. System requests the data necessary to place the order including the payment method.
7. Customer provides the necessary data.
8. Customer can give up the processes of product ordering at any time.
9. System records the order in a list of orders to be dispatched.
10. If after dispatching the order the quantity of any product would fall below the prescribed limit, System adds a record about the need to increase the quantity of the corresponding product to the restock plan.
11. The use case ends.

UC Search Products

Auxiliary Flow: Search Products

1. System displays the search options.
2. Customer sets the search options and runs searching.
3. System displays the items that have been found.

```
public class Ordering {
    ...
    public void order(Product product, int quantity) {
        ...
        new SearchProducts().find(product);
        ...
        if (getQuantity(product) >= quantity) {
            ...
        } else...
    }
    ...
}
```

Jacobson

UC Place an Order

Basic Flow: Place an Order

1. Customer selects to place an order.
2. UC Search Products, its Search Products auxiliary flow, is activated.
3. System puts the selected products into the cart.
4. Customer can continue in selecting products – the use case continues with step 2.
5. Customer orders the products in the cart.
6. System requests the data necessary to place the order including the payment method.
7. Customer provides the necessary data.
8. Customer can give up the processes of product ordering at any time.
9. System records the order in a list of orders to be dispatched.
10. The use case ends.

Alternative Flow: Modify the Restock Plan

In step 9 of the Place an Order basic flow, if after dispatching the order the quantity of any product would fall below the prescribed limit:

1. System adds a record to the restock plan about the need to increase the quantity of each such product in the order whose quantity would after dispatching the order fall below the prescribed limit.

step 2.
5. Customer orders the products in the cart.
6. System requests the data necessary to place the order including the payment method.
7. Customer provides the necessary data.
8. Customer can give up the processes of product ordering at any time.
9. System records the order in a list of orders to be dispatched.
10. The use case ends.

Alternative Flow: Modify the Restock Plan

In step 9 of the Place an Order basic flow, if after dispatching the order the quantity of any product would fall below the prescribed limit:

1. System adds a record to the restock plan about the need to increase the quantity of each such product in the order whose quantity would after dispatching the order fall below the prescribed limit.

UC Place an Order

Basic Flow: Place an Order

1. Customer selects to place an order.
2. UC Search Products, its Search Products auxiliary flow, is activated.
3. System puts the selected products into the cart.
4. Customer can continue in selecting products – the use case continues with step 2.
5. Customer orders the products in the cart.
6. System requests the data necessary to place the order including the payment method.
7. Customer provides the necessary data.
8. Customer can give up the processes of product ordering at any time.
9. System records the order in a list of orders to be dispatched.
10. The use case ends.

Extension Points:
• *Saving the Order*: step 9

UC Modify the Restock Plan

Alternative Flow: Modify the Restock Plan

In the Saving the Order extension point in UC Place an Order, if after dispatching the order the quantity of any product would fall below the prescribed limit:

1. System adds a record to the restock plan about the need to increase the quantity of each such product in the order whose quantity would after dispatching the order fall below the prescribed limit.

8. Customer can give up the processes of product ordering at any time.
9. System records the order in a list of orders to be dispatched.
10. The use case ends.

Extension Points:
• *Saving the Order*: step 9

UC Modify the Restock Plan

Alternative Flow: Modify the Restock Plan

In the Saving the Order extension point in UC Place an Order, if after dispatching the order the quantity of any product would fall below the prescribed limit:

1. System adds a record to the restock plan about the need to increase the quantity of each such product in the order whose quantity would after dispatching the order fall below the prescribed limit.

UC Place an Order

Basic Flow: Place an Order

1. Customer selects to place an order.
2. UC Search Products, its Search Products auxiliary flow, is activated.
3. System puts the selected products into the cart.
4. Customer can continue in selecting products – the use case continues with step 2.
5. Customer orders the products in the cart.
6. System requests the data necessary to place the order including the payment method.
7. Customer provides the necessary data.
8. Customer can give up the processes of product ordering at any time.
9. System records the order in a list of orders to be dispatched.
10. The use case ends.

Extension Points:
• *Saving the Order*: step 9

UC Modify the Restock Plan

Alternative Flow: Modify the Restock Plan

In the Saving the Order extension point in UC Place an Order, if after dispatching the order the quantity of any product would fall below the prescribed limit:

1. System adds a record to the restock plan about the need to increase the quantity of each such product in the order whose quantity would after dispatching the order fall below the prescribed limit.

```
public class Ordering {
    ...
    public void order(Product product, int quantity) {
        ...
        if (...) {
            saveOrder();
            for (Product product : orderItems) {
                if (getQuantity(product) <
                    product.minimumQuantity()) {
                    // add a record to the restock plan
                    ...
                }
        } else...
    }
    ...
}
```

```
public aspect RestockPlan {
    ...
    void around(Ordering ordering):
        call(* Ordering.saveOrder()
        && this(ordering) {
        for (Product product : orderItems) {
            if (getQuantity(product) <
                product.minimumQuantity()) {
                // add a record to the restock plan
                ...
                }
            } else...
    }
    ...
}
```

Would it be possible to express the extend relationship without having to bind to extension points or steps explicitely?

UC Modify the Restock Plan

Cockburn

Alternative Flow: Modify the Restock Plan

Whenever an order is being saved in the system, if after dispatching the order the quantity of any product would fall below the prescribed limit:

1. System adds a record to the restock plan about the need to increase the quantity of each such product in the order whose quantity would after dispatching the order fall below the prescribed limit.

UC Modify the Restock Plan

Alternative Flow: Modify the Restock Plan

Whenever an order is being saved in the system, if after dispatching the order the quantity of any product would fall below the prescribed limit:

1. System adds a record to the restock plan about the need to increase the quantity of each such product in the order whose quantity would after dispatching the order fall below the prescribed limit.

Cockburn

UC Modify the Restock Plan

Alternative Flow: Modify the Restock Plan

Whenever an order is being saved in UC Place an Order, if after dispatching the order the quantity of any product would fall below the prescribed limit:

1. System adds a record to the restock plan about the need to increase the quantity of each such product in the order whose quantity would after dispatching the order fall below the prescribed limit.

Can extend
be expressed as
include and vice
versa?

UC Place an Order

Basic Flow: *Place an Order*

1. Customer selects to place an order.
2. UC Search Products, its Search Products auxiliary flow, is activated.
3. System puts the selected products into the cart.
4. Customer can continue in selecting products – the use case continues with step 2.
5. Customer orders the products in the cart.
6. System requests the data necessary to place the order including the payment method.
7. Customer provides the necessary data.
8. Customer can give up the processes of product ordering at any time.
9. System records the order in a list of orders to be dispatched.
10. UC Modify the Restock Plan, its Modify the Restock Plan auxiliary flow, is activated.
11. The use case ends.

UC Modify the Restock Plan

Auxiliary Flow: Modify the Restock Plan

1. If after dispatching the order the quantity of any product would fall below the prescribed limit, System adds a record to the restock plan about the need to increase the quantity of each such product in the order whose quantity would after dispatching the order fall below the prescribed limit.

5. Customer orders the products in the cart.
6. System requests the data necessary to place the order including the payment method.
7. Customer provides the necessary data.
8. Customer can give up the processes of product ordering at any time.
9. System records the order in a list of orders to be dispatched.
10. UC Modify the Restock Plan, its Modify the Restock Plan auxiliary flow, is activated.
11. The use case ends.

UC Modify the Restock Plan

Auxiliary Flow: Modify the Restock Plan

1. If after dispatching the order the quantity of any product would fall below the prescribed limit, System adds a record to the restock plan about the need to increase the quantity of each such product in the order whose quantity would after dispatching the order fall below the prescribed limit.

UC Place a Quick Order

The use case specializes UC Place an Order. The steps are overriden as follows:

1. Customer selects to place a quick order.
2. Customer provides directly a product code.
3. ...
4.

UC Manage a Product (CRUD)

Basic Flow: *Create a Product*

1. Merchant selects to create a product.
1. System requests the product data.
2.Merchant provides the name, type, and picture of the product.
3. System adjusts the size of the product to the standard size.
4. Merchant sets the product category.
5. If Merchant confirms the provided data, System saves them.
6. The use case ends.

Basic Flow: Display *a Product*

1. Merchant selects to display an existing product.
2. UC Search Products (its equally named flow) is activated.
3. System displays the selected product.
4. Merchant observes the product.
5. The use case ends.

UC Manage a Product (CRUD)

Basic Flow: *Create a Product*

1. Merchant selects to create a product.
1. System requests the product data.
2.Merchant provides the name, type, and picture of the product.
3. System adjusts the size of the product to the standard size.
4. Merchant sets the product category.
5. If Merchant confirms the provided data, System saves them.
6. The use case ends.

Basic Flow: Display *a Product*

1. Merchant selects to display an existing product.
2. UC Search Products (its equally named flow) is activated.
3. System displays the selected product.
4. Merchant observes the product.
5. The use case ends.

Basic Flow: *Modify a Product*

1. Merchant selects to modify an existing product.
2. UC Search Products (its equally named flow) is activated.
3. System opens the selected product and enables its modification.
4. Merchant modifies the product data.
5. If Merchant confirms the changes, System saves them.
6. The use case ends.

Basic Flow: *Remove a Product*

1. Merchant selects to remove a product.
2. UC Search Products (its equally named flow) is activated.
3. If Merchant confirms the product removal, System sets the removed flag on this product.
4. The use case ends.

Jacobson

Use case modularization helps in coping with their complexity

UC Manage a Product (CRUD)

...

Basic Flow: Modify a Product

1. Merchant selects to modify an existing product.
2. UC Search Products (its equally named flow) is activated.
3. System opens the selected product and enables its modification.
4. Merchant modifies the product data.
5. If Merchant confirms the changes, System saves them.
6. The use case ends.

...

UC Manage a Product (CRUD)

...

Basic Flow: Modify a Product

1. Merchant selects to modify an existing product.
2. UC Search Products (its equally named flow) is activated.
3. System opens the selected product and enables its modification.
4. Merchant modifies the product data.
5. If Merchant confirms the changes, System saves them.
6. The use case ends.

...

Is everything in order with UC Search Product?

Who is actually an e-shop customer?

Can a merchant be a customer?

UC Search Products

Auxiliary Flow: Search Products

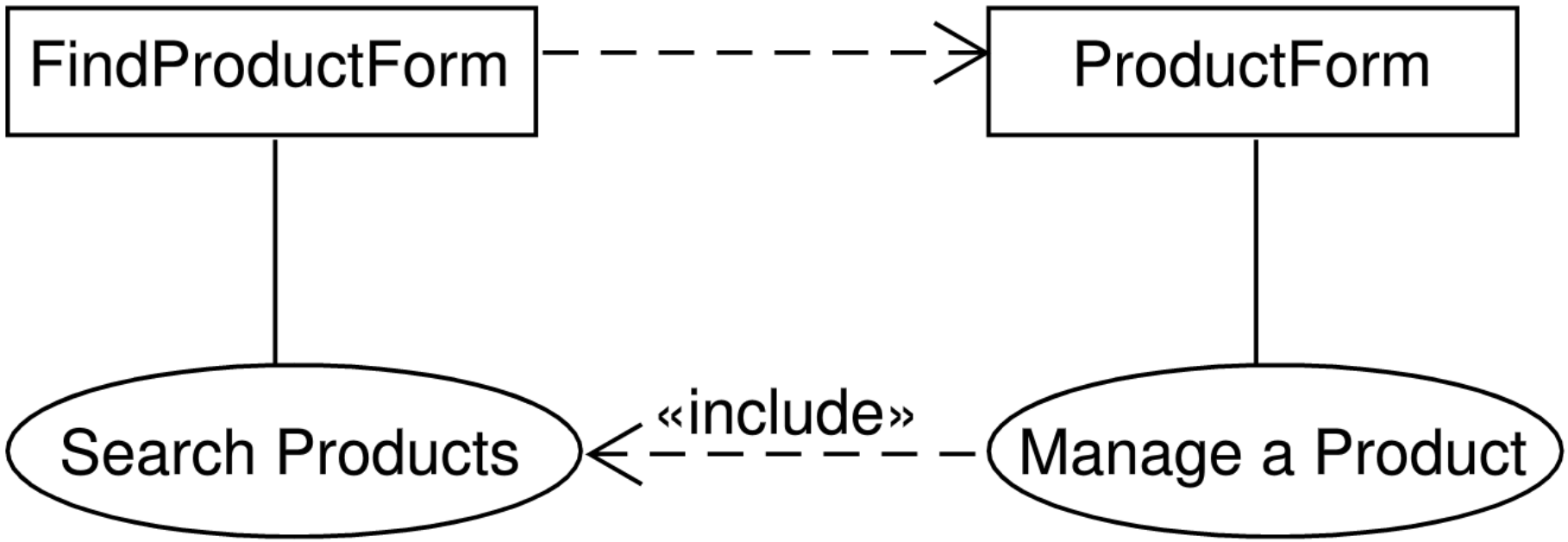
1. System displays the search options.
2. Customer sets the search options and runs searchin
3. System displays the items that have been found.

Customer and Merchant are (specific) users

A user maintains orders in Excel with a support at the level of summary calculations etc.

Can in such a case UC Record an Order be observed?

A use case is a part of the user mental model with a better or worse support in the software system.


FindProductForm
ProductForm
Search Products
«include»
Manage a Product

Use cases restrict the user interface, but they are not its analogy

What's the meaning of a use case diagram?

Is there something in it not present in the text?

A use case diagram is appropriate as an overview, but it is not the mainpart of the use case model

Use cases can be relatively straightforwardly expressed using activity diagrams, but only sequence diagrams help in determining the structure of the system becuase they express use cases as object collaborations

Use case modularization helps in coping with their complexity

Use cases restrict the user interface, but they are not its analogy

A use case diagram is appropriate as an overview, but it is not the mainpart of the use case model

Use cases can be relatively straightforwardly expressed using activity diagrams, but only sequence diagrams help in determining the structure of the system becuase they express use cases as object collaborations