Software Engineering

Project Preparation

Use Case Modeling

[Based on Chapters 3 & 4, Arlow and Neustadt, "UML and the Unified Process," Addison-Wesley, 2002]

Outline

- Defining requirements
- Use case modeling
  - Overview
  - Finding actors and use cases
  - Detailing use cases
  - Scenarios

Defining Requirements

- *Requirement*: "a specification of what should be implemented"
- There are two types of requirements:
  - *Functional requirements*: describe the desired behaviour of the system
  - *Non-functional requirements*: specify particular properties of or constraints on the system
- In theory, the set of requirements should be only about "what" the system should do, but in practice it is not possible to avoid "how" aspects of the system

.Defining Requirements.

- The SRS (*Systems Requirements Specification*) is the document that contains the set of requirements expected to be satisfied by the system, both functional and non-functional
- There are many ways to write a SRS ("company dependent" ways)
- The SRS provides the input for the analysis and design phases of the development process
- The bottom line regarding the SRS is: "does it help me to understand what the system should do or not?"

..Defining Requirements.

- Simple format recommended for well-formed requirements:

  <id> The <system> shall <function>

- Examples of functional requirements (what the system should do):

  1 The ATM shall check the validity of the ATM card inserted.
  2 The ATM shall validate the PIN number entered by the client.

- Examples of non-functional requirements (constraints on or properties of the system):

  1 The ATM shall be written in C++.
  2 The ATM shall validate the PIN in three seconds or less.

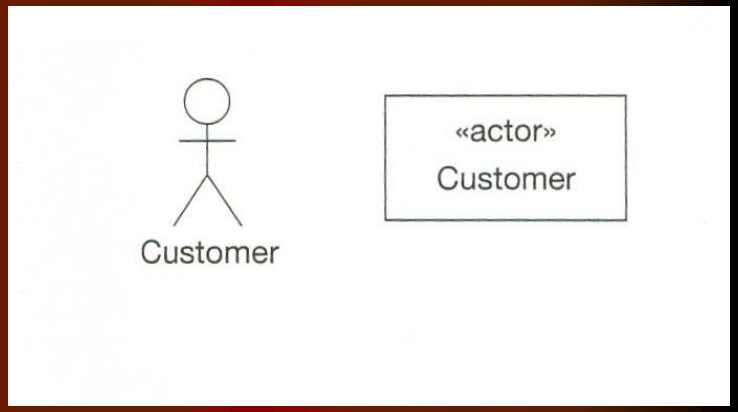
Use Case Modeling: Overview

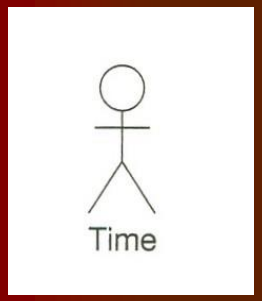
- The Use Case Model consists of the following:
  - Actors
  - Use cases
  - Relationships
  - System boundary
- Steps of use case modeling:
  - Find the system boundary
  - Find the actors
  - Find the use cases:
    - Specify the use cases
    - Create scenarios

Finding Actors and Use Cases....

- An *actor* is a role taken by an external entity when interacting with the system directly
- An actor is a stereotype of class with its own icon

  Fig. 4.3 and 4.4 [Arlow & Neustadt 2002]



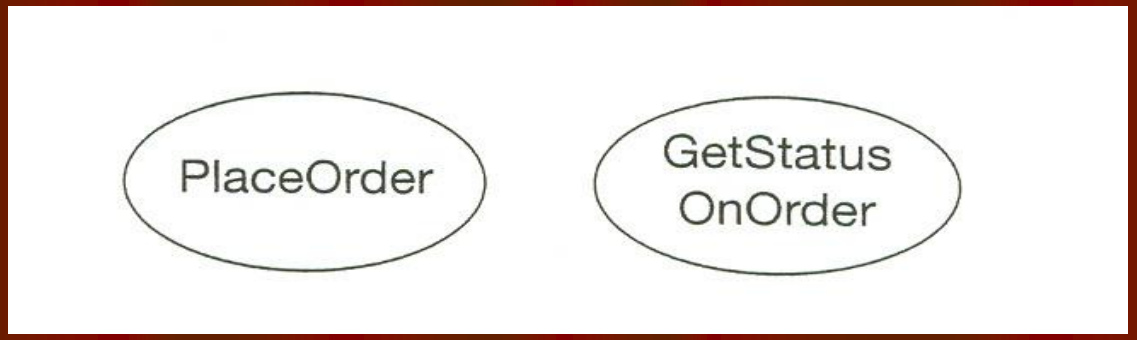


.Finding Actors and Use Cases...

- An actor:
  - Is always external to the system
  - Interacts directly with the system
  - Represents a role played by people of things, not specific people or specific things
- According to Rumbaugh, a *use case* is “a specification of sequences of actions, including variant sequences and error sequences, that a system, subsystem, or class can perform by interacting with outside actors”
- Use cases:
  - Are always started by an actor
  - Are always written from an actor’s point of view

..Finding Actors and Use Cases..

- Examples of use cases, Fig. 4.5 [Arlow & Neustadt 2002]



- Names of use cases should be verb phrases
- Candidate use cases can be discovered starting from the list of actors (how they interact with the system?)
- Finding use cases is an iterative process

...Finding Actors and Use Cases.

- The use case diagram shows the system boundary, the use cases internal to the system, and the actors external to the system, e.g. [Fig.4.6, Arlow and Neustadt, 2002]

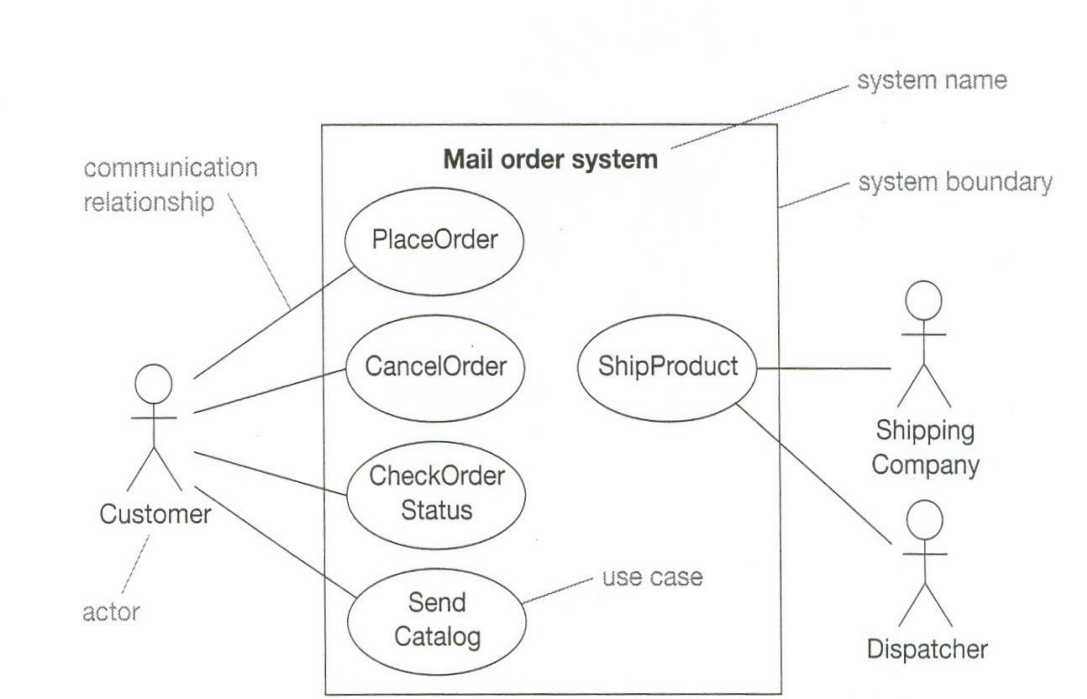


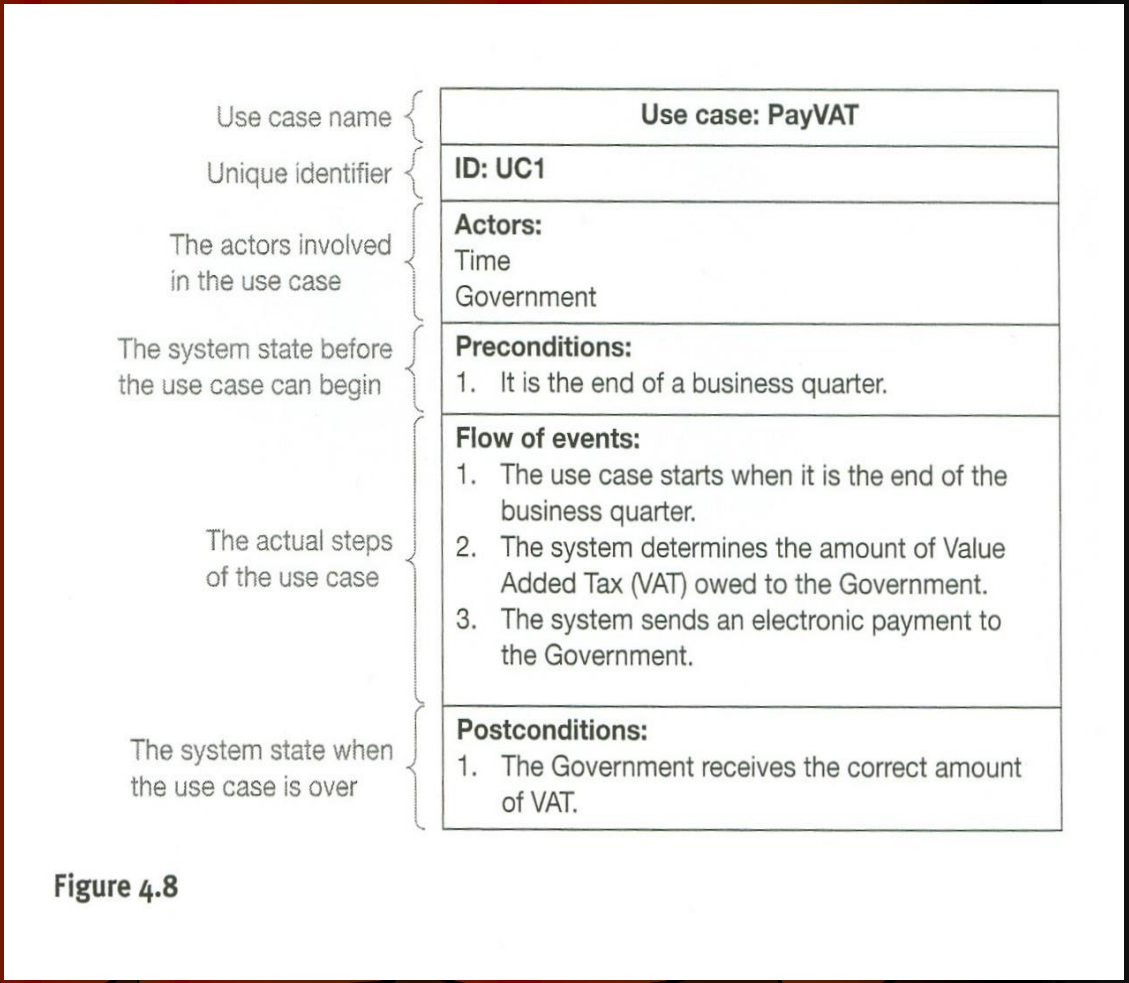
Figure 4.6

....Finding Actors and Use Cases

- The *project glossary*
  - Important project artefact
  - Provides a dictionary of key business terms
  - Captures business language and jargon
  - Should resolve synonyms and homonyms
  - Should be understandable by all stakeholders
  - UML does not set a standard for the project glossary
  - Synchronization between the project glossary and the UML model is needed

Detailing Use Cases….

- The output of this activity is a more detailed use case that consists at least of the use case name and use case specification.
- Most common template for *use case specification*, Fig. 4.8 [Arlow & Neustadt, 2002]



Use case name	**Use case: PayVAT**
Unique identifier	**ID: UC1**
The actors involved in the use case	**Actors:** Time Government
The system state before the use case can begin	**Preconditions:** 1. It is the end of a business quarter.
The actual steps of the use case	**Flow of events:** 1. The use case starts when it is the end of the business quarter. 2. The system determines the amount of Value Added Tax (VAT) owed to the Government. 3. The system sends an electronic payment to the Government.
The system state when the use case is over	**Postconditions:** 1. The Government receives the correct amount of VAT.

Figure 4.8

.Detailing Use Cases…

- Branching, repetition, and alternative flows are possible in a use case
- Example of branching using the keyword IF, Fig. 4.9 [Arlow and Neustadt, 2002]

Use case: ManageBasket
ID: UC10
Actors: Customer
Preconditions: 1. The shopping basket contents are visible.
Flow of events: 1. The use case starts when the Customer selects an item in the basket. 2. If the Customer selects "delete item" 2.1 The system removes the item from the basket. 3. If the Customer types in a new quantity 3.1 The system updates the quantity of the item in the basket.
Postconditions: 1. The basket contents have been updated.

..Detailing Use Cases..

- Example of alternative flows, Fig. 4.10 [Arlow and Neustadt, 2002]

Use case: DisplayBasket
ID: UC11
Actors: Customer
Preconditions: 1. The Customer is logged on the system.
Flow of events: 1. The use case starts when the Customer selects "display basket". 2. If there are no items in the basket 2.1 The system informs the Customer that there are no items in the basket yet. 2.2 The use case terminates. 3. The system displays a list of all items in the Customer's shopping basket including product ID, name, quantity and item price.
Postconditions:
Alternative flow 1: 1. At any time the Customer may leave the shopping basket screen.
Postconditions:
Alternative flow 2: 1. At any time the Customer may leave the system.
Postconditions:

...Detailing Use Cases.

- Example of repetition within a flow (FOR), Fig. 4.11 [Arlow and Neustadt, 2002]

Use case: FindProduct
ID: UC12
Actors: Customer
Preconditions:
Flow of events: 1. The Customer selects "find product". 2. The system asks the Customer for search criteria. 3. The Customer enters the requested criteria. 4. The system searches for products that match the Customer's criteria. 5. If the system finds some matching products then 5.1. For each product found 5.1.1. The system displays a thumbnail sketch of the product. 5.1.2. The system displays a summary of the product details. 5.1.3. The system displays the product price. 6. Else 6.1. The system tells the Customer that no matching products could be found.
Postconditions:
Alternative flow: 1. At any point the Customer may move to different page.
Postconditions:

....Detailing Use Cases

- Example of repetition within a flow (WHILE), Fig. 4.12 [Arlow and Neustadt, 2002]

Use case: ShowCompanyDetails
ID: UC13
Actors: Customer
Preconditions:
Flow of events: 1. The use case starts when the Customer selects "show company details". 2. The system displays a web page showing the company details. 3. While the Customer is browsing the company details 3.1. The system plays some background music. 3.2. The system displays special offers in a banner ad.
Postconditions:

Scenarios.

- Primary scenario of a use case, Fig. 4.13 [Arlow and Neustadt, 2002]
- Scenarios do not include IF, FOR, WHILE
- They are sort of “activity logs”

Use case: Checkout
ID: UC14
Actors: Customer
Preconditions:
Primary scenario: 1. The use case begins when the Customer selects “go to checkout”. 2. The system displays the customer order. 3. The system asks for the customer identifier. 4. The Customer enters a valid customer identifier. 5. The system retrieves and displays the Customer’s details. 6. The system asks for credit card information – name on card, card number and expiry date. 7. The Customer enters the credit card information. 8. The system asks for confirmation of the order. 9. The Customer confirms the order. 10. The system debits the credit card. 11. The system displays an invoice.
Secondary scenarios: InvalidCustomerIdentifier InvalidCreditCardDetails CreditCardLimitExceeded CreditCardExpired
Postconditions:

.Scenarios

- Secondary scenario of a use case, Fig. 4.14 [Arlow and Neustadt, 2002]

Use case: Checkout Secondary scenario: InvalidCustomerIdentifier
ID: UC15
Actors: Customer
Preconditions:
Secondary scenario: 1. The use case begins in step 3 of the use case Checkout when the Customer enters an invalid customer identifier. 2. For three invalid entries 2.1. The system asks the Customer to enter the customer identifier again. 3. The system informs the Customer that their customer identifier was not recognized.
Postconditions: