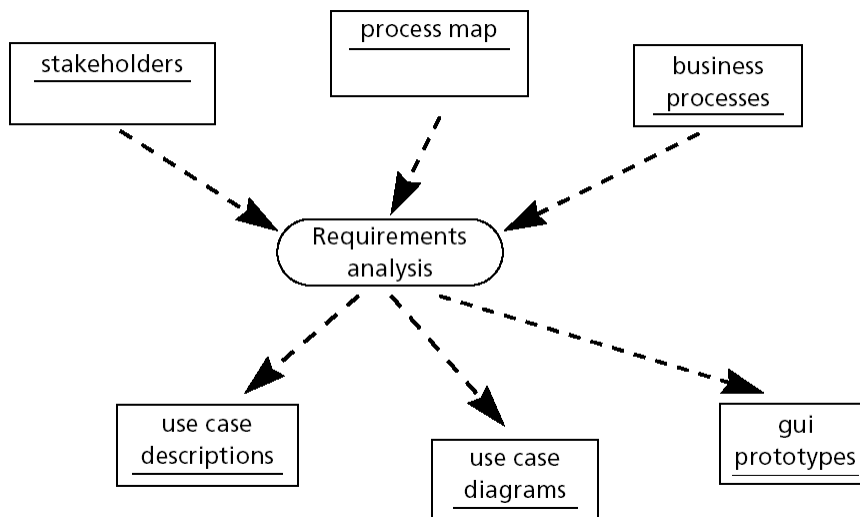
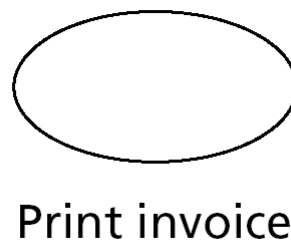


Figure 9.1 The key inputs and outputs of requirements analysis

Software Development with UML – Copyright Ken Lunn 2003

Figure 9.2 The UML notation for a use case to print an invoice

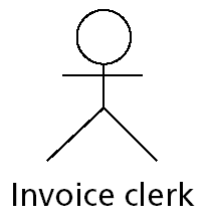
Software Development with UML – Copyright Ken Lunn 2003

Figure 9.3 An alternative UML notation for a use case to print an invoice



Software Development with UML – Copyright Ken Lunn 2003

Figure 9.4 The UML notation for an actor, an invoice clerk



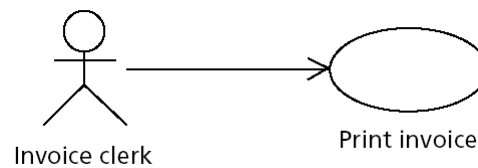
Software Development with UML – Copyright Ken Lunn 2003

Figure 9.5 An association between an actor and a use case in UML



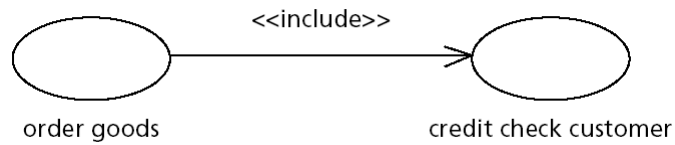
Software Development with UML – Copyright Ken Lunn 2003

Figure 9.6 An association between an actor and a use case, indicating the direction of initiation



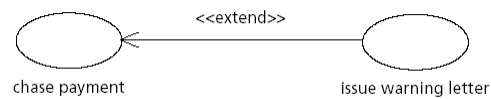
Software Development with UML – Copyright Ken Lunn 2003

Figure 9.7 An includes relationship between use cases, where one use case always includes the behaviour of another



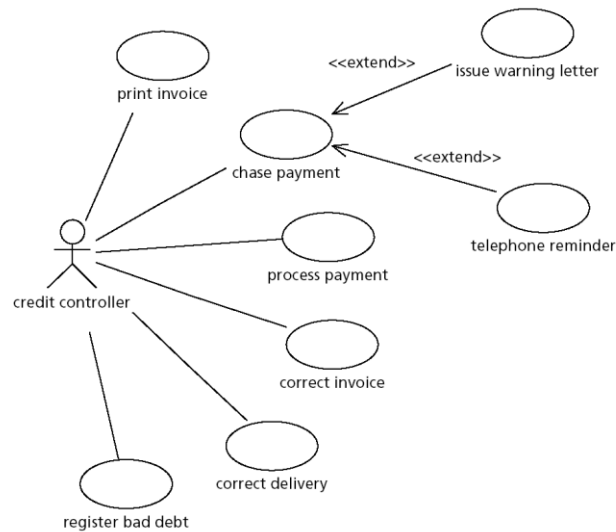
Software Development with UML – Copyright Ken Lunn 2003

Figure 9.8 An extends relationship where one use case extends another use case, the extender being sometimes called by the extended



Software Development with UML – Copyright Ken Lunn 2003

Figure 9.9 A use case diagram showing the functionality available to a credit control clerk



Software Development with UML – Copyright Ken Lunn 2003

Figure 9.10 A use case description format

Use case number:	Use case name:
Goal:	
Brief description:	
Actors:	
Frequency of execution:	
Scalability:	
Criticality:	
Other non-functional requirements:	
Primary path:	
Use cases related to primary path:	
Alternatives:	
Use cases related to alternatives:	
Exceptions:	
Use cases related to exceptions:	
Notes:	

Software Development with UML – Copyright Ken Lunn 2003

Figure 9.11 An example of a use case description

Use case number: 99	Use case name: Browse Catalogue
Goal: Produce invoice for a customer	
Brief description: The customer will browse through the catalogue of shoes.	
Actors: Estimated 100 hits per day within 3 months, rising to 1000 within a year.	
Frequency of execution: Estimated 100 hits per day within 3 months, rising to 1000 within a year.	
Scalability: Up to 100 concurrent users expected	
Criticality: Very. Without the Web site available, the company is not selling.	
Other non-functional requirements: None identified.	

Software Development with UML – Copyright Ken Lunn 2003

Figure 9.11 An example of a use case description (continued)

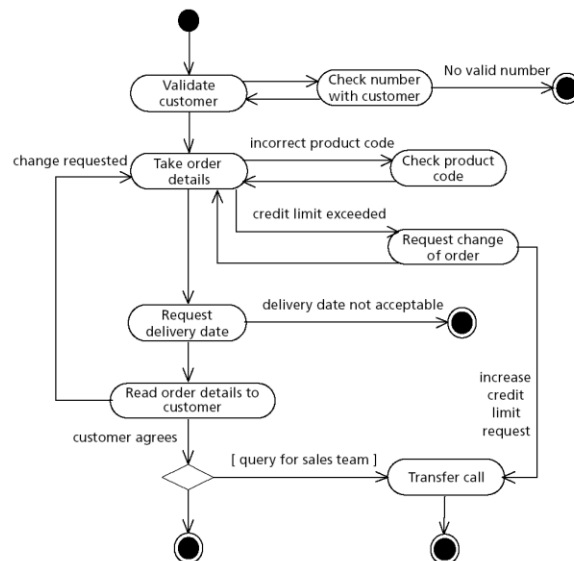
Primary path:
The following sequence is carried out for every customer on the sales ledger who has not been billed in the last month:
1. Get sales items from the sales ledger.
2. Get customer details from the customer file, covering billing address details.
3. Get any credits that the customer has.
4. Get discount details for customer.
5. Print the invoice header.
6. Print the line items on the invoice.
7. Calculate any discounts.
8. Apply any credits.
9. Calculate and print the invoice total.
10. Calculate and print the VAT.
11. Mark items on sales ledger as invoiced.
Use cases related to primary path:
None

Software Development with UML – Copyright Ken Lunn 2003

Figure 9.11 An example of a use case description (continued)

Alternatives:
2.1 No customer details on customer file, so print an error message on a report. Do not mark the items on the sales ledger as invoiced. The message needs to detail the sales items that have been entered.
Use cases related to alternatives:
Invoicing error report
Exceptions:
None
Use cases related to exceptions:
None
Notes:
Created by Ken Lunn, 14 August 2001

Software Development with UML – Copyright Ken Lunn 2003

Figure 9.12 Workflow for the telephone order process

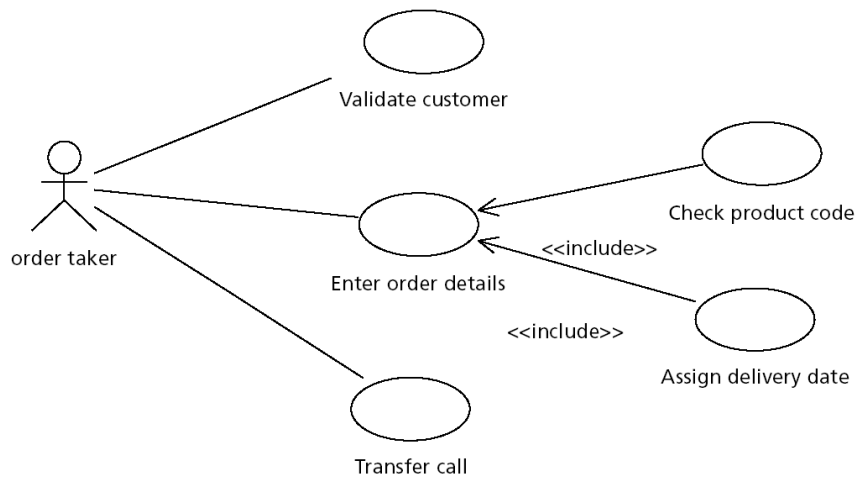
Software Development with UML – Copyright Ken Lunn 2003

Table 9.1 Candidate use cases for the ICANDO Chemicals telephone ordering process

Process step	Candidate use case
Validate Customer	Validate Customer
Check Number with Customer	None
Take Order Details	Enter Order Details
Check Product Code	Check Product Code
Request Delivery Date	Assign Delivery Date
Transfer Call	Transfer Call

Software Development with UML – Copyright Ken Lunn 2003

Figure 9.13 Use cases to support the telephone ordering process



Software Development with UML – Copyright Ken Lunn 2003

Figure 9.14 Use case description for Validate Customer

Use case number: 33	Use case name: Validate Customer
Goal: Ensure that a customer is valid before other functions for the customer are carried out.	
Brief description: This use case is applied whenever someone calls the call centre. It checks that there is a valid customer. Some calls may be handled without validation, but the norm for other use cases will be that this use case must have fired successfully.	
Actors: any call centre staff authorized to deal with customer queries	
Frequency of execution: The call centre takes approximately two thousand calls a day, and it is planned to take up to four thousand.	
Scalability: The peak number of agents expected is 500	
Criticality: This must run successfully for every call that involves orders or other customer service other than simple information. Therefore, highly critical.	
Other non-functional requirements: None identified.	
Preconditions: None	
Postconditions: The customer number is recorded as valid for the duration of the call.	

Software Development with UML – Copyright Ken Lunn 2003

Figure 9.14 Use case description for Validate Customer (continued)

Primary path:
<ol style="list-style-type: none"> 1. Customer supplies customer number and clerk types this in at the screen. 2. System responds with the customer name and address, and a unique password for the customer. 3. The operator asks for confirmation of the name and address and the password, and clicks a check box if the answers given conform to the data on the screen.
Use cases related to primary path: None
Alternatives:
<ol style="list-style-type: none"> 1.1 Customer does not have a customer number to hand. The operator must ask the customer to find the number before proceeding. 2.1 The system cannot find a customer record for the customer number. A message is displayed to indicate this. The clerk must request the customer to check the number and if it is incorrect the use case can restart again at 1. 3.1 The customer cannot supply the name and address or the password. The clerk must close the call. The system must register the failed attempt to enable checks for fraud.
Use cases related to alternatives:
None

Software Development with UML – Copyright Ken Lunn 2003

Figure 9.14 Use case description for Validate Customer (continued)

Exceptions: If the system is unavailable, the clerk must take the name and telephone number of the customer, and call them back as soon as the system is back online.
Use cases related to exceptions: None
Notes: None

Software Development with UML – Copyright Ken Lunn 2003

Figure 9.15 Use case description for Enter Order Details

Use case number: 34	Use case name: Enter Order Details
Goal: Capture order details for a customer.	
Brief description: Allows the clerk to enter the details of the order. Order details need to be checked against the product list, and to fit within the credit limit and contract of the customer.	
Actors: Any call centre staff authorized to deal with customer sales.	
Frequency of execution: The call centre takes approximately two thousand calls a day, and it is planned to take up to four thousand. Two thirds of these calls will be for orders.	
Scalability: The peak number of agents expected is 500	
Criticality: This must run successfully for every call requesting an order. Therefore, highly critical.	
Other non-functional requirements: None identified.	
Preconditions: The customer must have been validated.	
Postconditions: The order must have been recorded, and a delivery date confirmed with the customer. A log entry is made on the customer contact details, indicating the order number if the order is completed, or a reason for abandoning the order if the order is not completed.	

Software Development with UML – Copyright Ken Lunn 2003

Figure 9.15 Use case description for Enter Order Details (continued)**Primary path:**

1. Customer supplies product code and quantity, and the clerk types this on the screen. This is repeated as often as necessary.
2. The customer requests a particular delivery date, and the clerk types this in. The system confirms this is acceptable.
3. The clerk reads the order and the delivery date back to the customer, who agrees.
4. The clerk hits the confirm button, and the system responds saying that the order has been accepted. The clerk thanks the customer and asks if there is any other service needed.

Use cases related to primary path:

1. Check Product Code
2. Assign Delivery Date

Software Development with UML – Copyright Ken Lunn 2003

Figure 9.15 Use case description for Enter Order Details (continued)**Alternatives:**

- 1.1 Customer does not have a product code to hand, but does know the product name. The system should allow the entry of a product name instead of a product number, and then retrieve the product number.
 - 2.1 The system cannot meet the requested delivery date, but offers a nearby one. The customer may accept the nearer one, or suggest a different date. This can continue until a mutually acceptable date is found.
 - 3.1 The customer spots an error, and the clerk must change the detail.
 - 4.1 The order exceeds the customer's credit limit, or includes a product not scheduled on the contract. This must be relayed to the customer, and corrective action agreed.
- At any point, the order may be abandoned. This happens in a small number of cases. On exit, the clerk must choose from a set of reasons for abandoning the order, or write a reason for abandonment if there is not a prescribed reason.

Software Development with UML – Copyright Ken Lunn 2003

Figure 9.15 Use case description for Enter Order Details (continued)

Use cases related to alternatives: 1.1 Check Product Code 2.1 Assign Delivery Date
Exceptions: None identified
Use cases related to exceptions: None
Notes: None