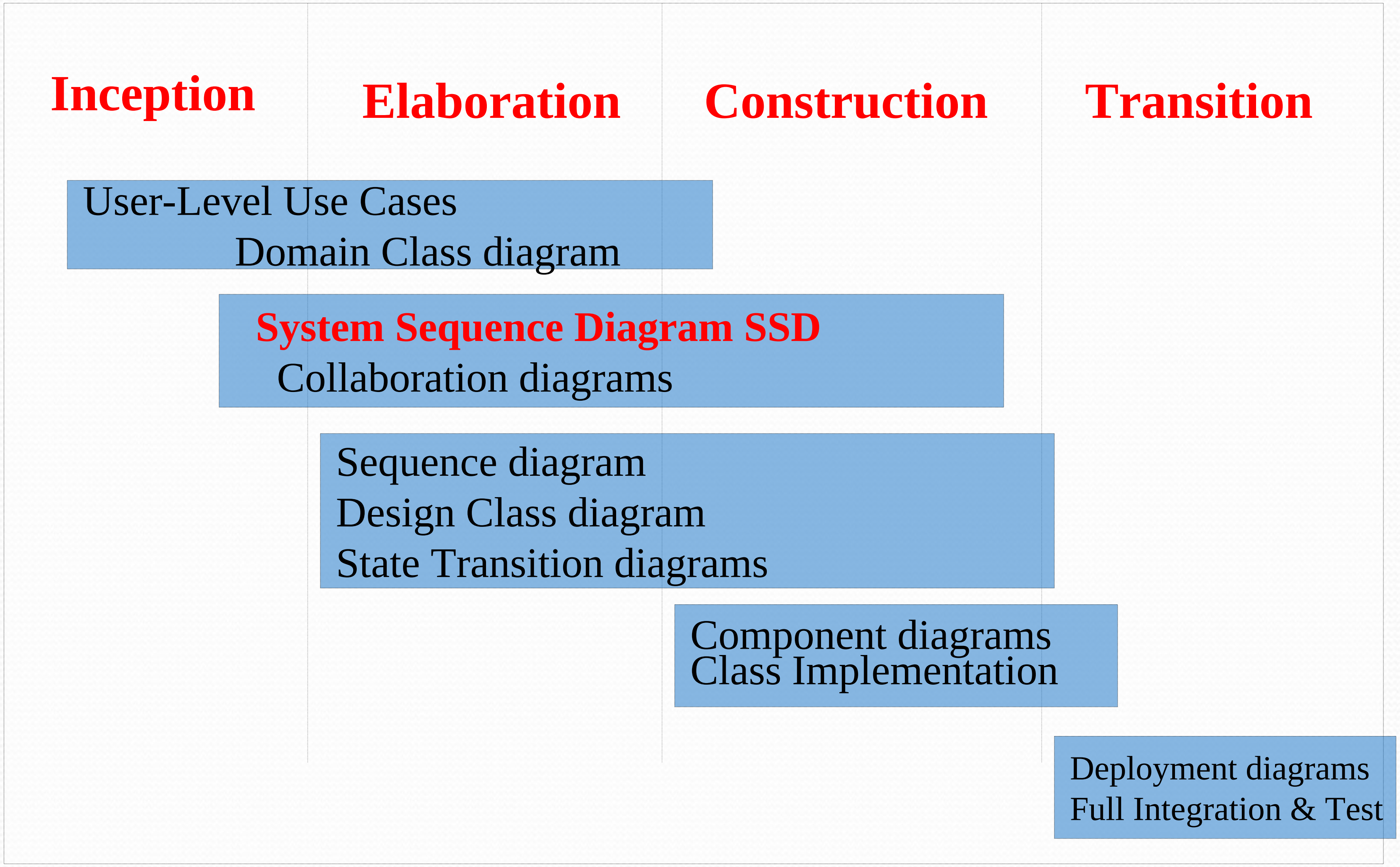
COMP 3711

(OOA and OOD)

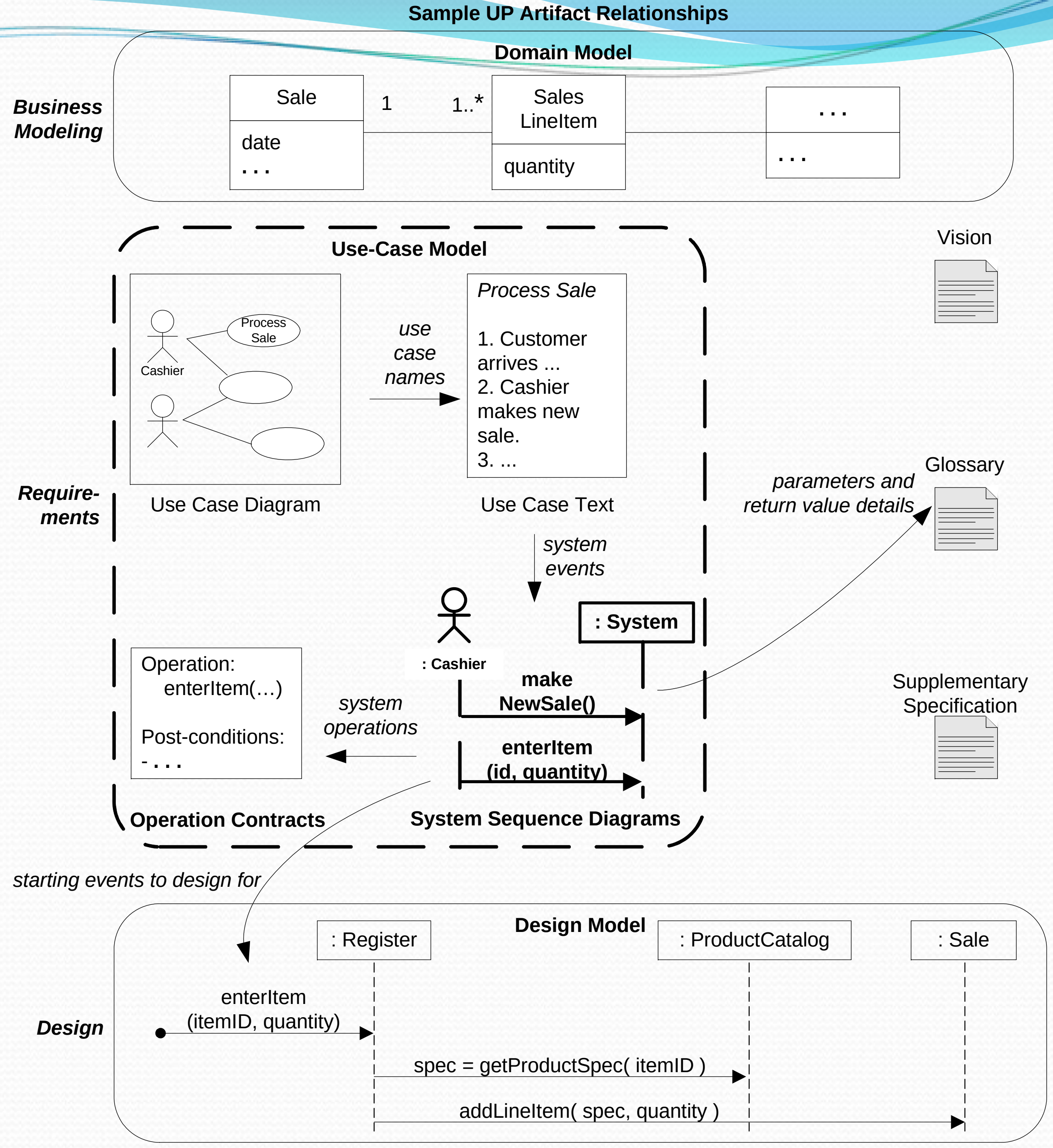
System Sequence Diagram

Larman Chapter 10, 32

UML And UP


Inception
Elaboration
Construction
Transition
User-Level Use Cases
Domain Class diagram
System Sequence Diagram SSD
Collaboration diagrams
Sequence diagram
Design Class diagram
State Transition diagrams
Component diagrams
Class Implementation
Deployment diagrams
Full Integration & Test

SSD
Part of requirements gathering



Already Constructed UML Model

- Use Case
  - The focus in the Use Case is the Actors interacting with the System.
  - The actor generates events to the system
  - Events initiate operations upon the system

- Domain Model
  - The focus in the Domain model is the relationships between the conceptual classes.

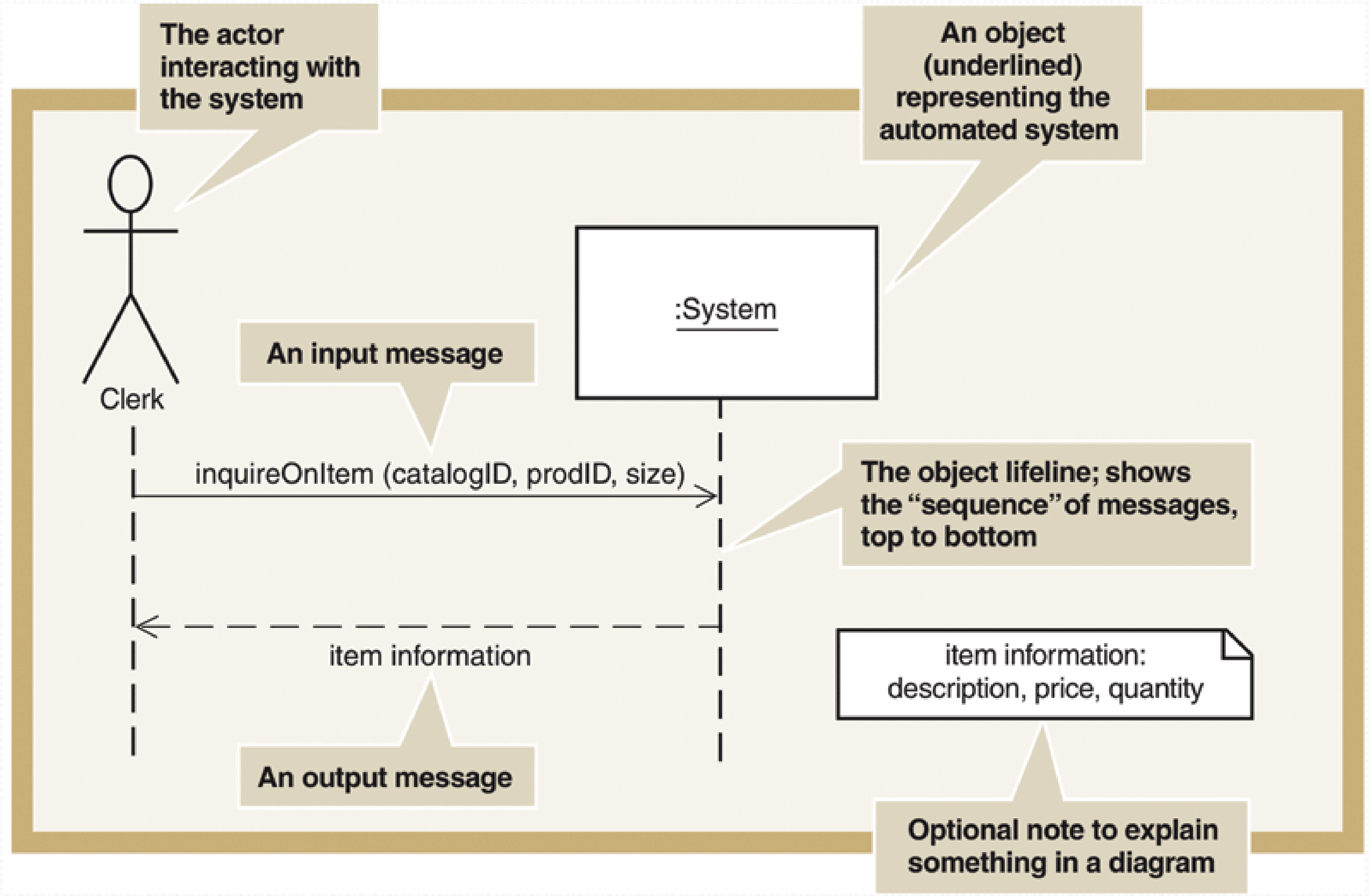
The System Sequence Diagram

- System Sequence Diagrams (SSDs) are derived from Use Cases.
- A SSD shows <u>*one*</u> Use Case Scenario.
- SSD is for a main success scenario of a Use Case.

The System Sequence Diagram

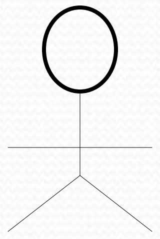
- SSD identifies Inputs and Outputs of a Use Case Scenario and models the input and output messaging requirements for a Use Case or scenario.
- Shows actors interacting with system.
- Shows sequence of interactions as messages during flow of activities.
- The system is shown as one object: a "black box"

System Sequence Diagram (SSD) Notation


The actor interacting with the system
An object (underlined) representing the automated system
Clerk
:System
An input message
inquireOnItem (catalogID, prodID, size)
The object lifeline; shows the "sequence" of messages, top to bottom
item information
item information: description, price, quantity
An output message
Optional note to explain something in a diagram

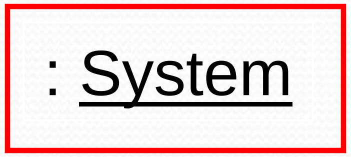
SSD (System Sequence Diagram) Notation

- **Actor** represented by a stick figure – a person (or role) that interacts with system by entering input data and receiving output data



SSD (System Sequence Diagram) Notation

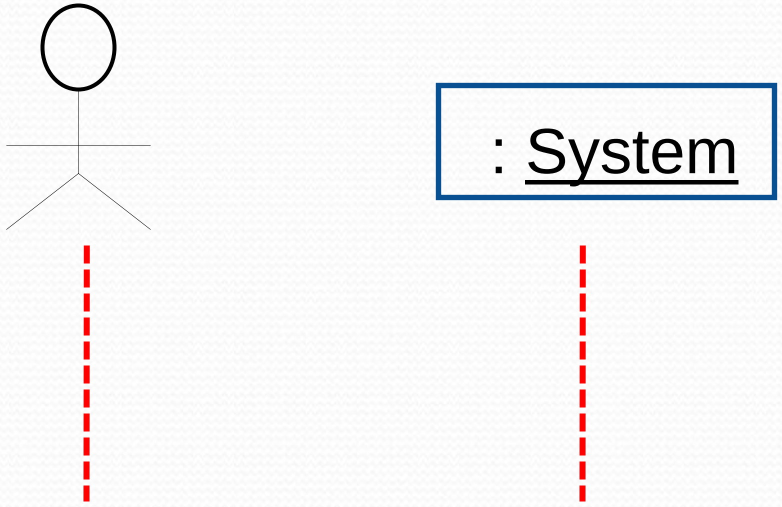
- **Object** is a rectangle with name of object underlined – shows individual object and not class of all similar objects (:System for SSD)



- Technically speaking the entities in sequence diagrams are objects.
- The class to which the object belongs is also included
- The object name may be omitted if the behaviour applies to all objects of a class
- No object attributes are listed.

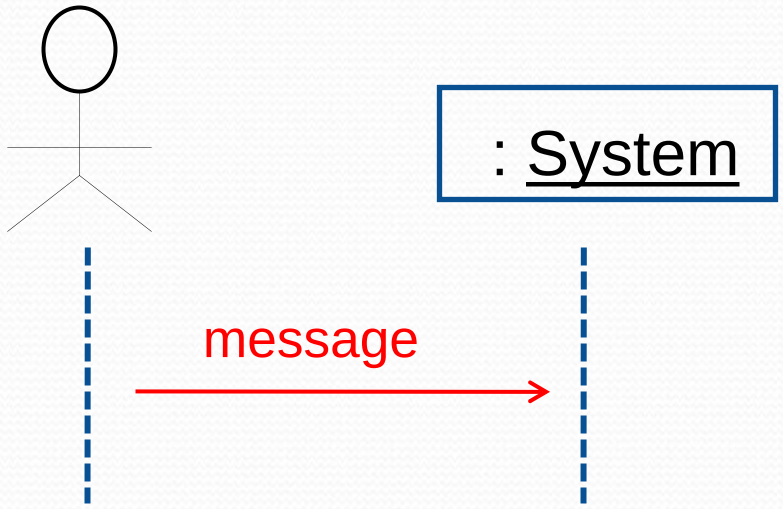
SSD (System Sequence Diagram) Notation

- Lifeline or object lifeline is a vertical line under object or actor to show passage of time for object
- If vertical line dashed
  - Creation and destruction of thing is not important for scenario



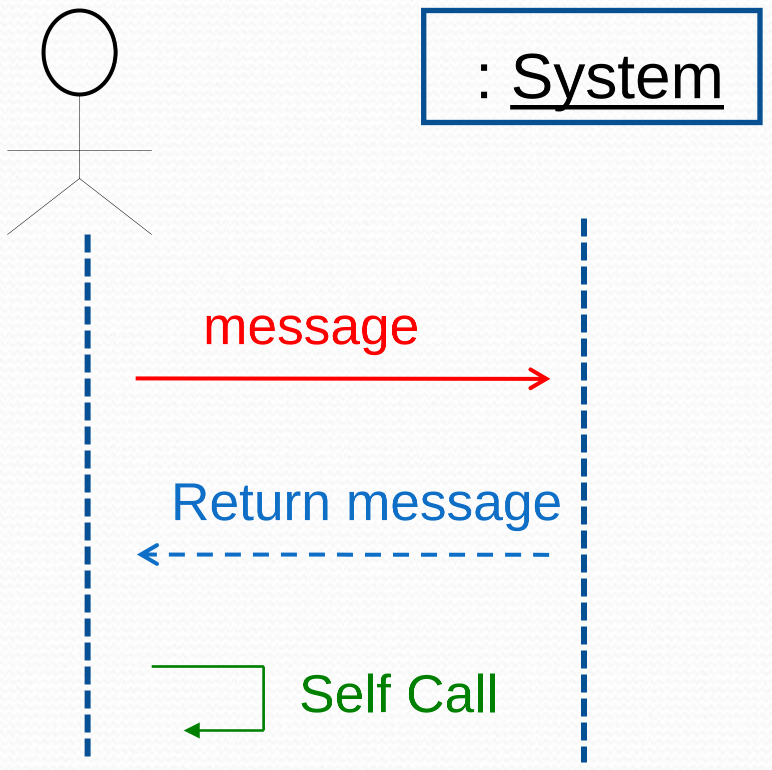
SSD (System Sequence Diagram) Notation

- Messages are internal events identified by the flow of objects in a scenario They are requests from one actor or object to another to do some action. A message invokes a particular method.

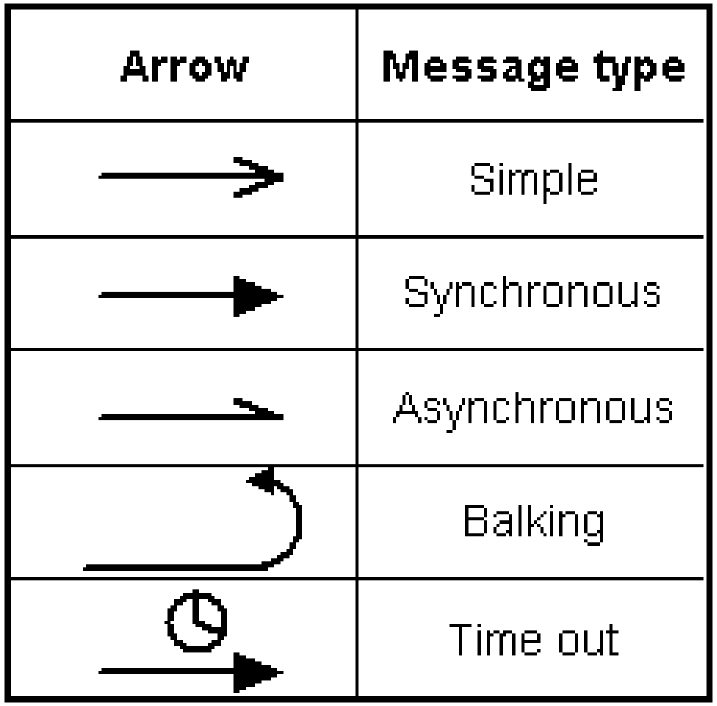


SSD (System Sequence Diagram) Notation

- Messages are labelled on arrows to show messages sent to or received by actor or system



SSD Extended Notation

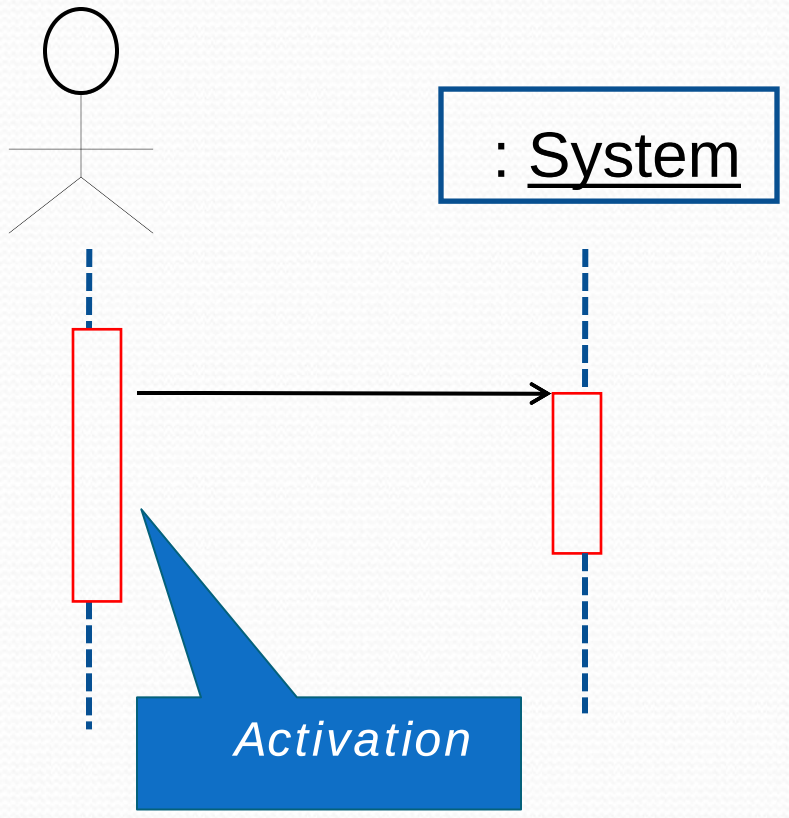


Arrow	Message type
→	Simple
➡	Synchronous
⇀	Asynchronous
↩	Balking
◷ ➡	Time out

- Synchronous – two objects exchanging a message with sender waits for handler completes (single-threaded)
- Asynchronous - message being passed to another object without yielding to control, (multi-threaded)
- Balking – object absorbs the message transfer from another due to immediate unavailability (e.g. refusal or queue full)
- Time Out – a fixed waiting time for receiver object to accept message, then balk

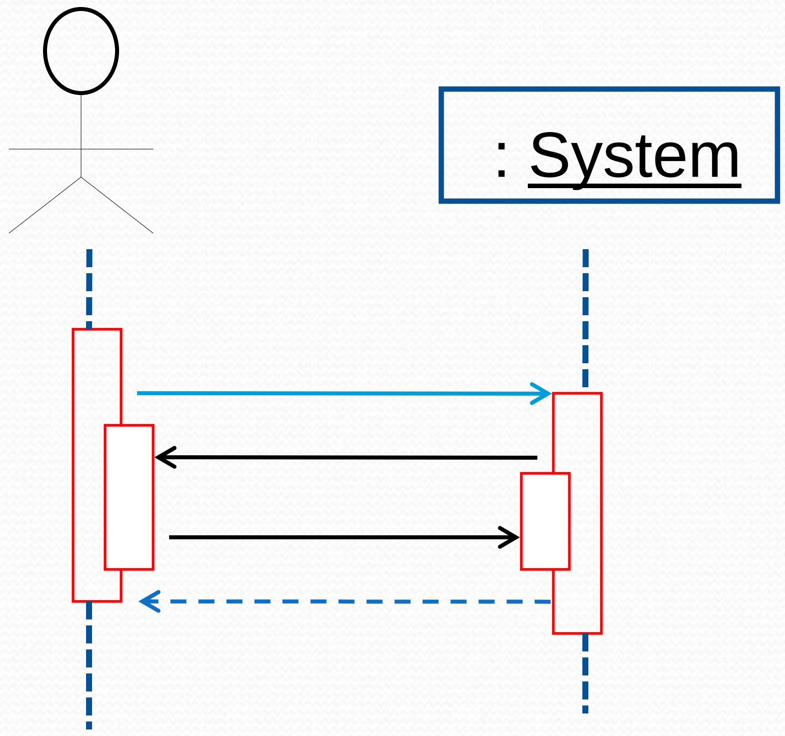
SSD (System Sequence Diagram) Notation

- Activation boxes represent the time an object (active) needs to complete a task



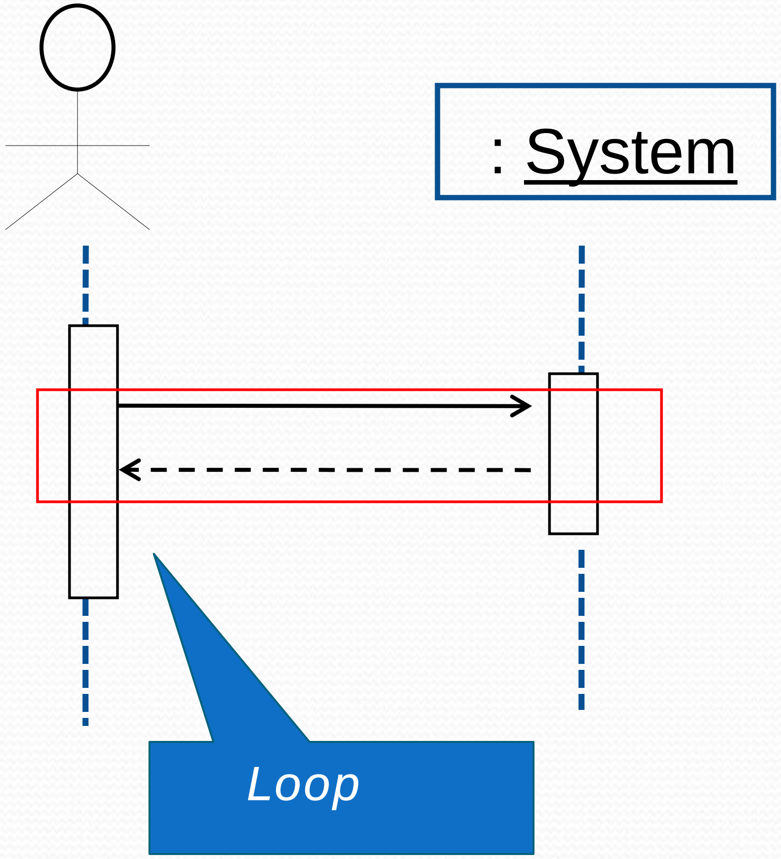
SSD (System Sequence Diagram) Notation

- Activation boxes make it easier to visualize and unambiguous as they emit from the activation for the message that returns the value



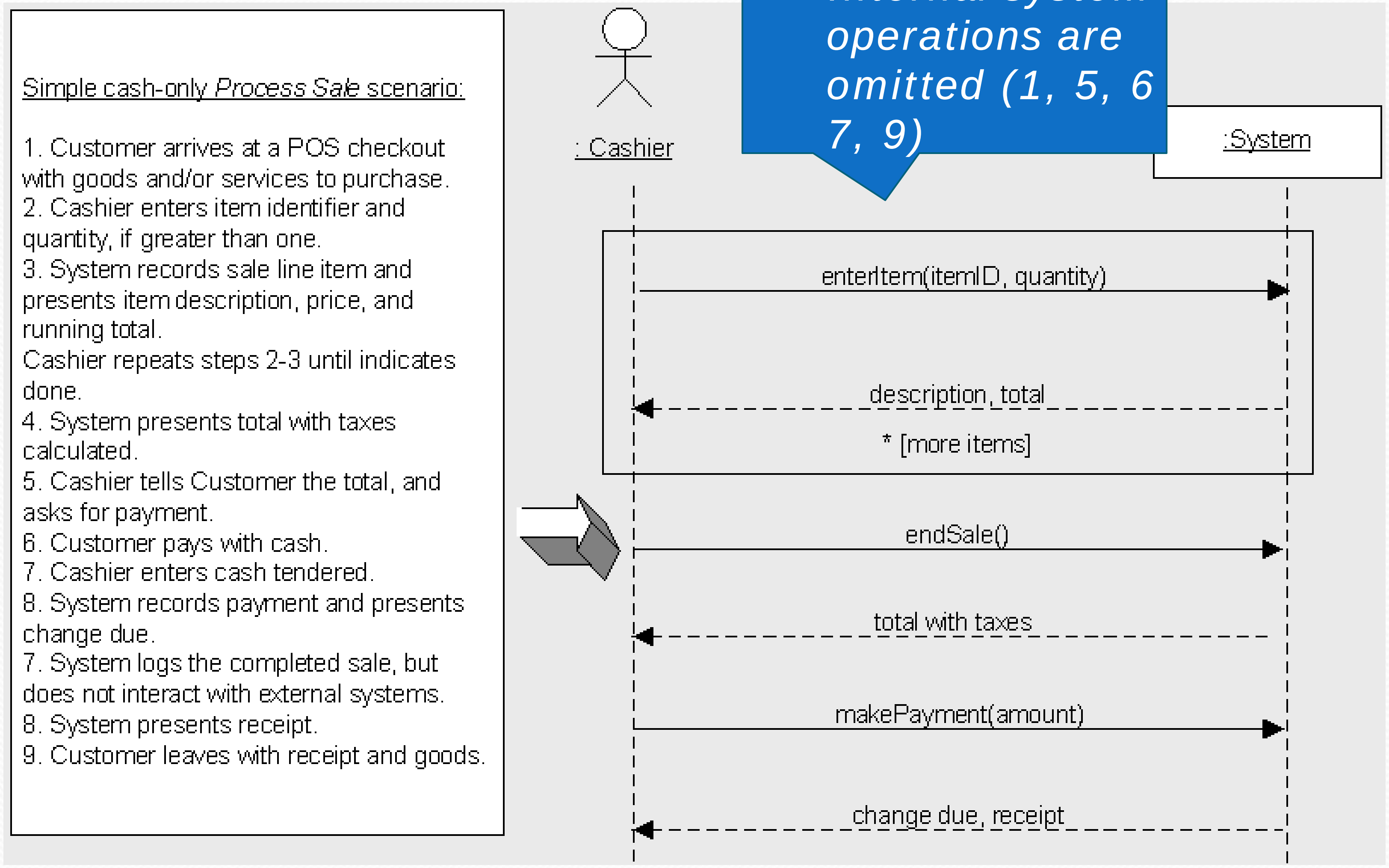
SSD (System Sequence Diagram) Notation

- Loop or Repetition within a SSD is depicted as a rectangle.
- Some notation has the exit loop condition placed in the bottom left corner with square brackets [ ]
- Loop is a type of the “Interaction Frame”, other examples: reference, options



Example: Use Case to SSD

Non-system operations and Internal system operations are omitted (1, 5, 6 7, 9)



Simple cash-only *Process Sale* scenario:

1. Customer arrives at a POS checkout with goods and/or services to purchase.
2. Cashier enters item identifier and quantity, if greater than one.
3. System records sale line item and presents item description, price, and running total.
Cashier repeats steps 2-3 until indicates done.
4. System presents total with taxes calculated.
5. Cashier tells Customer the total, and asks for payment.
6. Customer pays with cash.
7. Cashier enters cash tendered.
8. System records payment and presents change due.
7. System logs the completed sale, but does not interact with external systems.
8. System presents receipt.
9. Customer leaves with receipt and goods.

Sequence diagram shows events for a use case scenario

SSD (System Sequence Diagram)

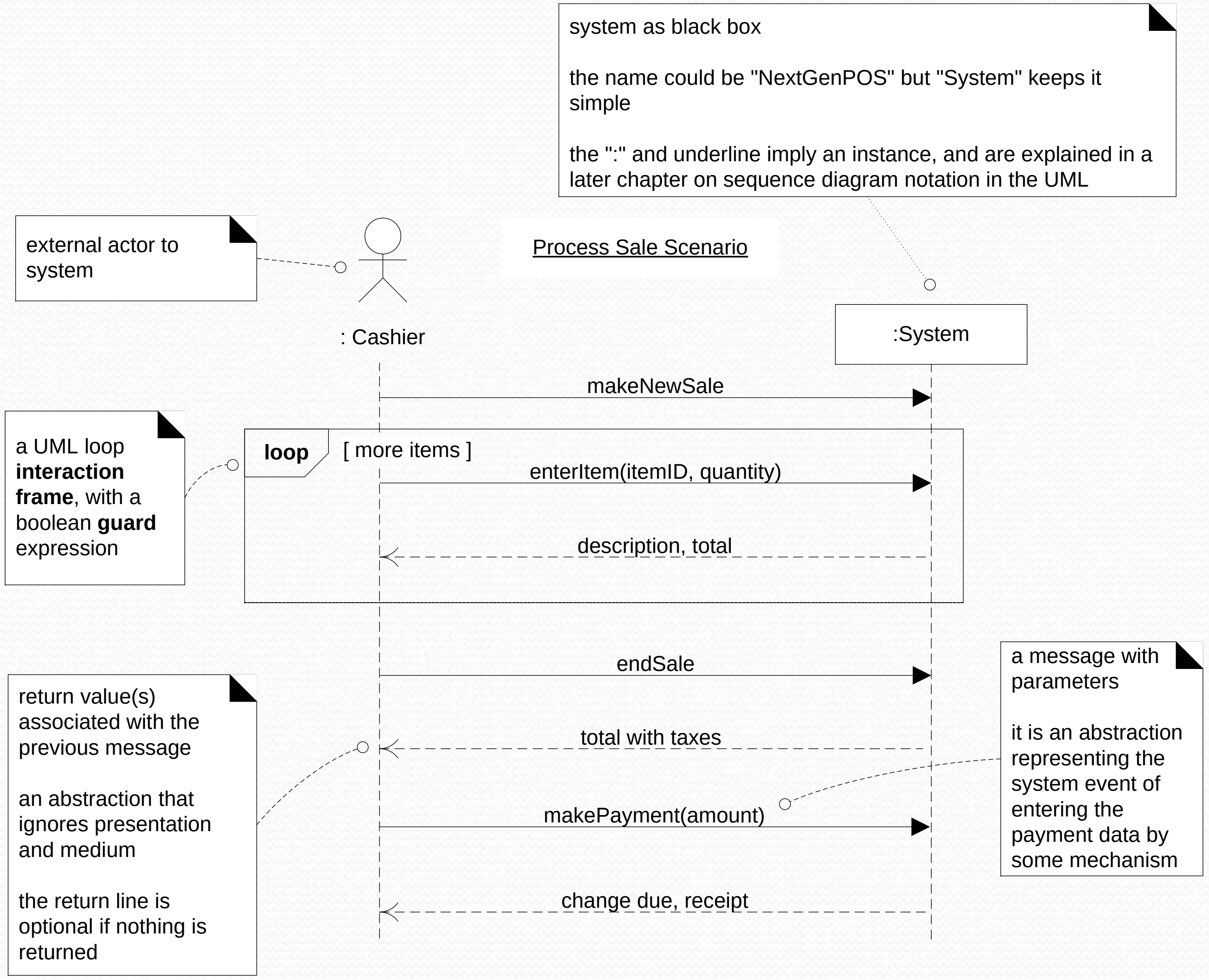

system as black box
the name could be "NextGenPOS" but "System" keeps it simple
the ":" and underline imply an instance, and are explained in a later chapter on sequence diagram notation in the UML
external actor to system
Process Sale Scenario
: Cashier
:System
makeNewSale
a UML loop **interaction frame**, with a boolean **guard** expression
loop
[more items]
enterItem(itemID, quantity)
description, total
endSale
a message with parameters
it is an abstraction representing the system event of entering the payment data by some mechanism
return value(s) associated with the previous message
an abstraction that ignores presentation and medium
the return line is optional if nothing is returned
total with taxes
makePayment(amount)
change due, receipt

Fig 10.2 – Larman

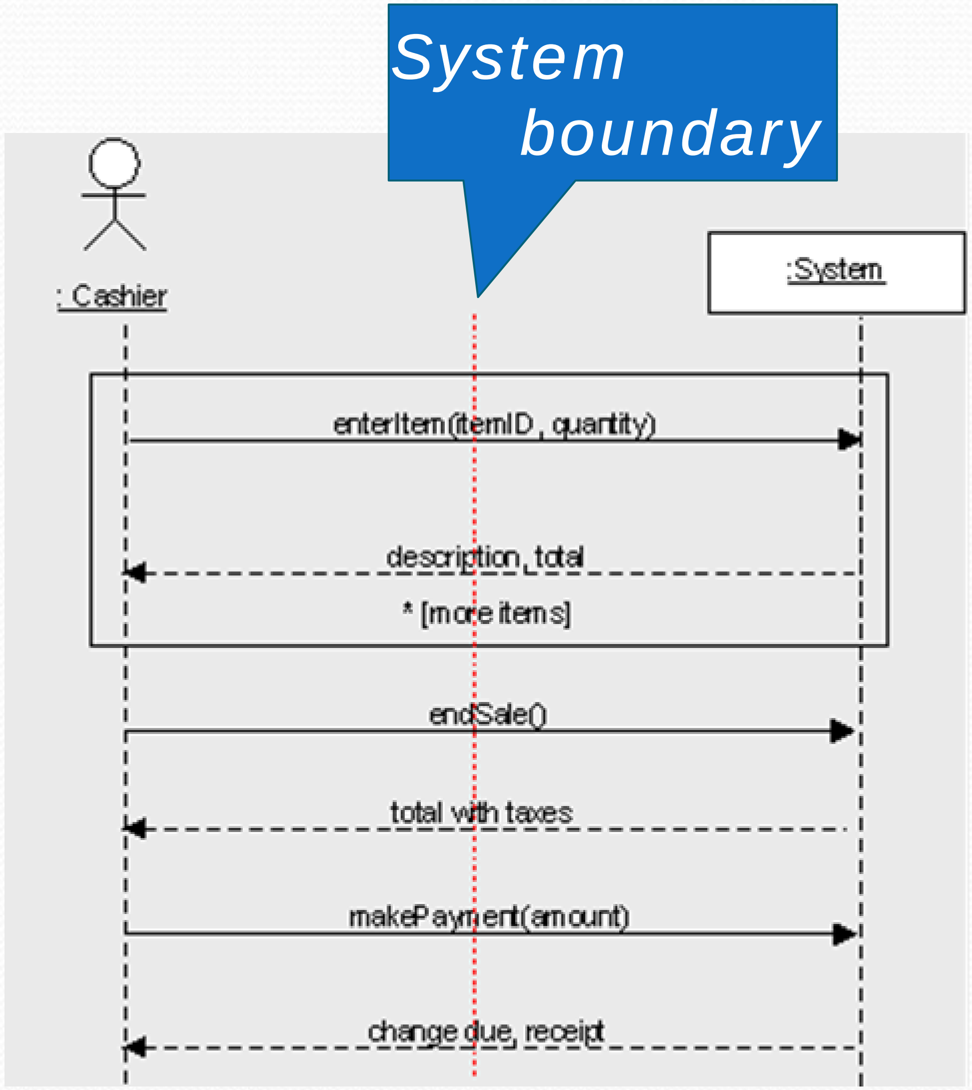
Naming SSD system events and operators

- Express events and operations at the level of intent rather than in terms of physical input medium
- Start event names with a verb:
  - addXXX
  - enterXXX
  - endXXX
  - makeXXX ….

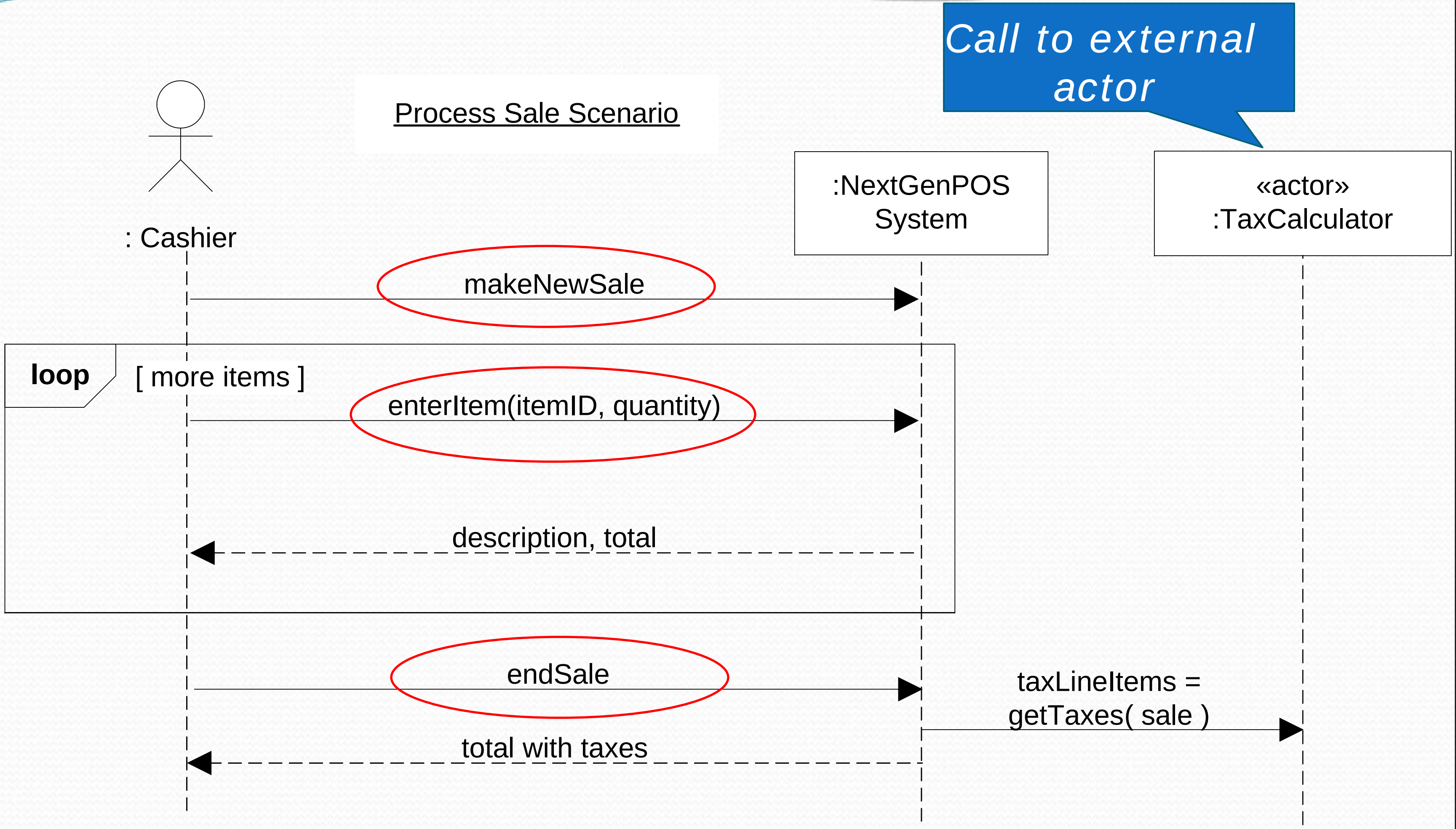
"Capture Intent not input device"

Identify System Boundary

- System Boundary must be clearly defined.
- System Boundary is usually chosen to be the software system itself.
- Notice that internal system operations are not included in SSD.



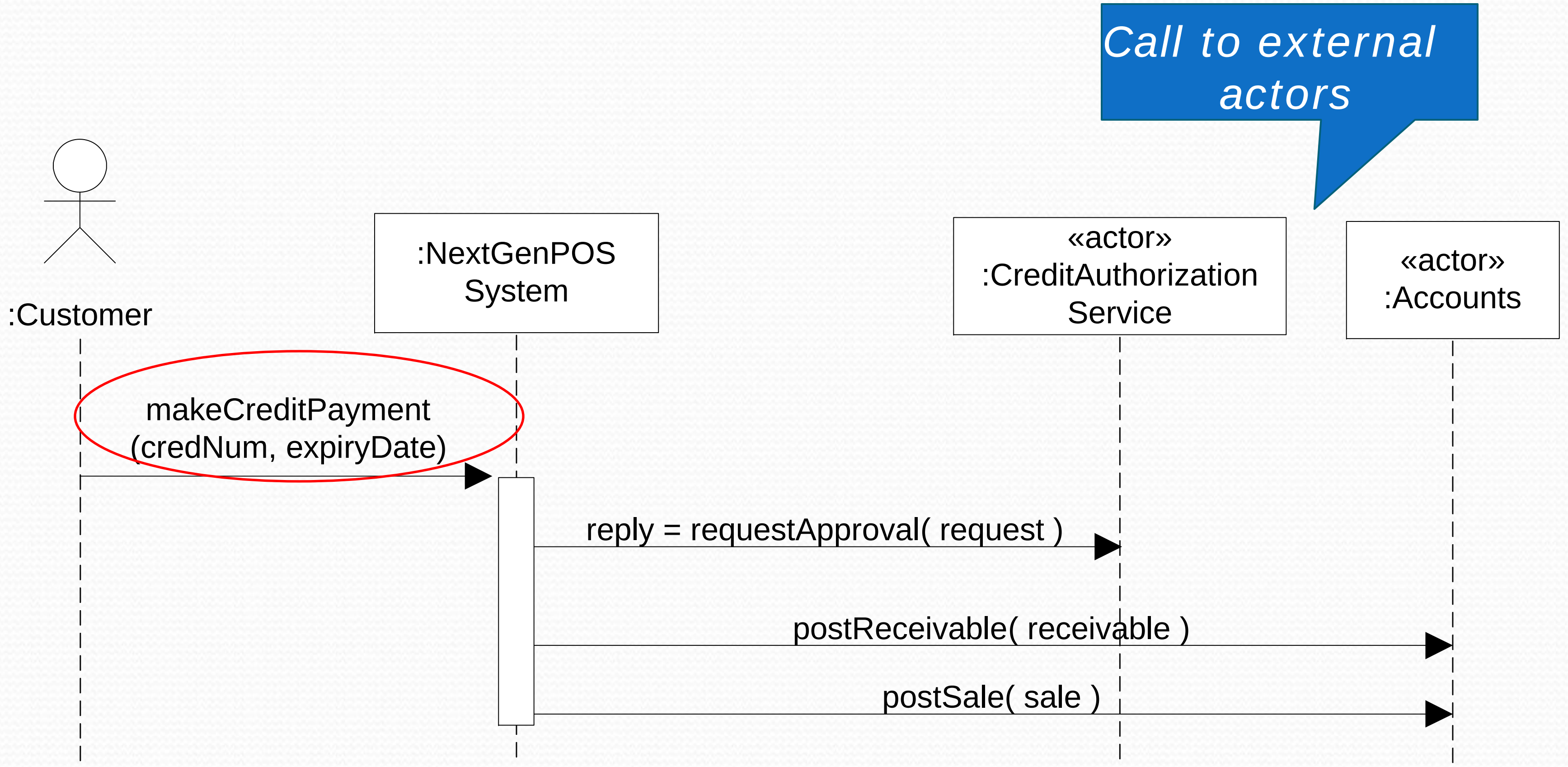
Extend POS - Process Sales Scenario



- Cashier processes the customer purchase with three operations: makeNewSales, enterItem, endSale

Larman Fig 32.1

Extend POS – SSD Process Sales

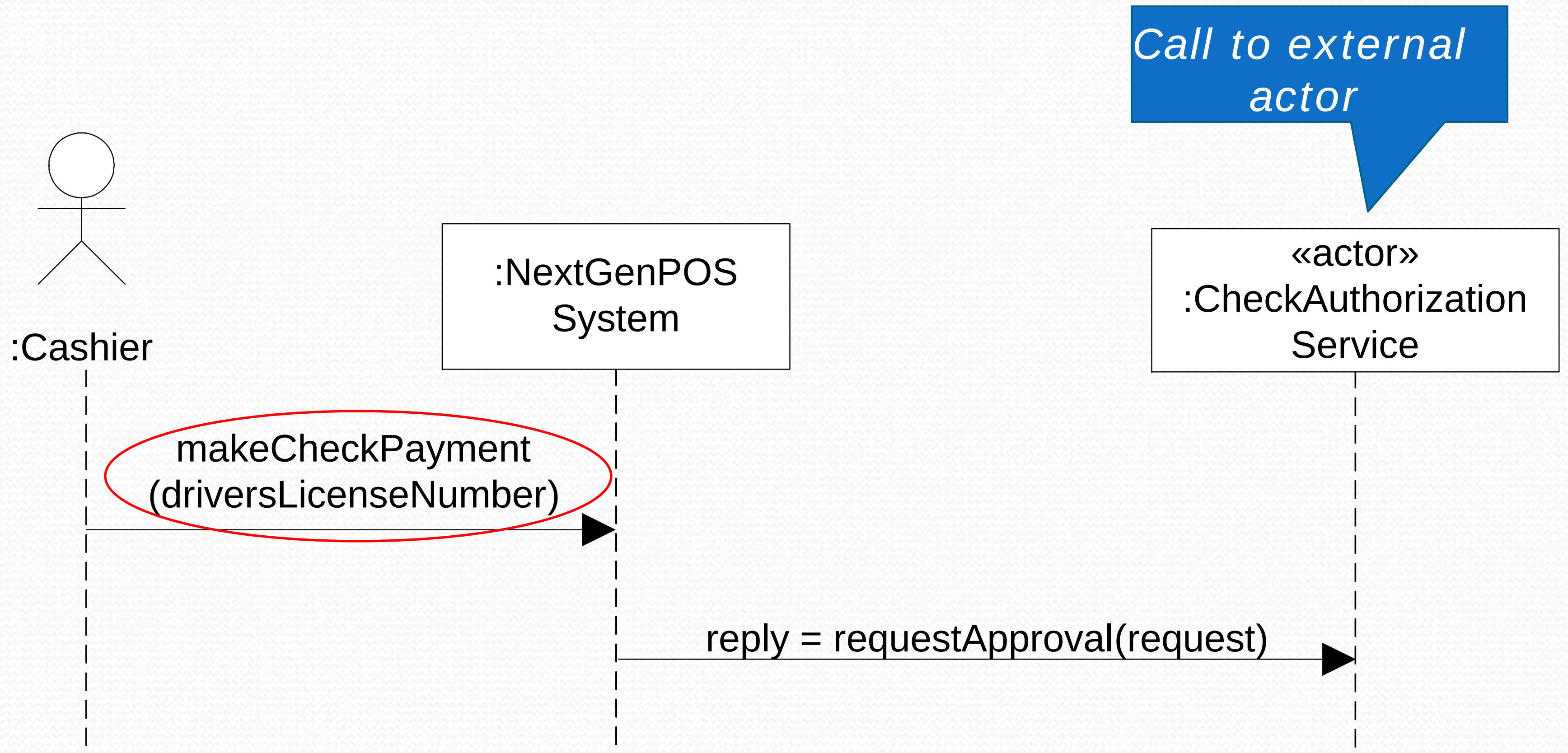


Call to external actors
:Customer
:NextGenPOS System
«actor» :CreditAuthorization Service
«actor» :Accounts
makeCreditPayment (credNum, expiryDate)
reply = requestApproval(request)
postReceivable(receivable)
postSale(sale)

- For Credit Card Payment, the customer interacts with the POS. The payment needs to be authorized and then added to accounts receivable.

Larman Fig 32.2

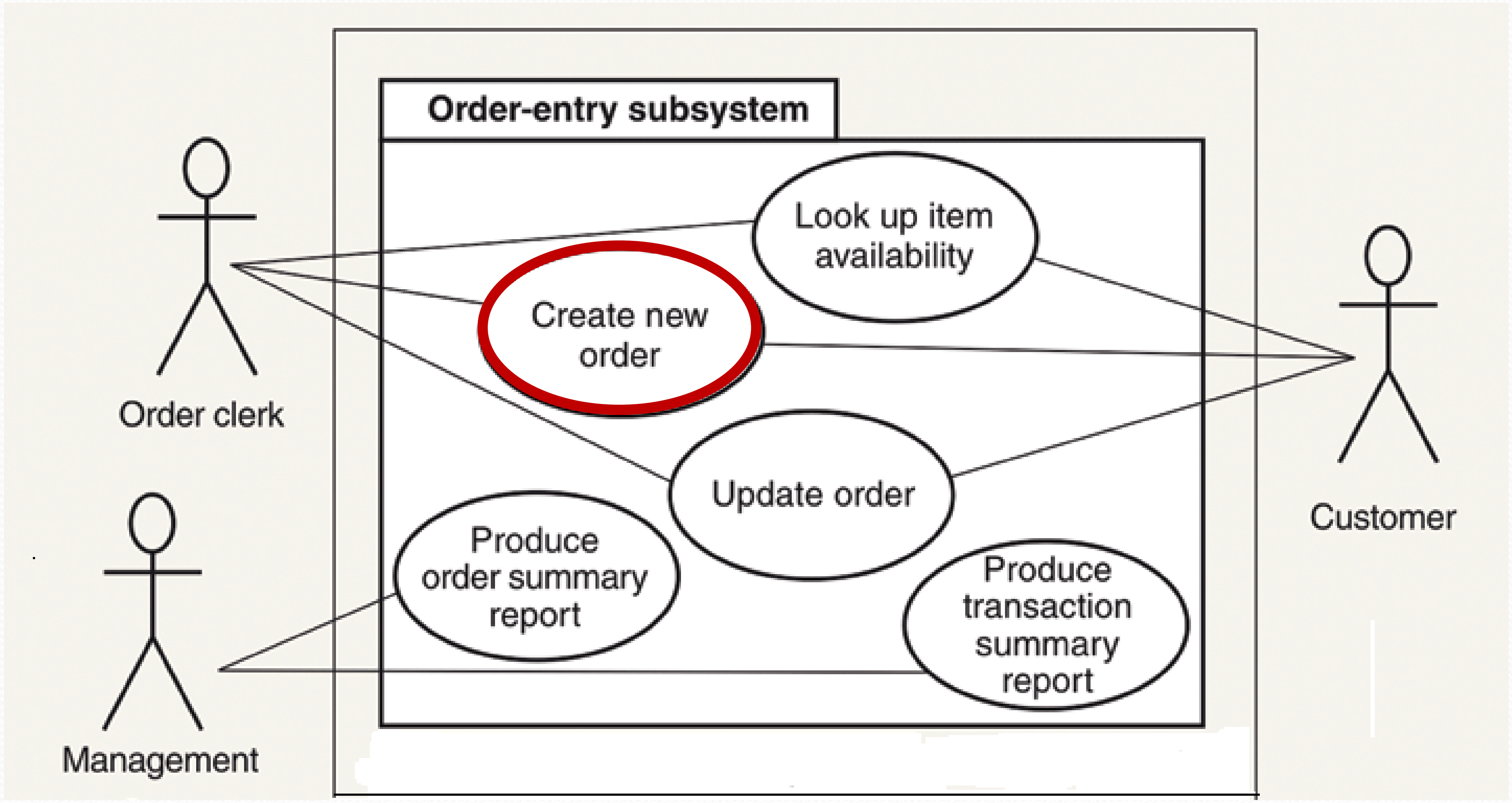
Extend POS – SSD Process Sales


Call to external actor
:Cashier
:NextGenPOS System
«actor» :CheckAuthorization Service
makeCheckPayment (driversLicenseNumber)
reply = requestApproval(request)

- For Check Payment, the customer interacts with the Cashier who process the sales through the POS. The driver licence must be recorded for validation.

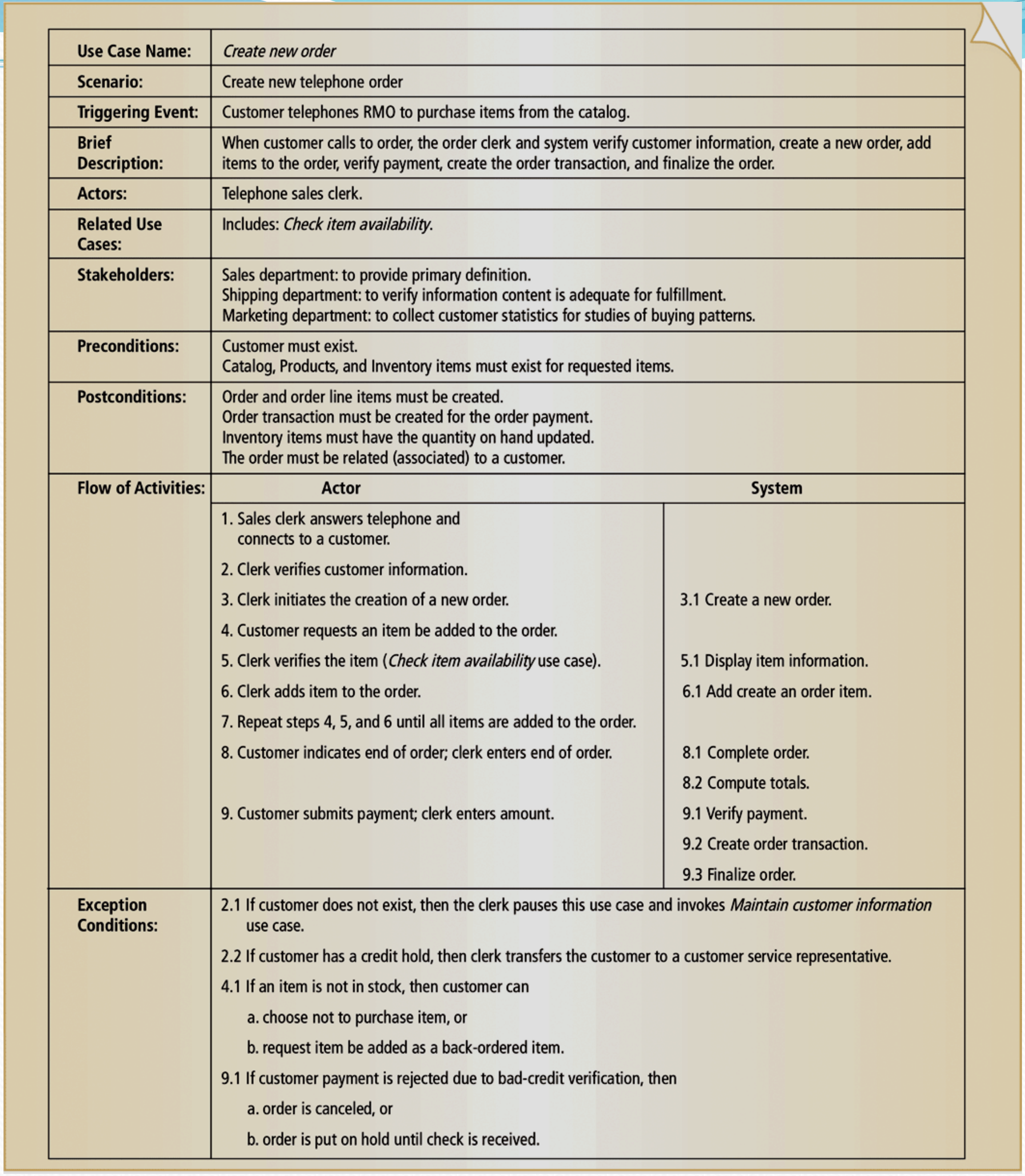
Larman Fig 32.3

Example: An Old Familiar Use Cases of Order Entry Subsystem


Order-entry subsystem
Order clerk
Management
Customer
Look up item availability
Create new order
Update order
Produce order summary report
Produce transaction summary report

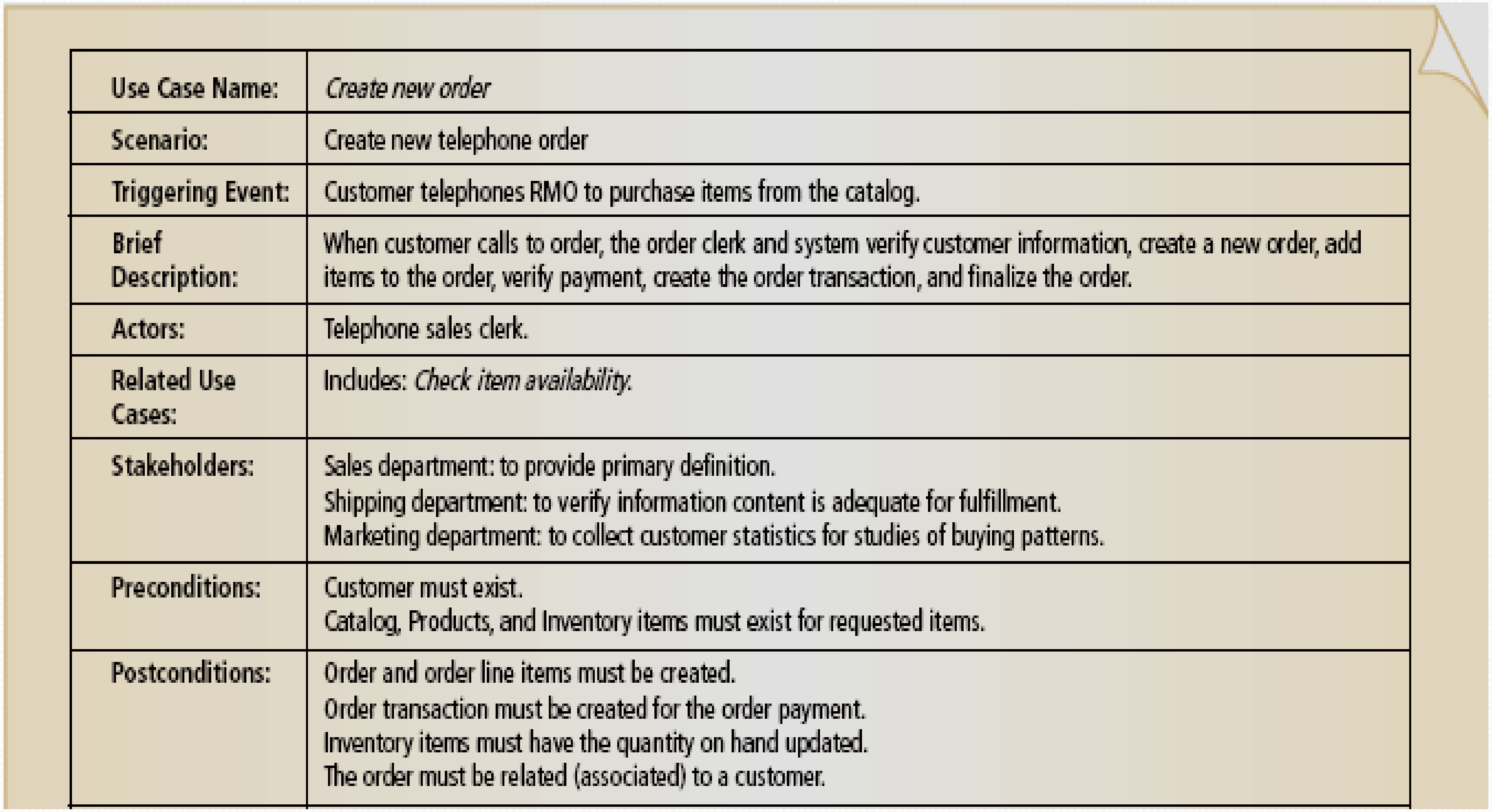
Use Case Description -

Example of Telephone Order Scenario for *Create New Order* Use Case



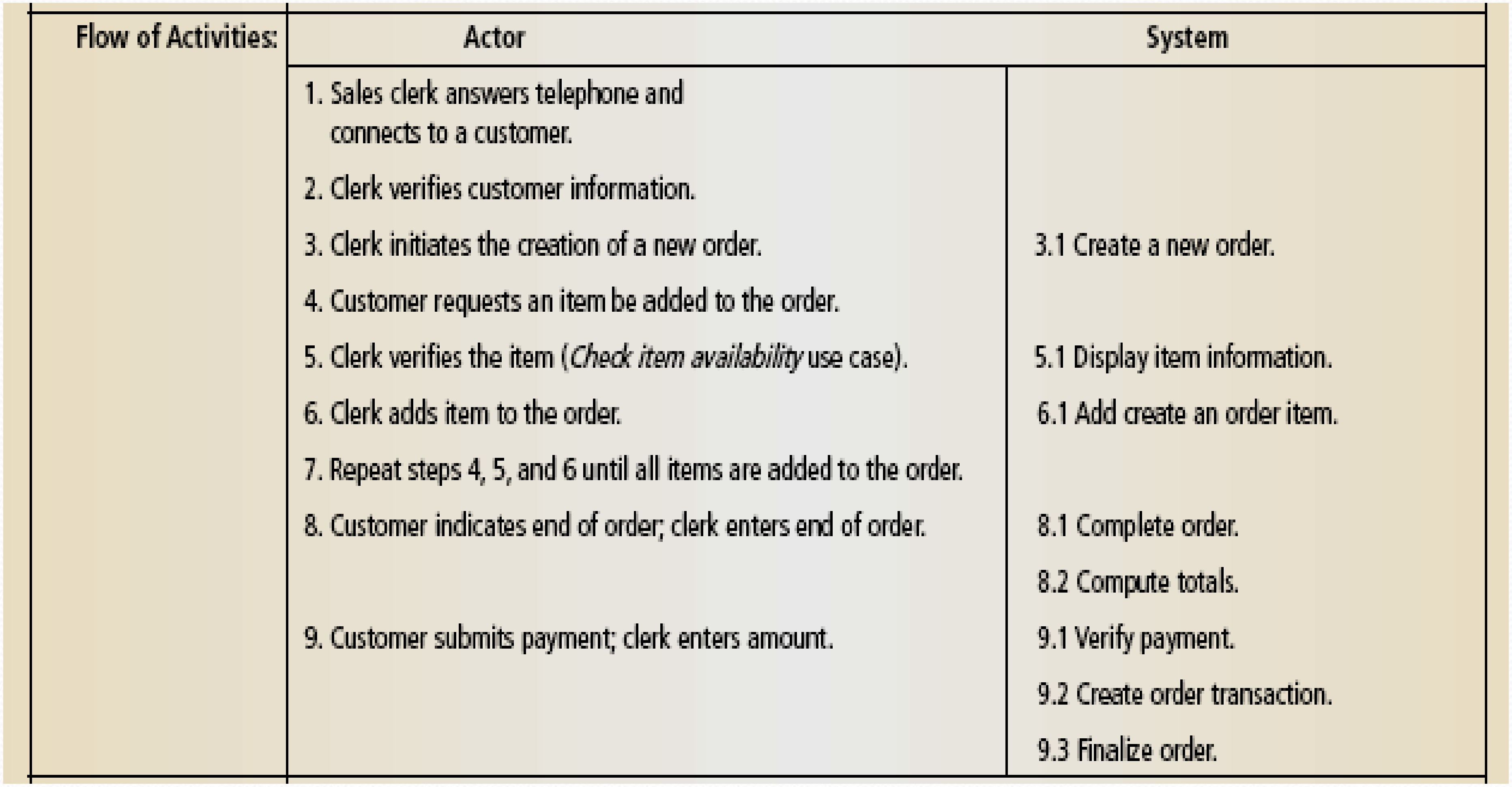
Use Case Name:	*Create new order*	
Scenario:	Create new telephone order	
Triggering Event:	Customer telephones RMO to purchase items from the catalog.	
Brief Description:	When customer calls to order, the order clerk and system verify customer information, create a new order, add items to the order, verify payment, create the order transaction, and finalize the order.	
Actors:	Telephone sales clerk.	
Related Use Cases:	Includes: *Check item availability.*	
Stakeholders:	Sales department: to provide primary definition. Shipping department: to verify information content is adequate for fulfillment. Marketing department: to collect customer statistics for studies of buying patterns.	
Preconditions:	Customer must exist. Catalog, Products, and Inventory items must exist for requested items.	
Postconditions:	Order and order line items must be created. Order transaction must be created for the order payment. Inventory items must have the quantity on hand updated. The order must be related (associated) to a customer.	
Flow of Activities:	**Actor**	**System**
	1. Sales clerk answers telephone and connects to a customer.	
	2. Clerk verifies customer information.	
	3. Clerk initiates the creation of a new order.	3.1 Create a new order.
	4. Customer requests an item be added to the order.	
	5. Clerk verifies the item (*Check item availability* use case).	5.1 Display item information.
	6. Clerk adds item to the order.	6.1 Add create an order item.
	7. Repeat steps 4, 5, and 6 until all items are added to the order.	
	8. Customer indicates end of order; clerk enters end of order.	8.1 Complete order. 8.2 Compute totals.
	9. Customer submits payment; clerk enters amount.	9.1 Verify payment. 9.2 Create order transaction. 9.3 Finalize order.
Exception Conditions:	2.1 If customer does not exist, then the clerk pauses this use case and invokes *Maintain customer information* use case. 2.2 If customer has a credit hold, then clerk transfers the customer to a customer service representative. 4.1 If an item is not in stock, then customer can a. choose not to purchase item, or b. request item be added as a back-ordered item. 9.1 If customer payment is rejected due to bad-credit verification, then a. order is canceled, or b. order is put on hold until check is received.	

Top Detail from Use Case Description Telephone Order



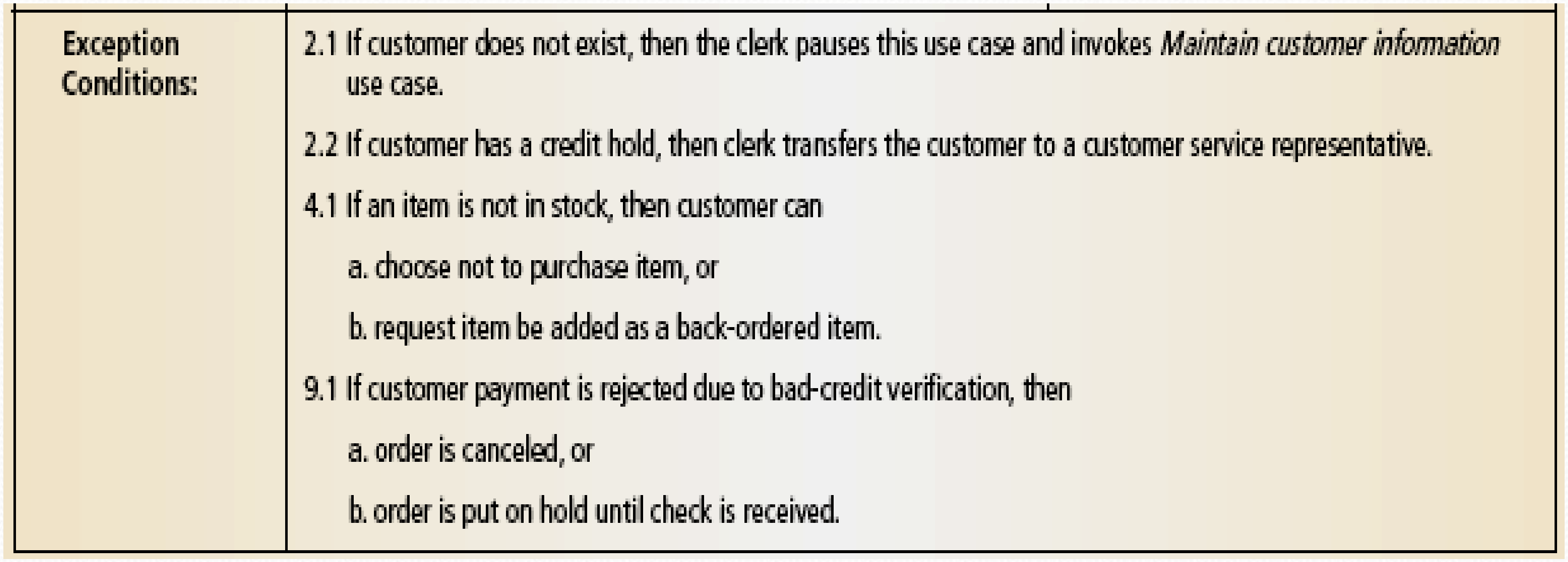
Use Case Name:	*Create new order*
Scenario:	Create new telephone order
Triggering Event:	Customer telephones RMO to purchase items from the catalog.
Brief Description:	When customer calls to order, the order clerk and system verify customer information, create a new order, add items to the order, verify payment, create the order transaction, and finalize the order.
Actors:	Telephone sales clerk.
Related Use Cases:	Includes: *Check item availability.*
Stakeholders:	Sales department: to provide primary definition. Shipping department: to verify information content is adequate for fulfillment. Marketing department: to collect customer statistics for studies of buying patterns.
Preconditions:	Customer must exist. Catalog, Products, and Inventory items must exist for requested items.
Postconditions:	Order and order line items must be created. Order transaction must be created for the order payment. Inventory items must have the quantity on hand updated. The order must be related (associated) to a customer.

Middle Detail from Use Case Description Telephone Order



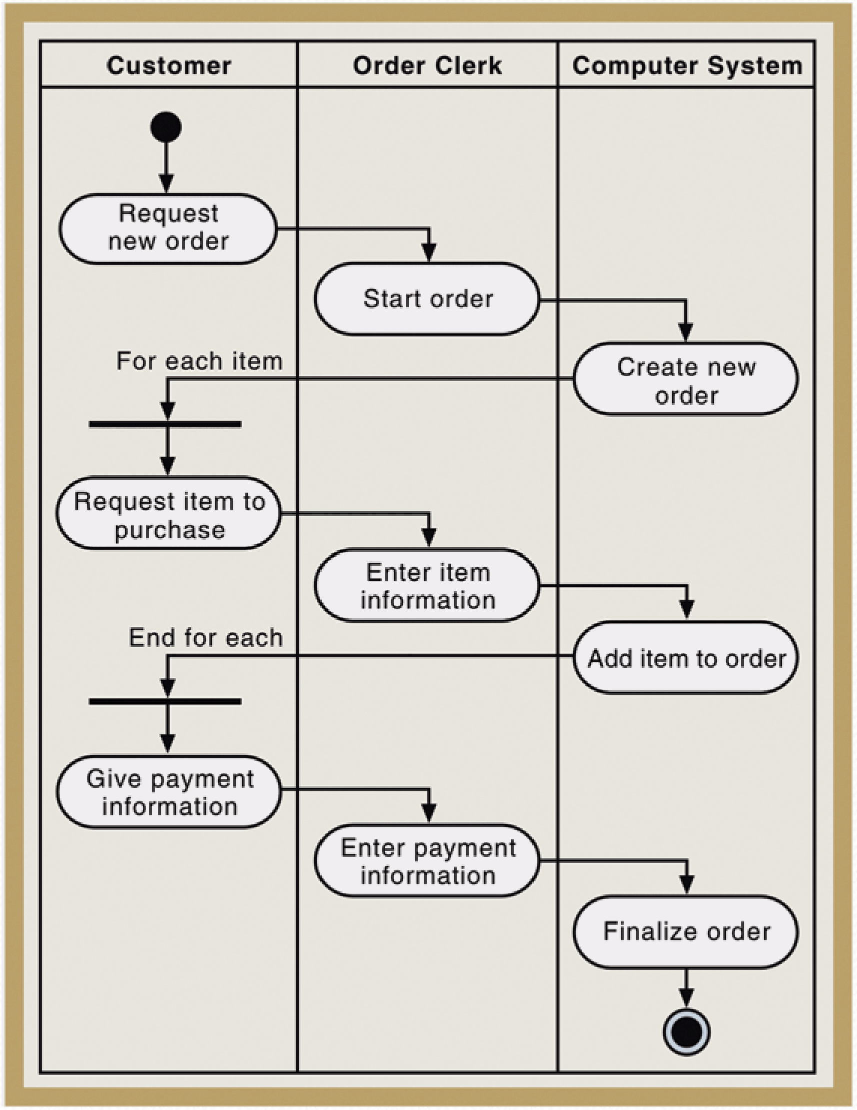
Flow of Activities:	Actor	System
	1. Sales clerk answers telephone and connects to a customer.	
	2. Clerk verifies customer information.	
	3. Clerk initiates the creation of a new order.	3.1 Create a new order.
	4. Customer requests an item be added to the order.	
	5. Clerk verifies the item (*Check item availability* use case).	5.1 Display item information.
	6. Clerk adds item to the order.	6.1 Add create an order item.
	7. Repeat steps 4, 5, and 6 until all items are added to the order.	
	8. Customer indicates end of order; clerk enters end of order.	8.1 Complete order.
		8.2 Compute totals.
	9. Customer submits payment; clerk enters amount.	9.1 Verify payment.
		9.2 Create order transaction.
		9.3 Finalize order.

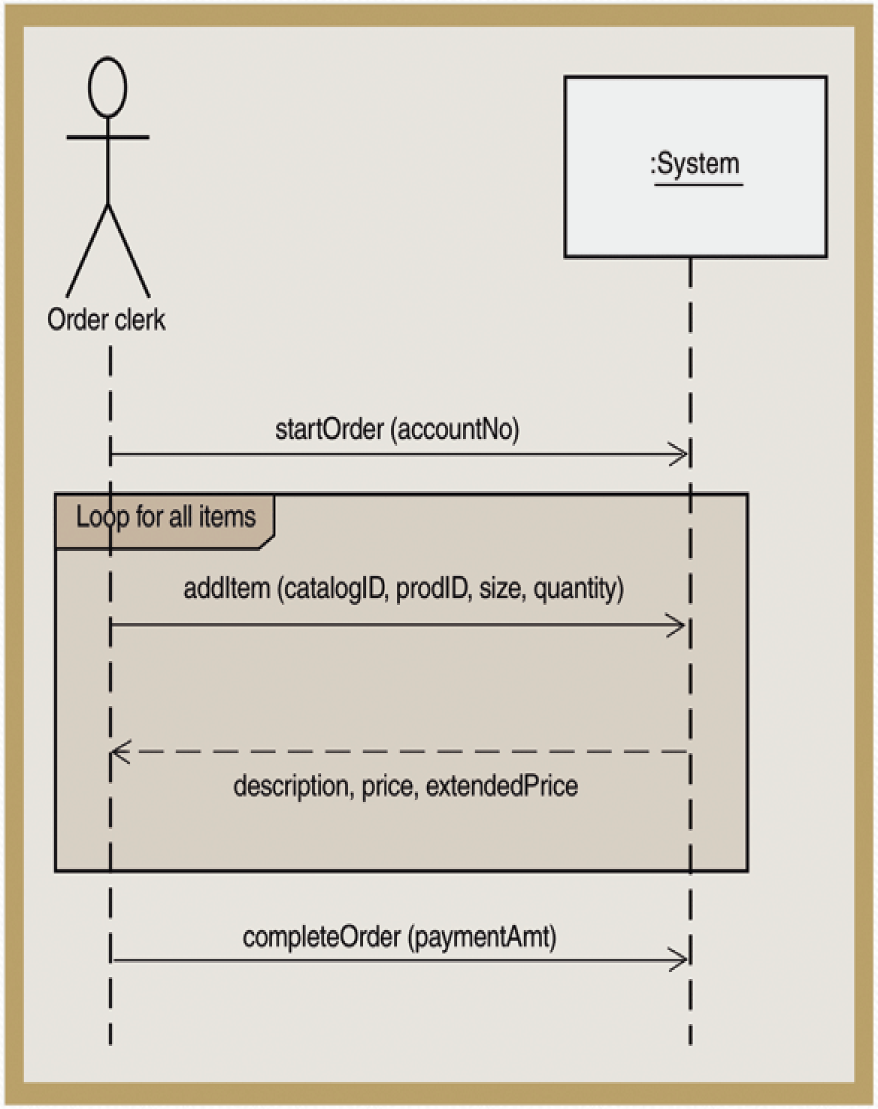
Bottom Detail from Use Case Description Telephone Order



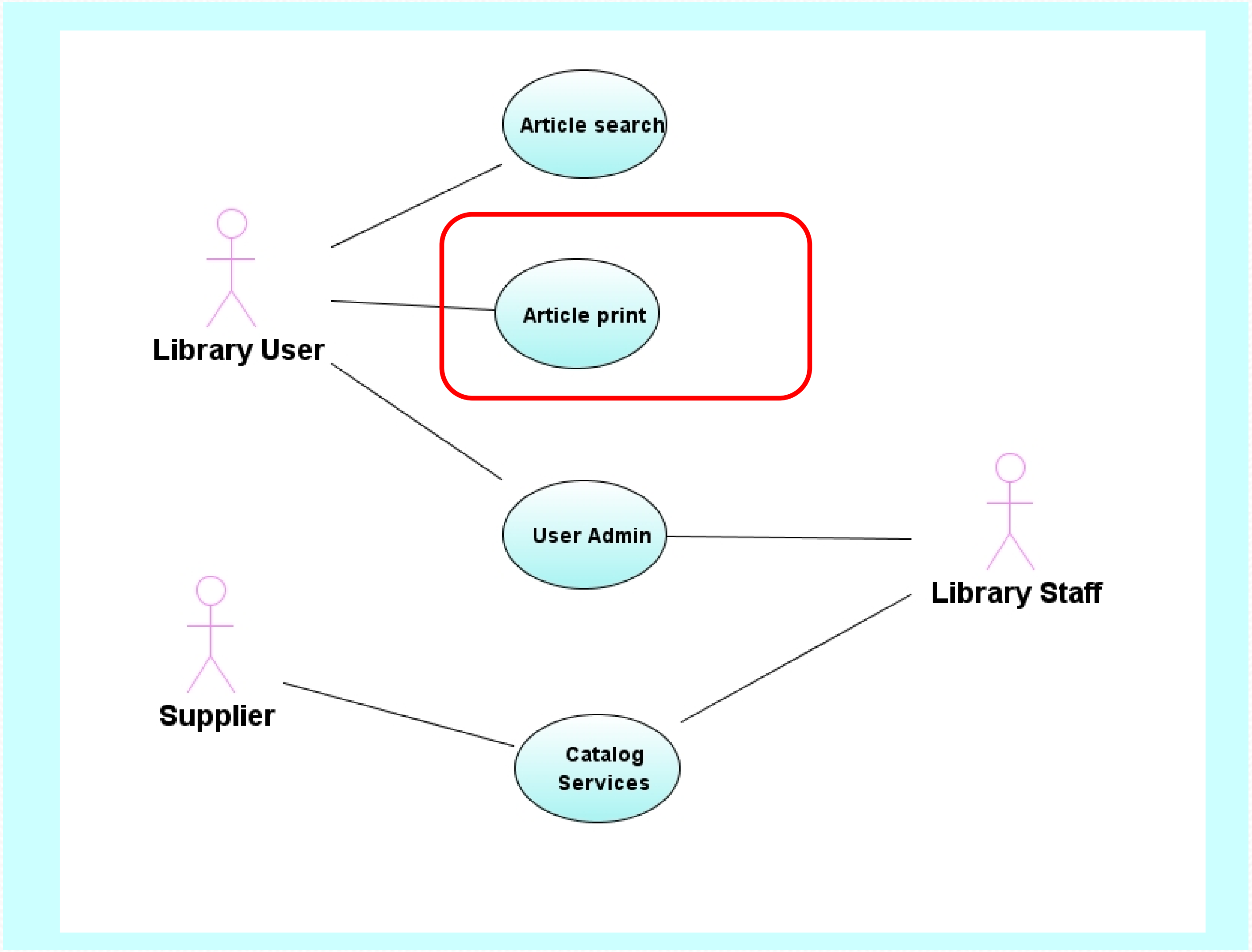
Exception Conditions:	2.1 If customer does not exist, then the clerk pauses this use case and invokes *Maintain customer information* use case. 2.2 If customer has a credit hold, then clerk transfers the customer to a customer service representative. 4.1 If an item is not in stock, then customer can a. choose not to purchase item, or b. request item be added as a back-ordered item. 9.1 If customer payment is rejected due to bad-credit verification, then a. order is canceled, or b. order is put on hold until check is received.

Activity Diagram and Resulting SSD for Telephone Order




Order clerk
:System
startOrder (accountNo)
Loop for all items
addItem (catalogID, prodID, size, quantity)
description, price, extendedPrice
completeOrder (paymentAmt)

Example - Library Use Case Diagram



Exercise 1 - Construct a System Sequence Diagram for the print article use case

- The activities flow for the Library User interacting with the print article use case:

  - The Library user requests the system to print article
  - The system prints the article
  - The Library user deletes the printed article from the system

Exercise 2 - Construct a System Sequence Diagram for the print article use case

- The activities flow for the print article user case:
  - The Library user requests the system to print article
  - The system sends the article to the system print queue
  - The system print queue sends the article to the printer
  - The printer prints the article
  - The printer updates the completed print status to the system
  - The Library user deletes printed article from the system