Student Online Teaching Advice Notice

The materials and content presented within this session are intended solely for use in a context of teaching and learning at Trinity.

Any session recorded for subsequent review is made available solely for the purpose of enhancing student learning.

Students should not edit or modify the recording in any way, nor disseminate it for use outside of a context of teaching and learning at Trinity.

Please be mindful of your physical environment and conscious of what may be captured by the device camera and microphone during videoconferencing calls.

Recorded materials will be handled in compliance with Trinity's statutory duties under the Universities Act, 1997 and in accordance with the University's policies and procedures.

Further information on data protection and best practice when using videoconferencing software is available at https://www.tcd.ie/info_compliance/data-protection/.

University of Dublin
Trinity College

Information Modeling Using The Unified Modelling Language (UML)

… the **art of communication** of the design of information..

University of Dublin
Trinity College

Interaction Diagrams:
1. UML Activity Diagram
2. UML Sequence Diagram

To model the **dynamic aspects** of the information system

University of Dublin
Trinity College

Interaction Diagram: UML Activity Diagram

Informally: Activity Diagram

Represents the workflow of the process

Describes how activities are coordinated.

Is particularly useful when you know that a process has to achieve **a number of different things**, and you want to model what the **essential dependencies** between them are, before you decide **in what order to do them**.

Records the dependencies between activities, such as which things can happen in parallel and what must be finished before something else can start.

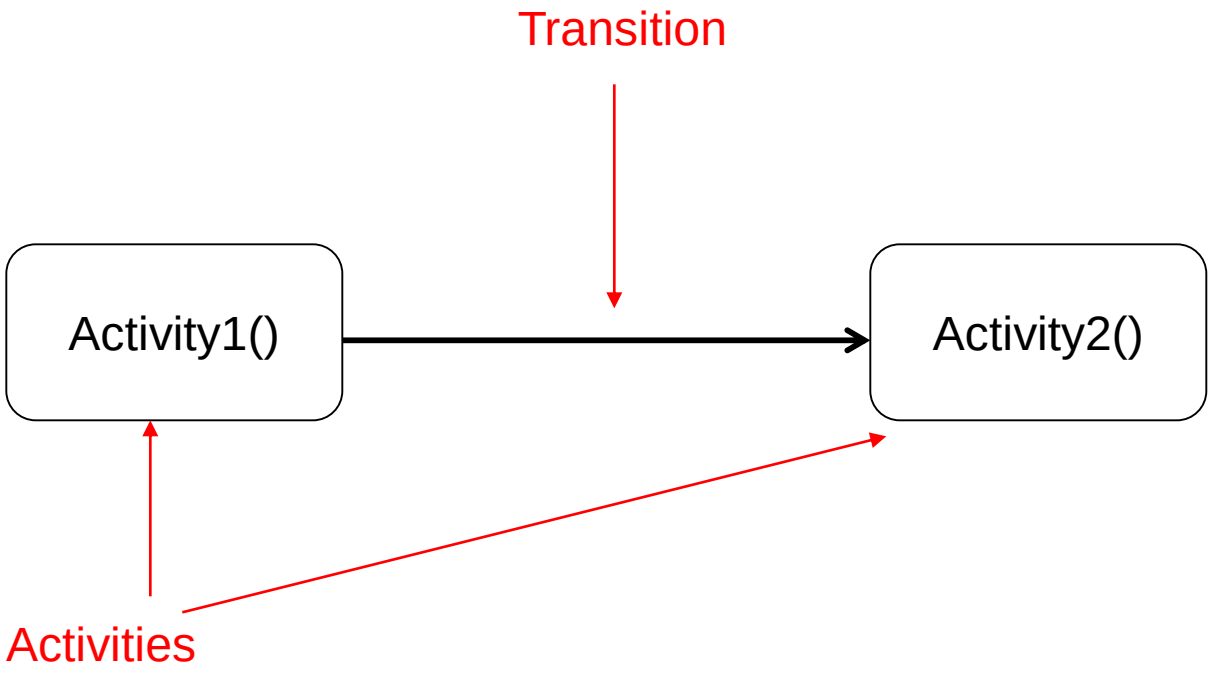
More Formally: Activity Diagram

An activity diagram is a graph of nodes and flows that shows the flow of control (and optionally data) through the steps of a computation.

- *Execution of steps can be both concurrent and sequential.*
- *An activity involves both synchronization and branching constructs, similar to but more powerful than a traditional flow chart, which only supports sequential and branching constructs.*

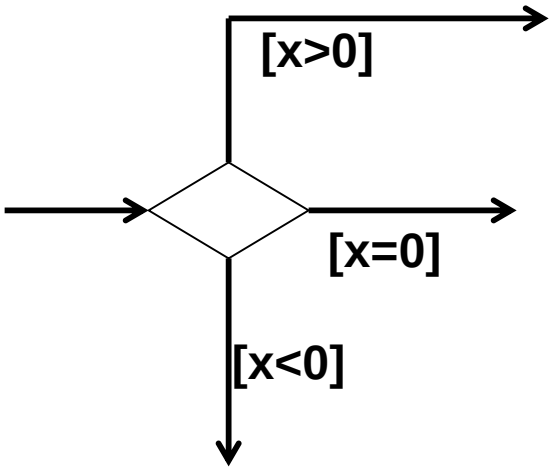
[The Unified Modeling Language Reference Manual, *James Rumbaugh, Ivar Jacobson, Grady Booch*,Addison Wesley]

Notation: Activities and Transitions



Three Features of Activity Diagrams

1. A transition may **branch** into two or more mutually exclusive transitions**.** **Guard expressions (inside [ ])** label the transitions coming out of a branch. A branch and its subsequent merge marking the end of the branch appear in the diagram as **hollow diamonds**.



The Branch

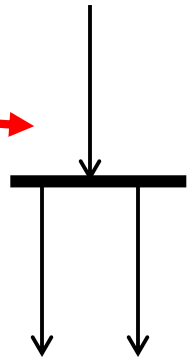
2. A transition may **fork** into two or more parallel activities. The fork and the subsequent **join** of the threads coming out of the fork appear in the diagram as **solid bars**.

3. Activity diagrams can be divided into object **swimlanes** that determine which object is responsible for which activity. A single transition comes out of each activity, connecting it to the next activity.

Fork and Join

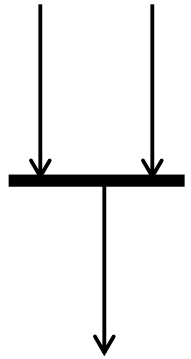
A fork may have one incoming transitions and two or more outgoing transitions

- each transition represents an independent flow of control
- conceptually, the activities of each of outgoing transitions are concurrent
  - *either truly concurrent*
  - *or sequential yet interleaved*

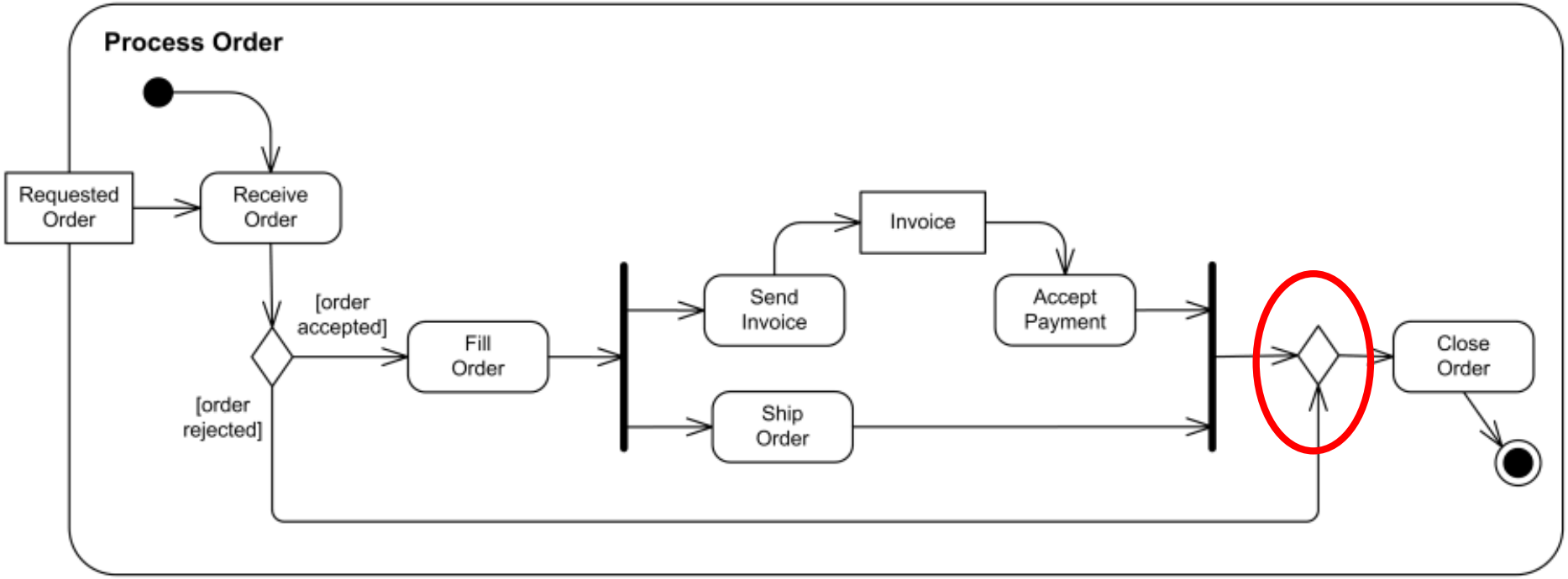


A join may have two or more incoming transitions and one outgoing transition

- above the join, the activities associated with each of these paths continues in parallel
- at the join, the concurrent flows synchronize
  - *each waits until all incoming flows have reached the join, at which point one flow of control continues on below the join*



Simple Example Activity Diagram : Processing an Order



CSSU2041: Use Case Exercise

From the statement below

1. Identify Actors, Use Cases and draw use case diagram
2. Write a textual description for the “withdraw money using a visa card” use case [where the visa customer is not a customer of the bank], (a) for a normal scenario and (b) for an error scenario

---

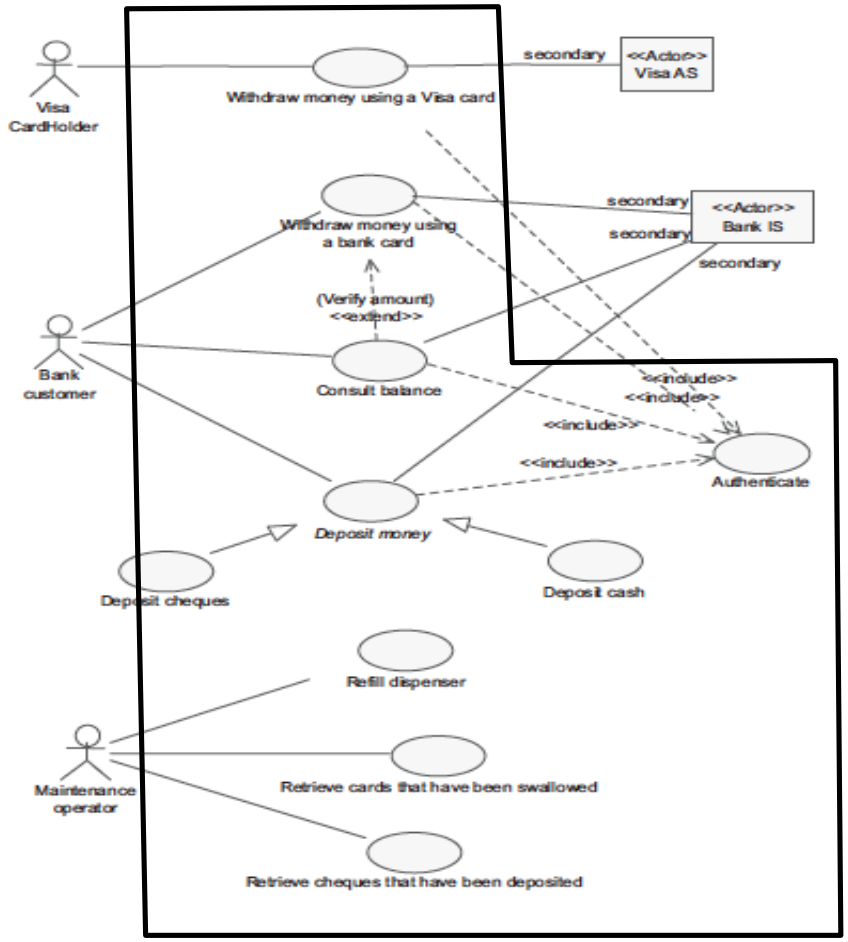
This case study concerns a simplified system of the automatic teller machine (ATM). The ATM offers the following services:

1. Distribution of money to every holder of a smartcard via a card reader and a cash dispenser.
2. Consultation of account balance, cash and cheque deposit facilities for bank customers who hold a smartcard from their bank.

Do not forget either that:

3. All transactions are made secure.
4. It is sometimes necessary to refill the dispenser, etc.

A possible solution ATM



This case study concerns a simplified system of the automatic teller machine (ATM). The ATM offers the following services:

1. Distribution of money to every holder of a smartcard via a card reader and a cash dispenser.
2. Consultation of account balance, cash and cheque deposit facilities for bank customers who hold a smartcard from their bank.

Do not forget either that:

3. All transactions are made secure.
4. It is sometimes necessary to refill the dispenser, etc.

What questions/clarifications would you ask for the next iteration of modelling?

Use Case Description

Title: Withdraw money using a Visa card

Summary: this use case allows a Visa card holder, who is not a customer of the bank, to withdraw money if his or her daily limit allows it.

Actors: Visa CardHolder (primary), Visa AS (secondary).

Creation date: 02/03/02 **Date of update:** 08/19/03

Version: 2.2 **Person in charge:** Pascal Roques

Flow of events

Preconditions:

- The ATM cash box is well stocked.
- There is no card in the reader.

Use Case Description: Normal Scenario

1. The Visa CardHolder inserts his or her card in the ATM's card reader.
2. The ATM verifies that the card that has been inserted is indeed a Visa card.
3. The ATM asks the Visa CardHolder to enter his or her pin number.
4. The Visa CardHolder enters his or her pin number.
5. The ATM compares the pin number with the one that is encoded on the chip of the card.
6. The ATM requests an authorisation from the VISA authorisation system.
7. The VISA authorisation system confirms its agreement and indicates the daily balance.
8. The ATM asks the Visa CardHolder to enter the desired withdrawal amount.
9. The Visa CardHolder enters the desired withdrawal amount.
10. The ATM checks the desired amount against the daily balance.
11. The ATM asks the Visa CardHolder if he or she would like a receipt.
12. The Visa CardHolder requests a receipt.
13. The ATM returns the card to the Visa CardHolder.
14. The Visa CardHolder takes his or her card.
15. The ATM issues the notes and a receipt.
16. The Visa CardHolder takes the notes and the receipt.

What questions/clarifications would you ask for the next iteration of modelling?

Use Case Description: Error Scenario

Error sequences:

E1: invalid card
The E1 sequence starts at point 2 of the main success scenario.

3. The ATM informs the Visa CardHolder that the smartcard is not valid (unreadable, expired, etc.) and confiscates it; the use case fails.

E2: conclusively incorrect pin number
The E2 sequence starts at point 5 of the main success scenario.

6. The ATM informs the Visa CardHolder that the pin is incorrect for the third time.
7. The ATM confiscates the smartcard.
8. The VISA authorisation system is notified; the use case fails.

E3: unauthorised withdrawal
The E3 sequence starts at point 6 of the main success scenario.

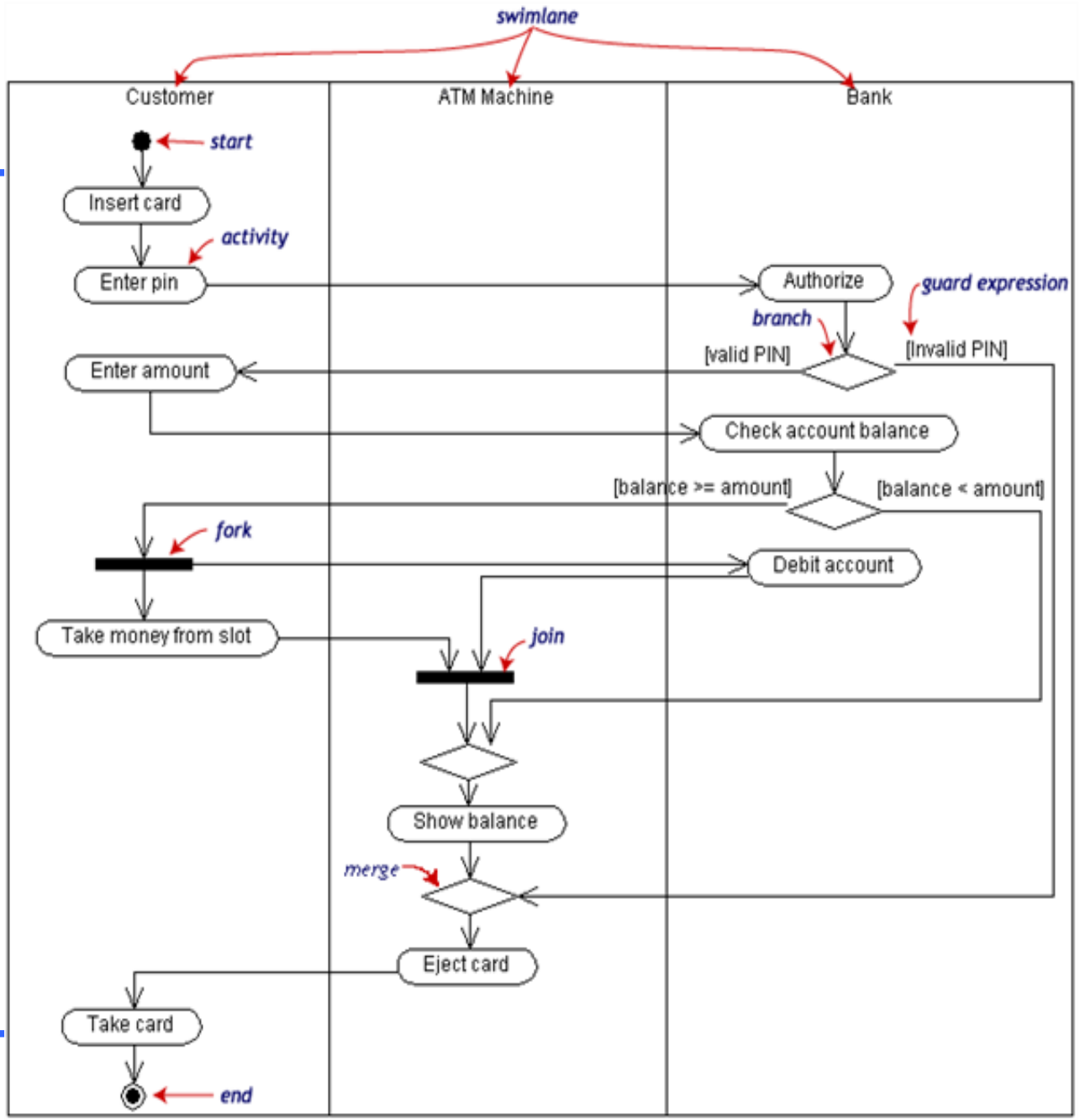
7. The VISA authorisation system forbids any withdrawal.
8. The ATM ejects the smartcard; the use case fails.

What questions/clarifications would you ask for the next iteration of modelling?

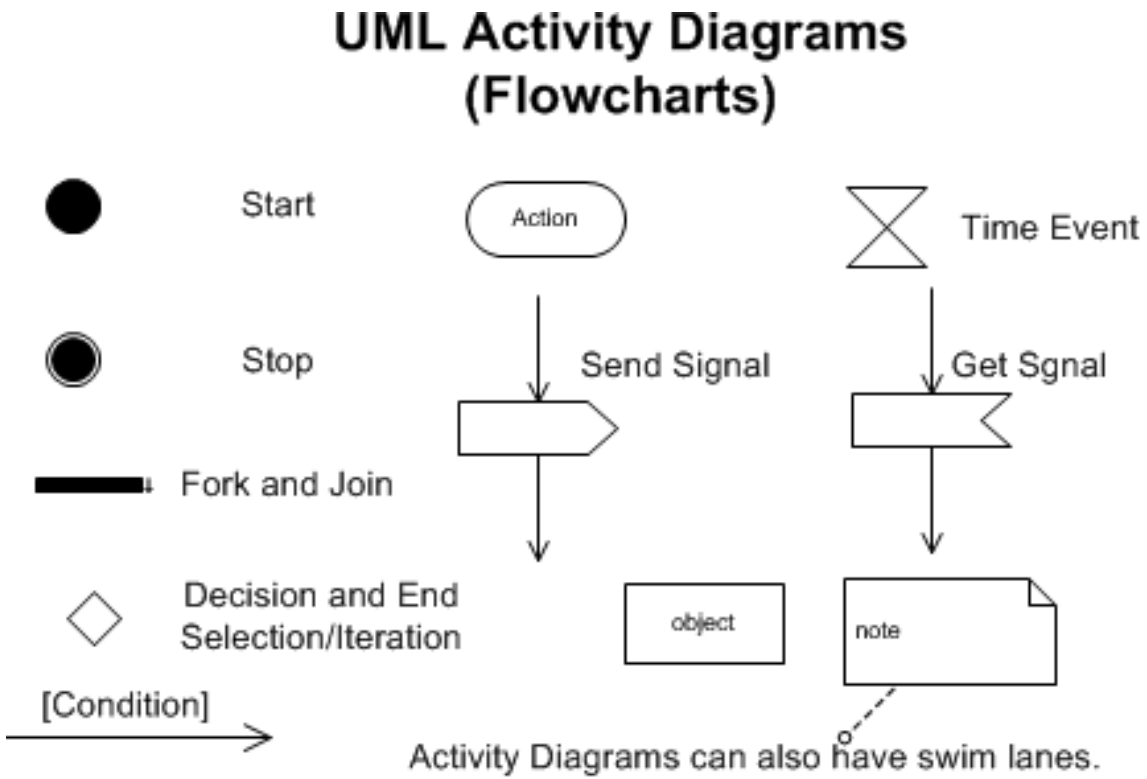
What questions/clarifications would you ask for the next iteration of modelling?

Example Activity Diagram

"Withdraw money from a bank account through an ATM.



Additional Activity Diagram Notation



University of Dublin
Trinity College

Interaction Diagram: UML Sequence Diagram

Sequence Diagram

A sequence diagram is an interaction diagram that details how **operations** are carried out -- what messages are sent and when.

- Sequence diagrams are organized according to time.
- The time progresses as you go down the page.
- The objects/Actors involved in the operation are listed from left to right according to when they take part in the message sequence.

Draw Sequence Diagram for "Process Sale"

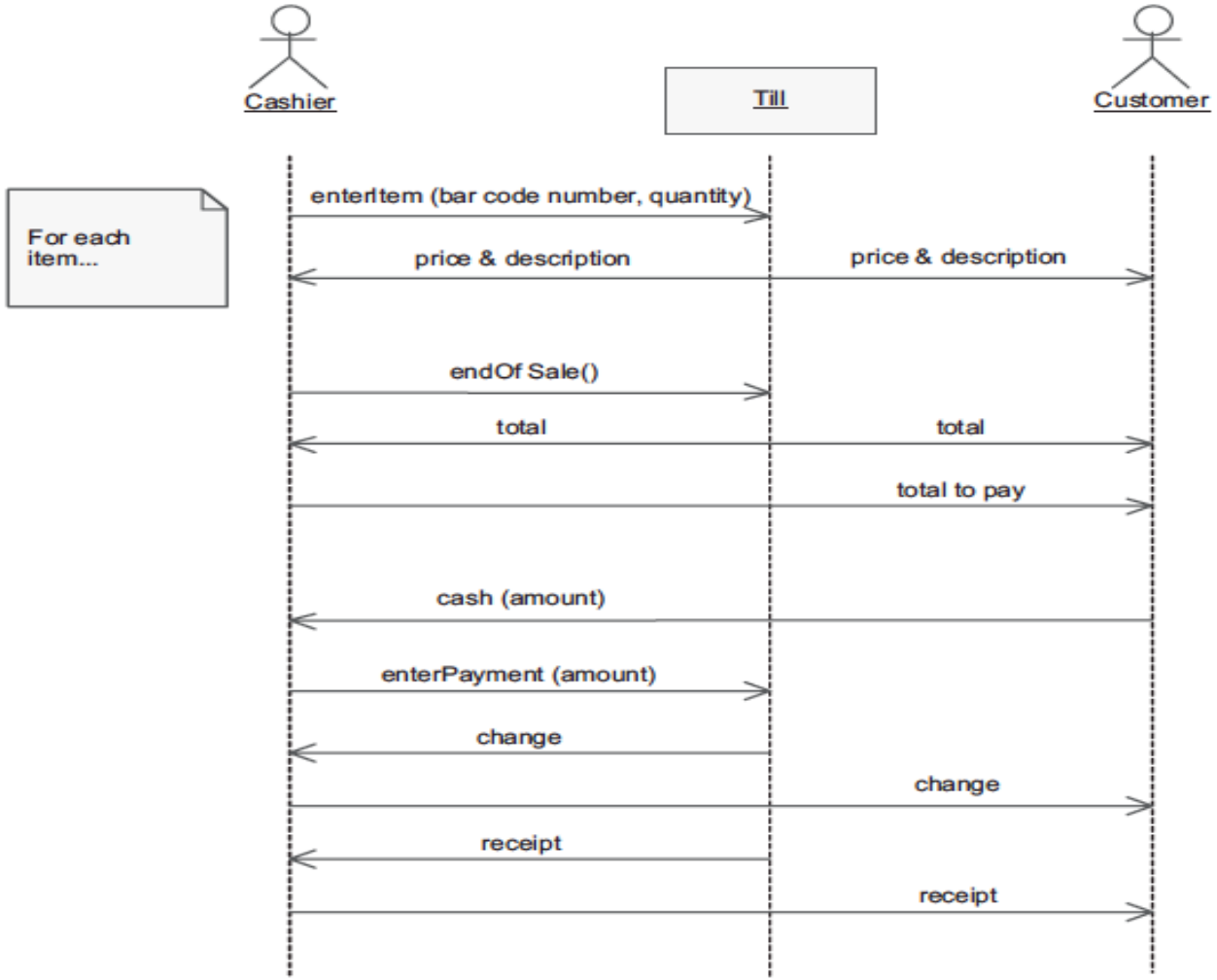
Preconditions:

- The cash register is open; a checkout assistant is signed on to it.

Main success scenario:

1. This use case starts when a customer arrives at the checkout with items that he or she would like to purchase.
2. The cashier records each item. If there is more than one of the same item, the cashier also indicates the quantity.
3. The cash register establishes the price of the item and adds the information on the item to the sale in progress. The cash register displays the description and the price of the item in question.
4. Once the cashier has recorded all the items, he or she indicates that the sale is finished.
5. The cash register calculates and displays the total amount of the sale.
6. The cashier informs the customer of the total amount.
7. The customer chooses a payment method:
   a. In the case of cash payment, execute the "Process cash payment" use case;
   b. In the case of credit card payment, execute the "Process credit card payment" use case;
   c. In the case of cheque payment, execute the "Process cheque payment" use case.
8. The cash register records the sale that has been carried out and prints a receipt.
9. The cashier gives the cash register receipt to the customer.
10. The customer leaves with the items he or she has purchased.

Sequence Diagram: Process Sale (Cash)



Features of a Sequence Diagram

Each vertical dotted line is a lifeline, representing the time that an object exists.

Each arrow is a message call. An arrow goes from the sender to the top of the activation bar(if present) of the message on the receiver's lifeline.

The activation bar (if present) represents the duration of execution of the message.

The expression in square brackets, [ ], is a condition,

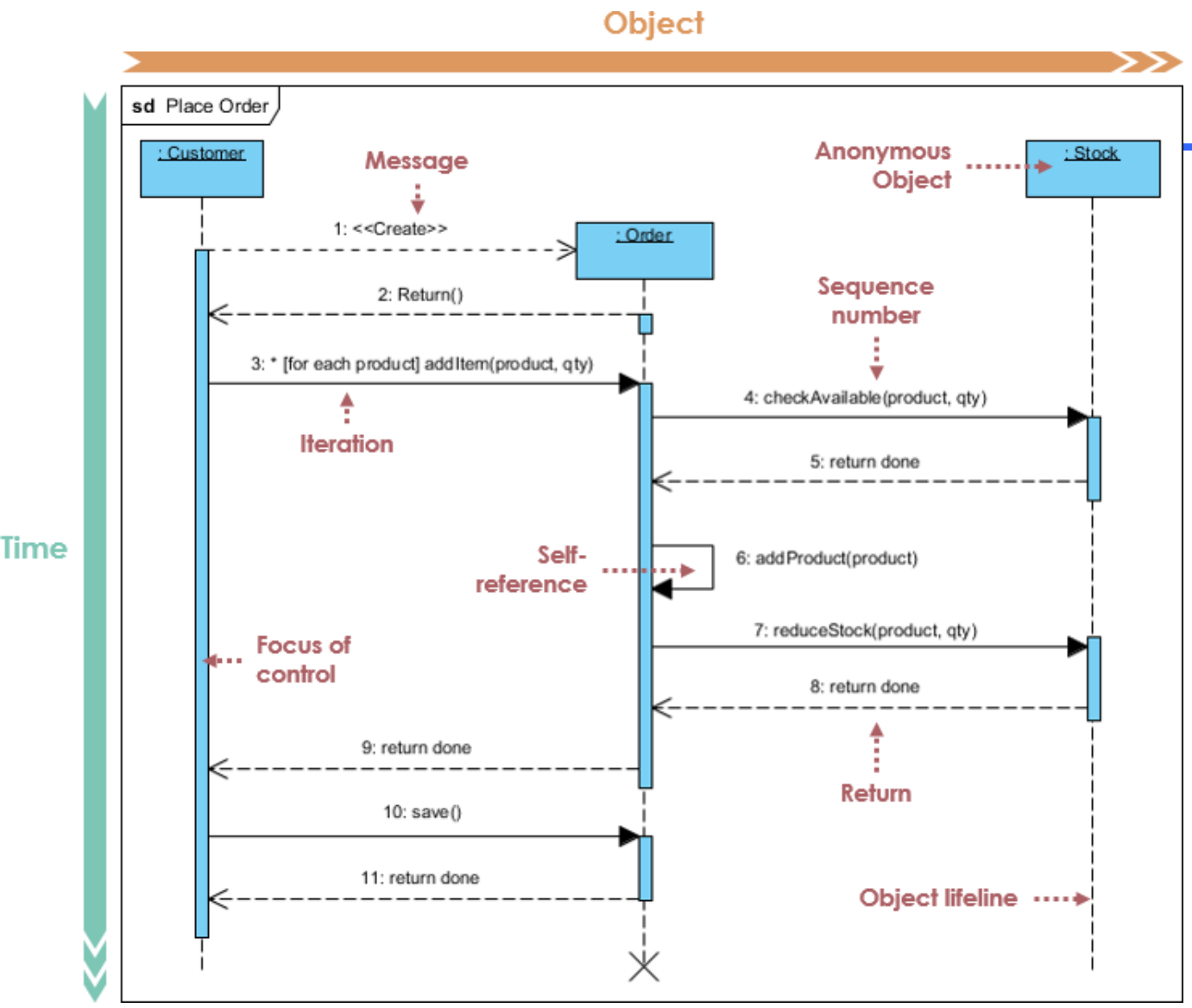
The diagram has a clarifying note, which is text inside a dog-eared rectangle. Notes can be put into any kind of UML diagram.

Return Values

Optionally indicated using a dashed arrow with a label indicating the return value.

- Don’t model a return value when it is obvious what is being returned, e.g. getTotal()

- Good Practice- Model a return value only when you need to refer to it elsewhere, e.g. as a parameter passed in another message.



Step 1 and 2: Customer creates an order.

Step 3: Customer add items to the order.

Step 4, 5: Each item is checked for availability in inventory.

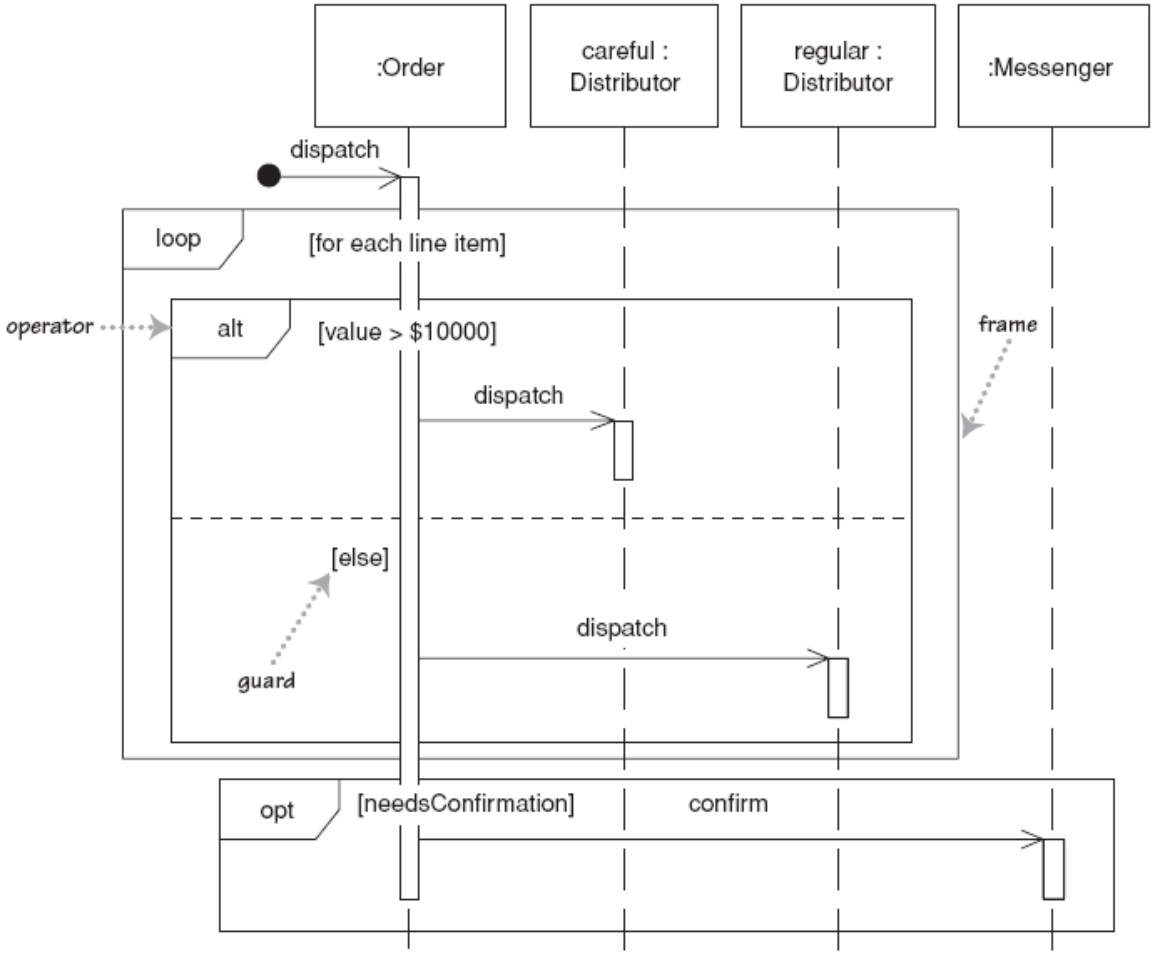
Step 6, 7, 8 : If the product is available, it is added to the order.

Step 9 return

Step 10, 11: save and destroy order

https://www.visual-paradigm.com/learning/handbooks/software-design-handbook/sequence-diagram.jsp

Indicating selection and loops



frame: box around part of a sequence diagram to indicate a sequence fragment

loop- [condition or items to loop over]

alt – if/else

[condition], separated by horiz. dashed line

opt- if [condition]

List of Sequence Fragments

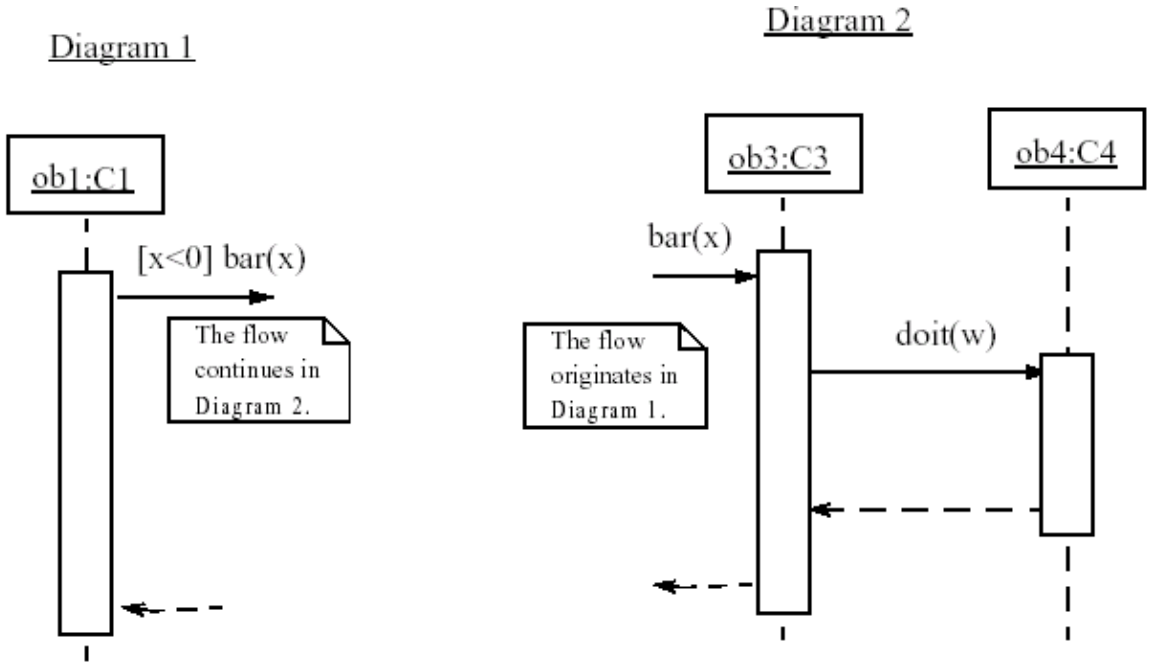
Operator	Meaning
alt	**Alternative multiple fragments**: only the one whose condition is true will execute.
opt	**Optional**: the fragment executes only if the supplied condition is true. Equivalent to an alt only with one trace.
par	**Parallel**: each fragment is run in parallel.
loop	**Loop**: the fragment may execute multiple times, and the guard indicates the basis of iteration.
critical	**Critical region**: the fragment can have only one thread executing it at once.
neg	**Negative**: the fragment shows an invalid interaction.
ref	**Reference**: refers to an interaction defined on another diagram. The frame is drawn to cover the lifelines involved in the interaction. You can define parameters and a return value.
sd	**Sequence diagram**: used to surround an entire sequence diagram.

https://www.visual-paradigm.com/learning/handbooks/software-design-handbook/sequence-diagram.jsp

linking sequence diagrams

if one sequence diagram is too large or refers to another diagram, This can be indicated with ..

- an unfinished arrow and note/comment box



That's All Folks Thank You for Listening