# CH4.Activity Diagram

School of Computer Science
Prof. Euijong Lee

Objectives

❖ **Understand the purpose of an activity diagram**

❖ **Understand components of an activity diagram**

❖ **Understand the development processes of an activity diagram**
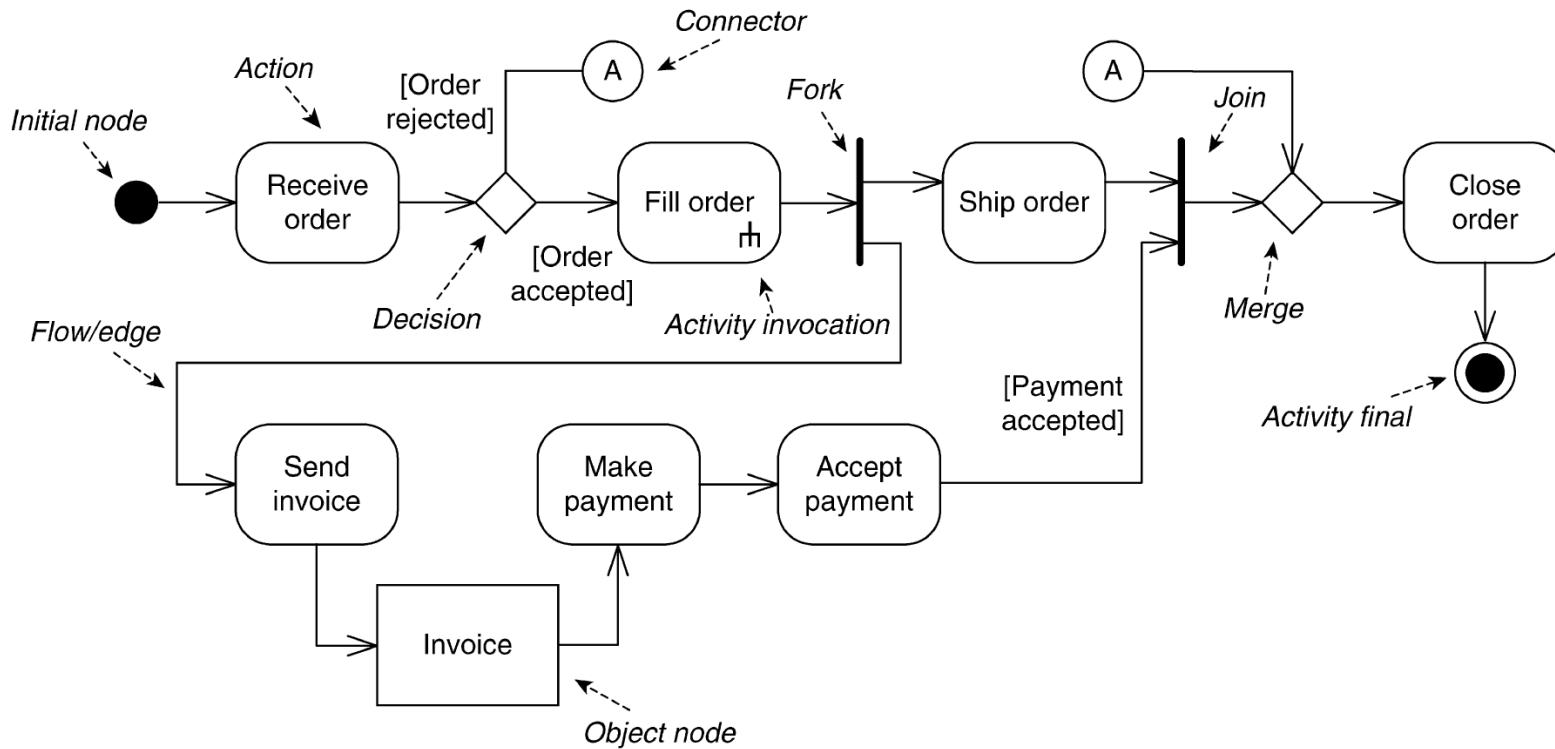
BPM With Activity Diagram

❖ **A number of activities support a business process across several departments**
- (Business) process model: depicts how a business system operates
- Process or activity that are performed, and how object(data) move among them
- Current system(as-is) and new system(to-be) are both included
- Do not consider whether computerized or not

❖ **Activity diagram**
- High-level behavior model in a business process
- A Logical model, problem domain model
- Sophisticated data flow diagram
- Addresses parallel concurrent activities and complex

Activity diagrams

❖ **Useful to specify software or hardware system behavior**

❖ **Based on data flow models – a graphical representation (with a Directed Graph) of how data move around an information system**
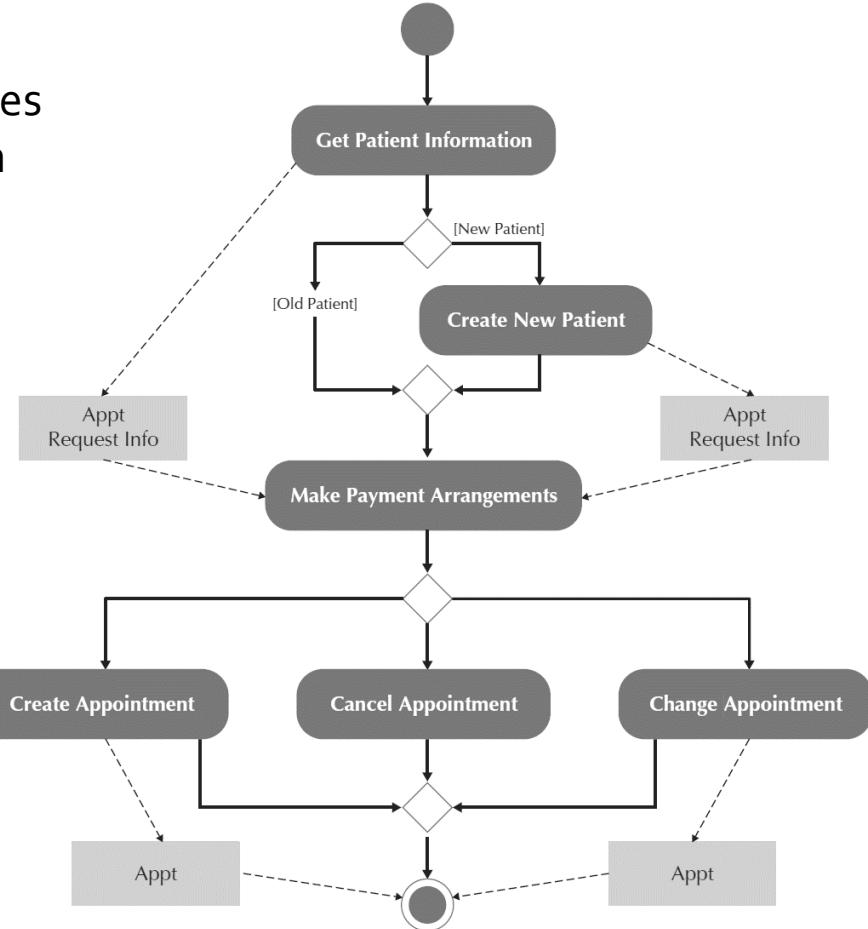
Activity Diagram Example
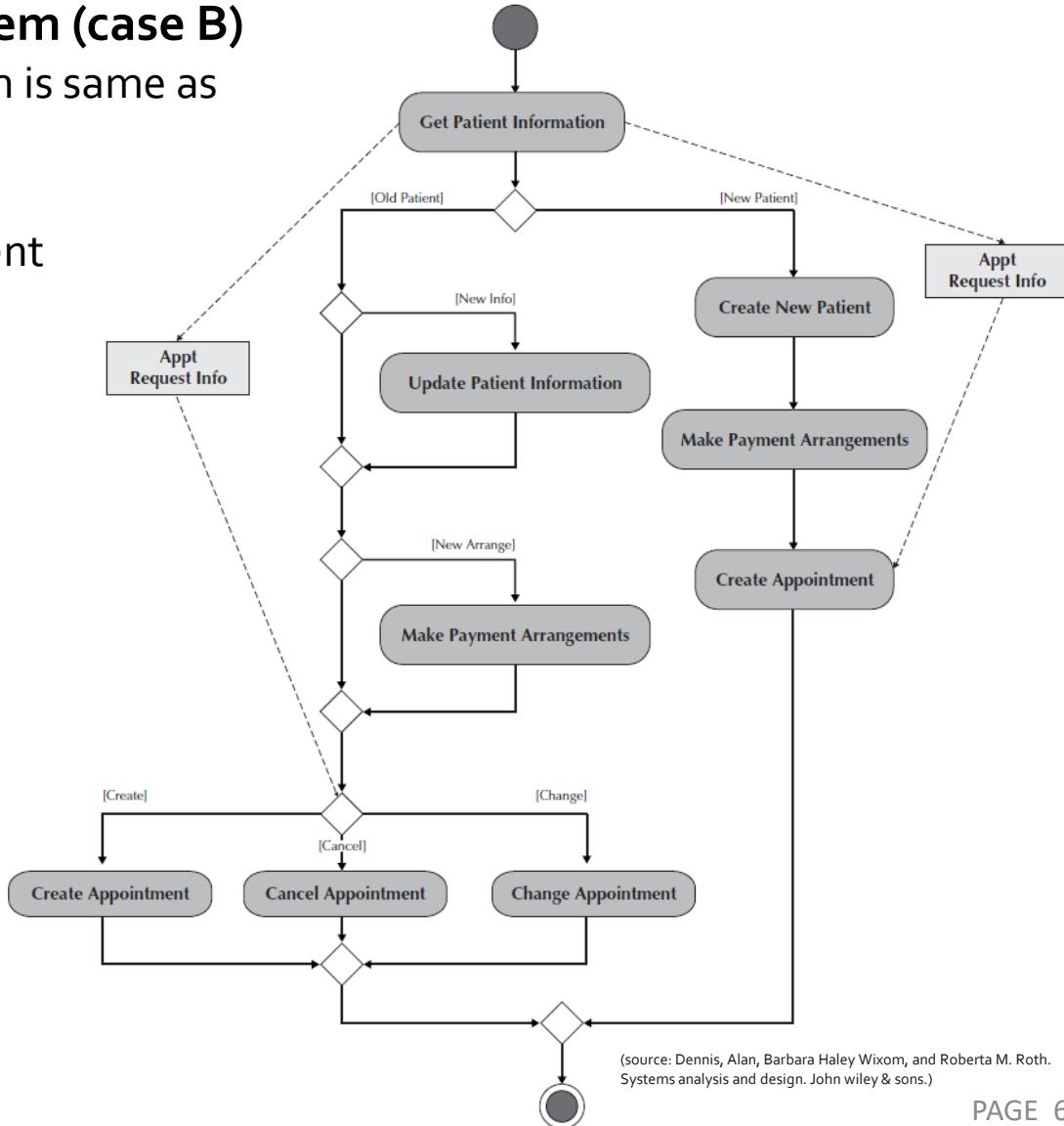
❖ **Medical Appointment System (case A)**
  - Appointment system in hospital

  - The diagram shows flow of activities to make an appointment between patient and hospital

Activity Diagram Example

❖ **Medical Appointment System (case B)**
- The purpose of the diagram is same as the previous

- But, composed with different activities and transitions

(source: Dennis, Alan, Barbara Haley Wixom, and Roberta M. Roth. Systems analysis and design. John wiley & sons.)

Elements of Activity Diagram (1/4)

❖ **Name, Meaning and its Symbol**

|	
An action: ■ Is a simple, nondecomposable piece of behavior. ■ Is labeled by its name.	Action
An activity: ■ Is used to represent a set of actions. ■ Is labeled by its name.	Activity
An object node: ■ Is used to represent an object that is connected to a set of object flows. ■ Is labeled by its class name.	Class Name

Object

=   ,
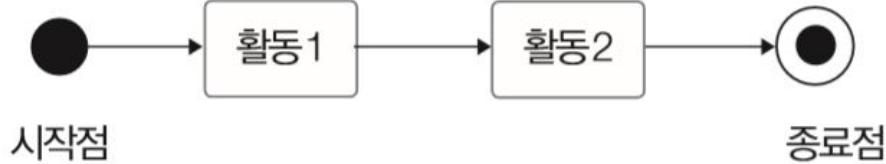
Elements of Activity Diagram (2/4)

❖ **Name, Meaning and its Symbol**
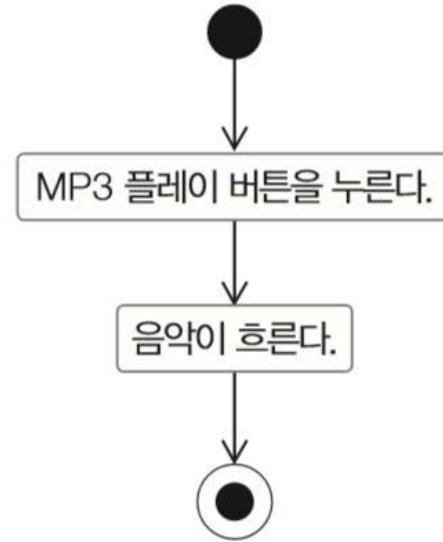
A control flow: ■ Shows the sequence of execution.	⟶
An object flow: ■ Shows the flow of an object from one activity (or action) to another activity (or action).	⇢
An initial node: ■ Portrays the beginning of a set of actions or activities.	●
A final-activity node: ■ Is used to stop all control flows and object flows in an activity (or action).	◉

Elements of Activity Example(1)

❖ **Example of activity diagram**

<Basic structure of activity diagram>

<Example of activity diagram with initial and final-activity nodes>

CHUNGBUK NATIONAL UNIVERSITY
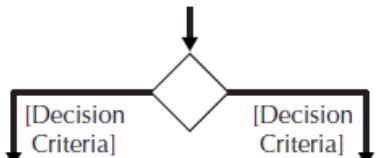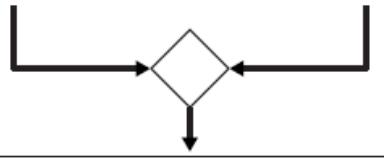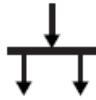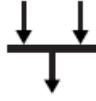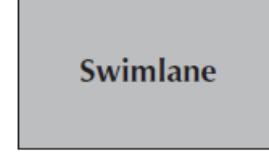
Elements of Activity Diagram (3/4)

❖ **Name, Meaning and its Symbol**

A final-flow node: ■ Is used to stop a specific control flow or object flow.	
A decision node: ( ) ■ Is used to represent a test condition to ensure that the control flow or object flow only goes down one path. ■ Is labeled with the decision criteria to continue down the specific path.	
A merge node: ■ Is used to bring back together different decision paths that were created using a decision node.	

Elements of Activity Diagram (4/4)

❖ **Name, Meaning and its Symbol**
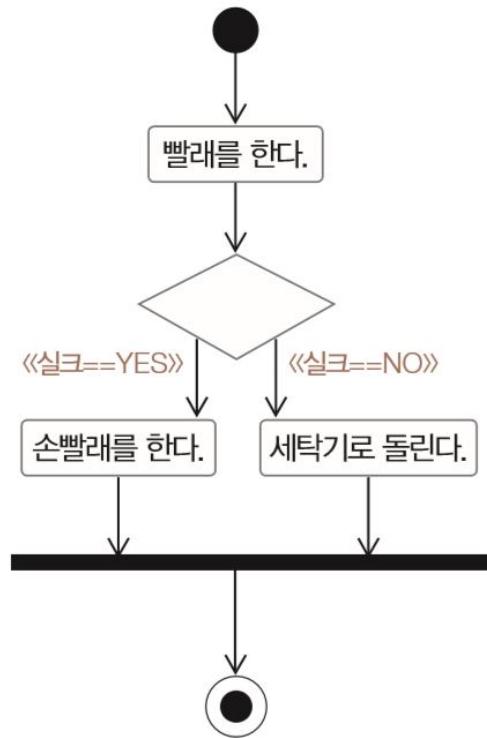
A fork node: 2 Is used to split behavior into a set of parallel or concurrent flows of activities (or actions)	
A join node: Is used to bring back together a set of parallel or concurrent flows of activities (or actions)	
A swimlane: Is used to break up an activity diagram into rows and columns to assign the individual activities (or actions) to the individuals or objects that are responsible for executing the activity (or action) Is labeled with the name of the individual or object responsible	Swimlane

or
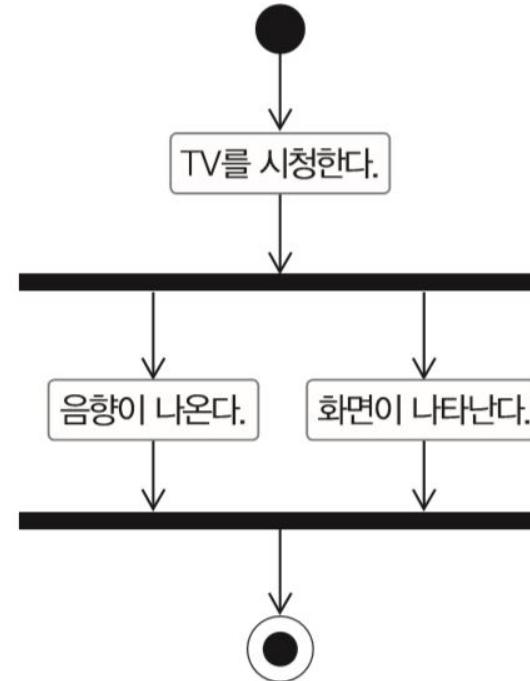
Elements of Activity Example(2)

❖ **Example of activity diagram**

<An example of join node>

<An example of synchronization>
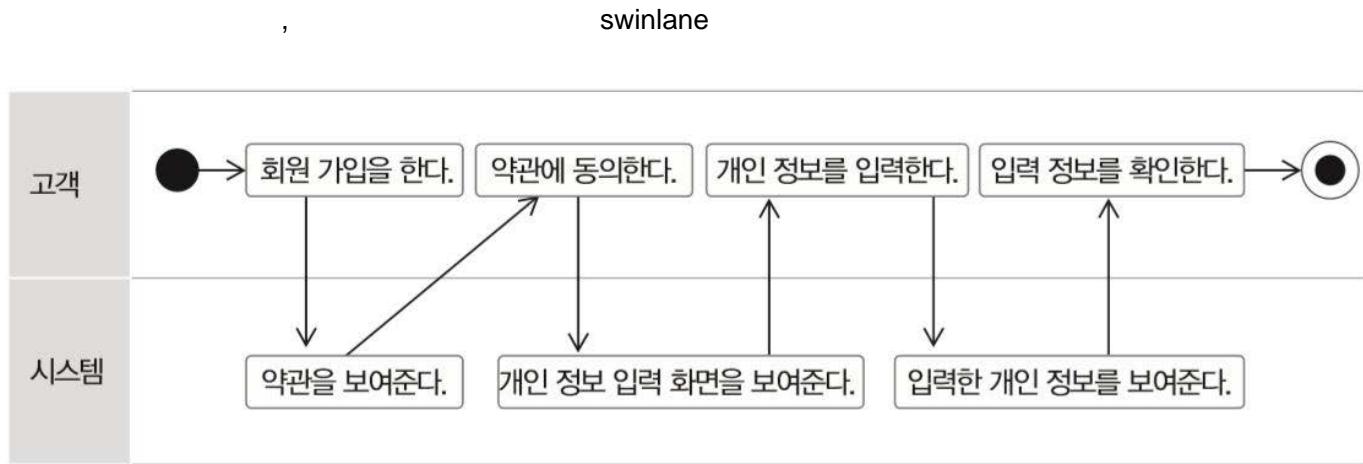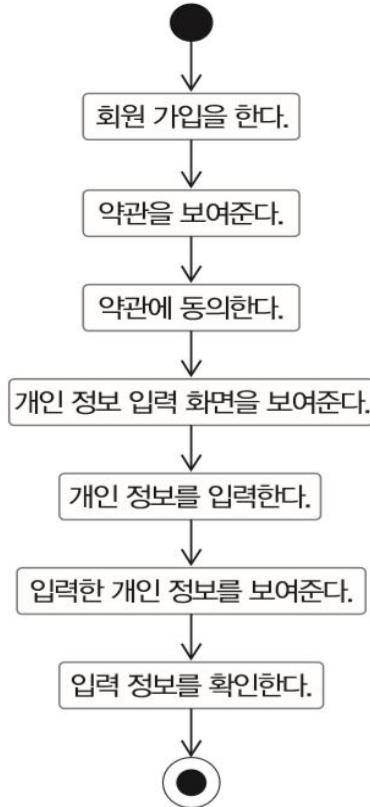
Elements of Activity Example(3)

❖ **Example of activity diagram**

,   swinlane

<Example with swimlane>

(source: UML 기초와 응용. 한빛 아카데미.)

(a) 구획면이 없는 회원 가입 과정

<Example without a swimlane>

CHUNGBUK NATIONAL UNIVERSITY

More about Elements of Activity Diagram

- ❖ **Action**
- ❖ **Pins**
- ❖ **Activity**
  - Activity Nodes
  - Activity Edges
- ❖ **Control Nodes**
- ❖ **Signals**
- ❖ **Interruptible Region**
- ❖ **Exception**
- ❖ **Swimlanes**
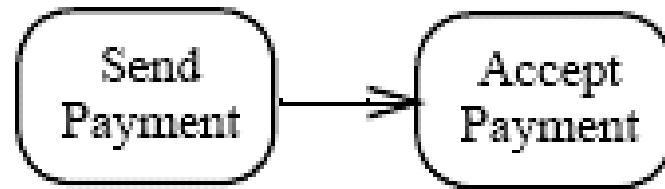
Action

❖ **The fundamental unit of executable functionality in an activity**

❖ **The execution of an action represents some transformations or processes in the modeled system (creating objects, setting attribute values, linking objects together, invoking user-defined behaviors, etc.)**

<Example of Action node>
(Name: description of the action behaviour)

<Example of Actions and their connection>

CHUNGBUK NATIONAL UNIVERSITY
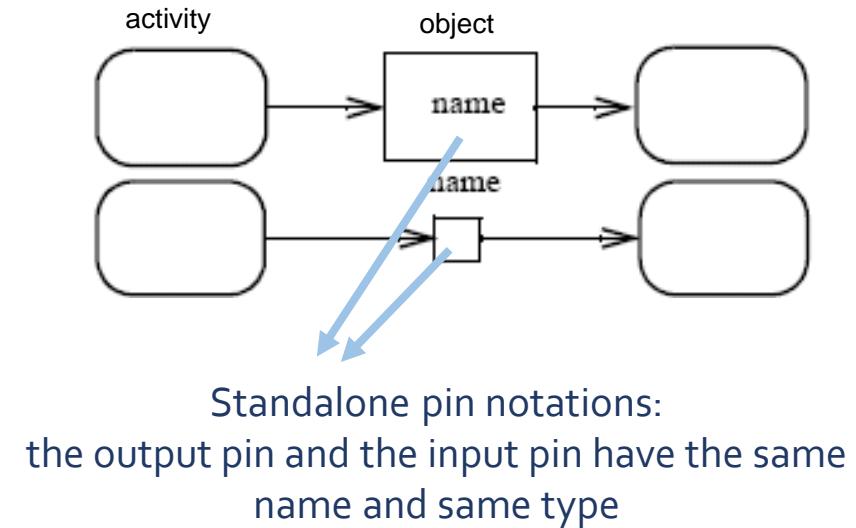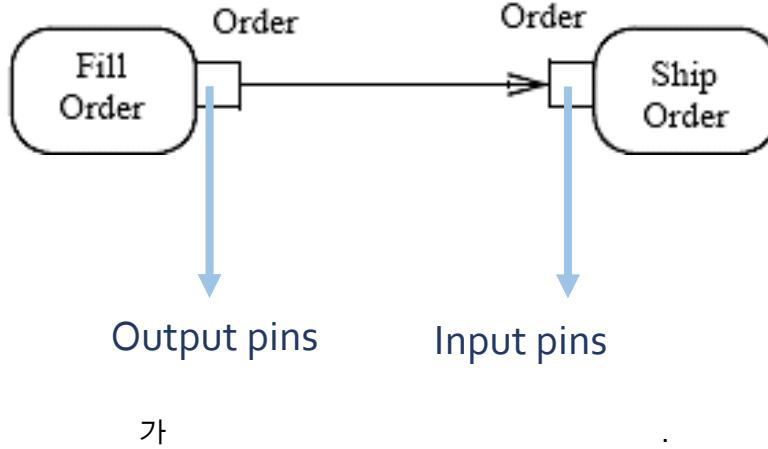
Pins

❖ **Actions can have inputs and outputs, through the pins**

❖ **Hold inputs to actions until the action starts, and hold the outputs of actions before the values move downstream**

❖ **The name of a pin is not restricted: generally recalls the type of objects or data that flow through the pin**

activity   object

Order   Order

Fill Order   Ship Order

name

name

Output pins   Input pins

Standalone pin notations:
the output pin and the input pin have the same name and same type
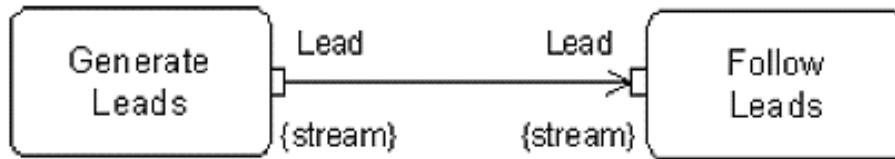
(Source: Debenedetti Emanuele, Activity diagrams in UML 2.0)

Special kind of pins
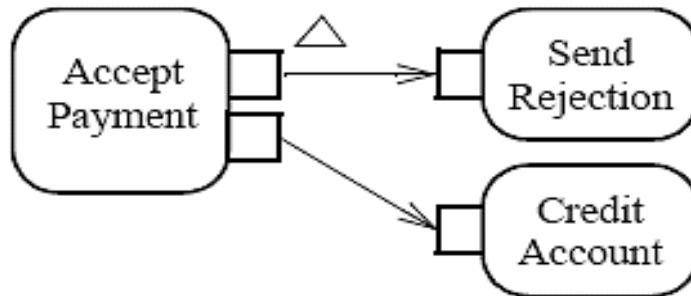
❖ **Streaming Parameters (notated with {STREAM})**
  - Accept or provide one or more values while an action is executing

❖ **Exception Output Parameters (notated with a triangle)**
  - Provide values to the exclusion of any other output parameter or outgoing control of the action
  - The action must immediately terminate, and output cannot quit
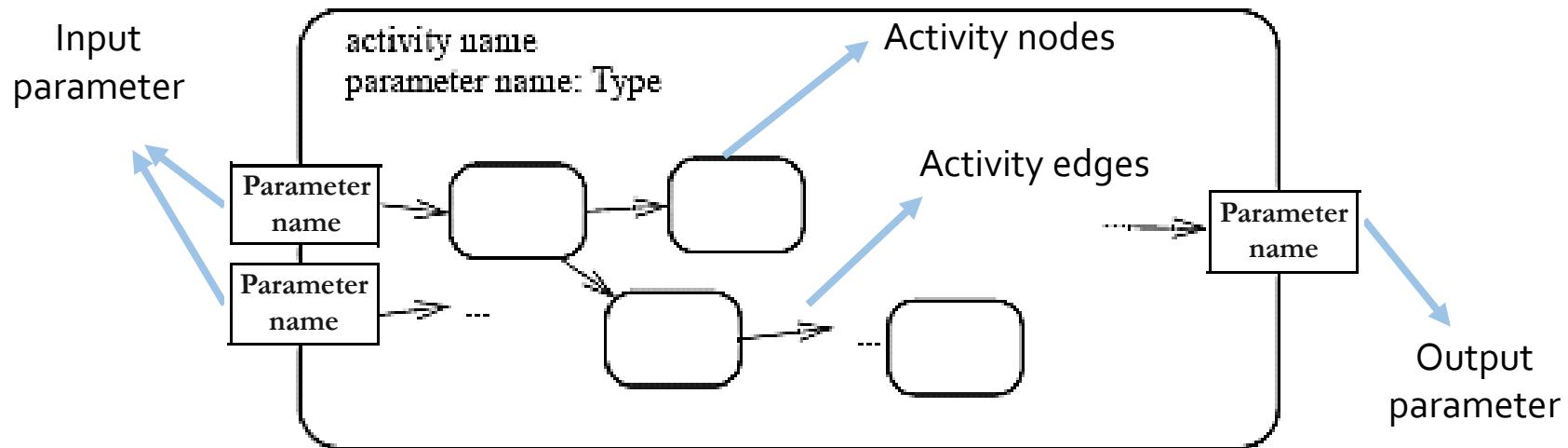
(Source: Debenedetti Emanuele, Activity diagrams in UML 2.0)

Activities

❖ **An activity is the specification of parameterized behaviour as the coordinated sequencing of subordinate units whose individual elements are action**

❖ **Uses parameters to receive and provide data to the invoker**

Input parameter

activity name
parameter name: Type

Activity nodes

Activity edges

Parameter name

Parameter name

Parameter name

Output parameter

(Source: Debenedetti Emanuele, Activity diagrams in UML 2.0)
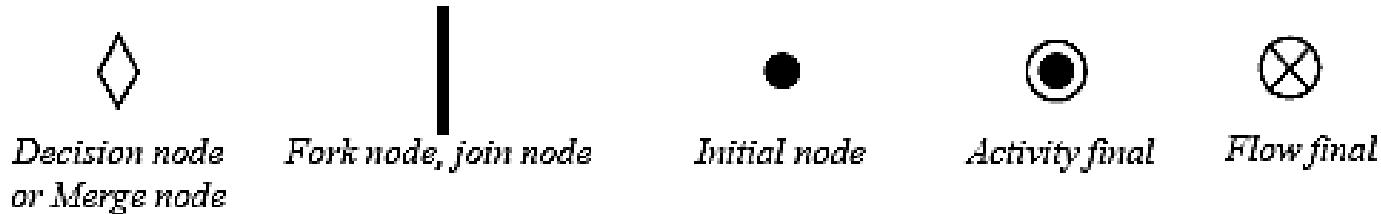
Activities nodes

❖ **Three type of activity nodes:**

- **Action nodes :**   name

- **Control nodes :** coordinate flows in an activity diagram between other nodes

◇ | ●   ◉   ⊗

Decision node   Fork node, join node   Initial node   Activity final   Flow final
or Merge node
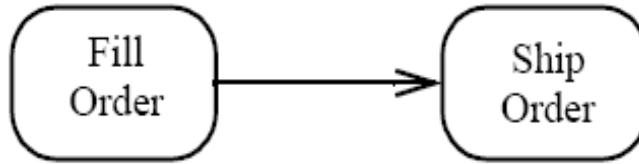
- **Object nodes :** indicate an instance of a particular object, may be available at a particular point in the activity(i. e Pins are object nodes)

label   label □
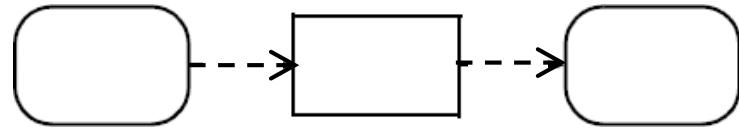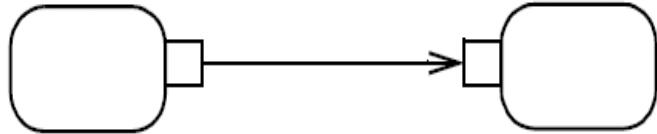
Activities edges

❖ **Directed connections**

❖ **Two kinds of edges :**
- **Control flow nodes** : is an edge which starts an activity node after the completion of the previous one by passing a control token

- **Object flow nodes** : models the flow of values to or from object node by passing object or data tokens

(Source: Debenedetti Emanuele, Activity diagrams in UML 2.0)

Control nodes – initial nodes

❖ **In an activity, the flow starts at initial nodes**
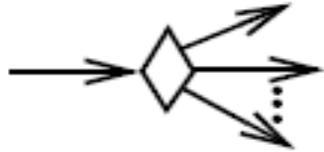
❖ **If there are more then one initial node**
- A control is placed in each initial node when the activity is started, initating multiple flows

❖ **If an initial node has more than one outgoing edge**
- Only one of these edges will receive control

A or B ?
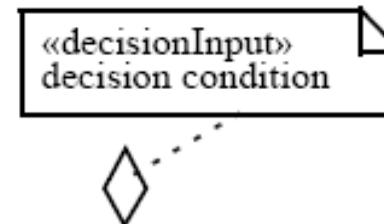
Control nodes – decision nodes
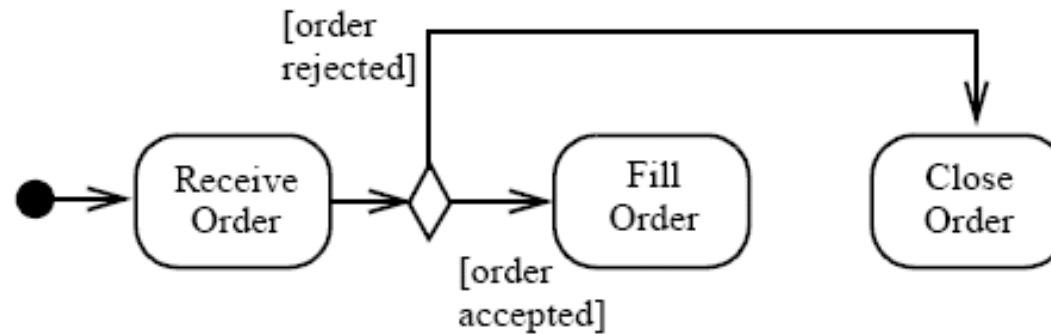
❖ **Route the flow to one of the outgoing edges**
❖ **Guards are specified on the outgoing edges**

<Decision node with multiple branches>

«decisionInput»
decision condition

<Decision node with Condition Annotation>

[order rejected]

Receive Order

Fill Order

Close Order

[order accepted]

<Example of Decision Node>
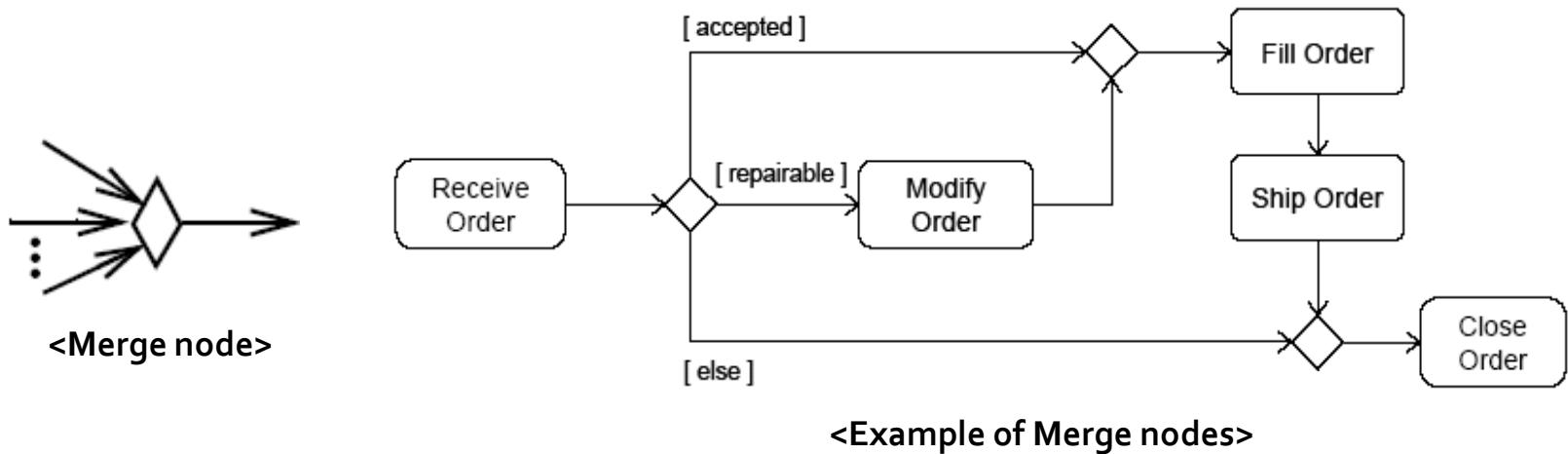
(Source: Debenedetti Emanuele, Activity diagrams in UML 2.0)

Control nodes – merge nodes
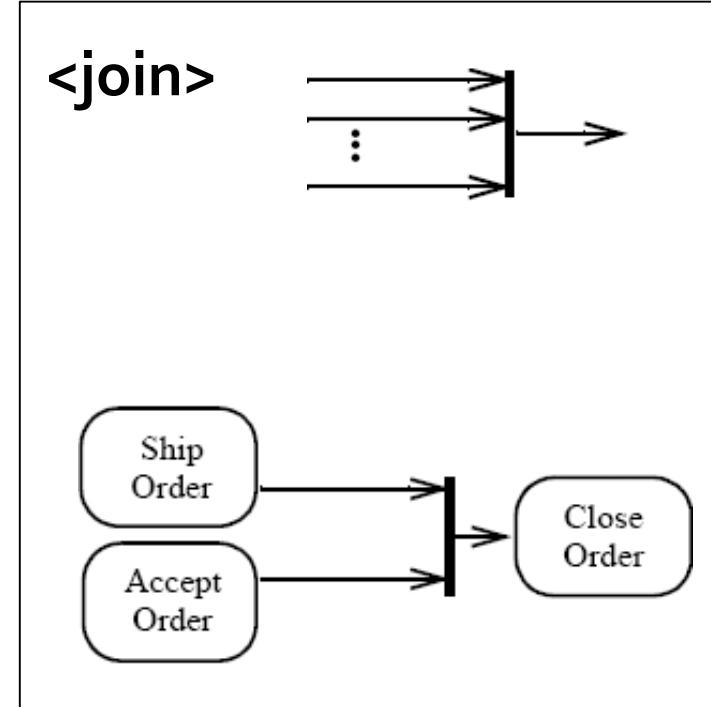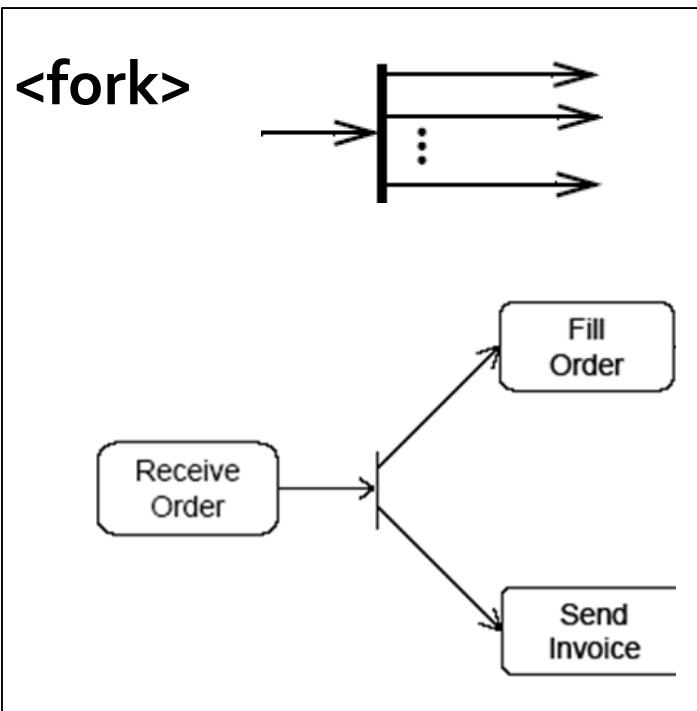
❖ **Bring together multiple alternate flows**

❖ **All controls and data arriving at a merge node are immediately passed to the outgoing edge**

❖ **There is no synchronization of flows**

<Merge node>

<Example of Merge nodes>

(Source: Debenedetti Emanuele, Activity diagrams in UML 2.0)

Control nodes – fork / join nodes

❖ **Fork nodes split flows into multiple concurrent flows (Cointrols are duplicated)**

❖ **Join nodes synchronize multiple flows**

(Source: Debenedetti Emanuele, Activity diagrams in UML 2.0)

Control nodes – final nodes
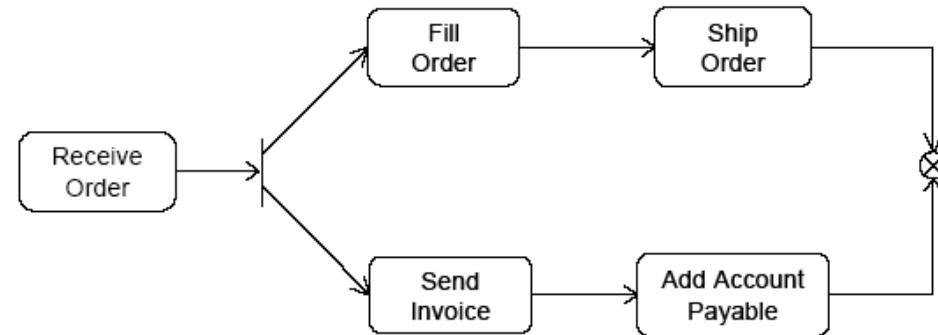
❖ **Flow final :**
- Stop the control that arrive into it
- The activity is terminated when all controls in the graph are stopped
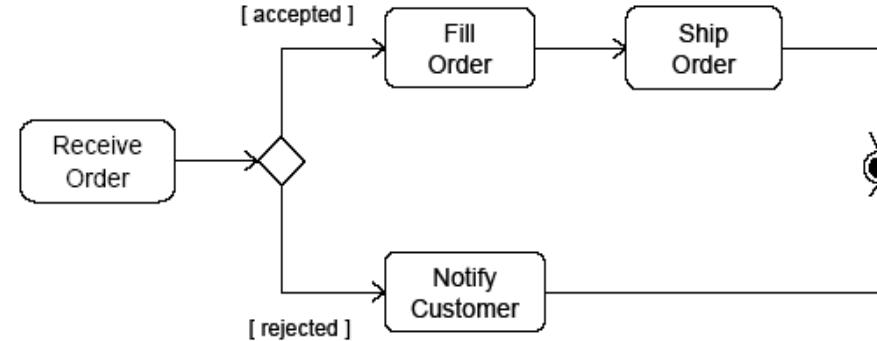
❖ **Final node :**
- The activity is termiated when the first control arrives

<Example of Flow Final>

<Example of Final Node>

CHUNGBUK NATIONAL UNIVERSITY

# Signal: SendSignalAction

❖ **Creates a signal instance from its inputs, and transmits it to the target object**

❖ **A signal is an asynchronous stimulus that triggers a reaction in the receiver in an asynchronous way and without a reply**

❖ **Any reply message is ignored**

<Symbol of SendSignal>

<Example of SendSignal Action>
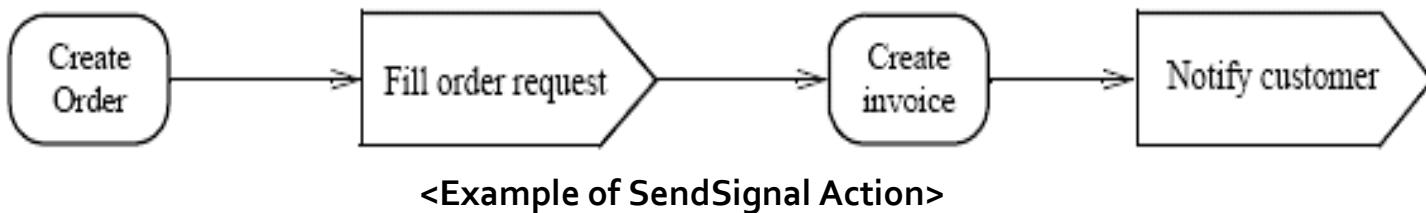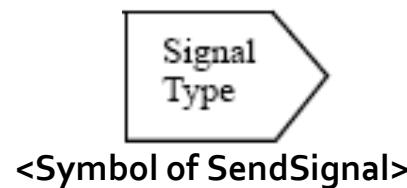
(Source: Debenedetti Emanuele, Activity diagrams in UML 2.0)

Signal: SendSignal / ReceiveSignal

<Example of Print Activity>
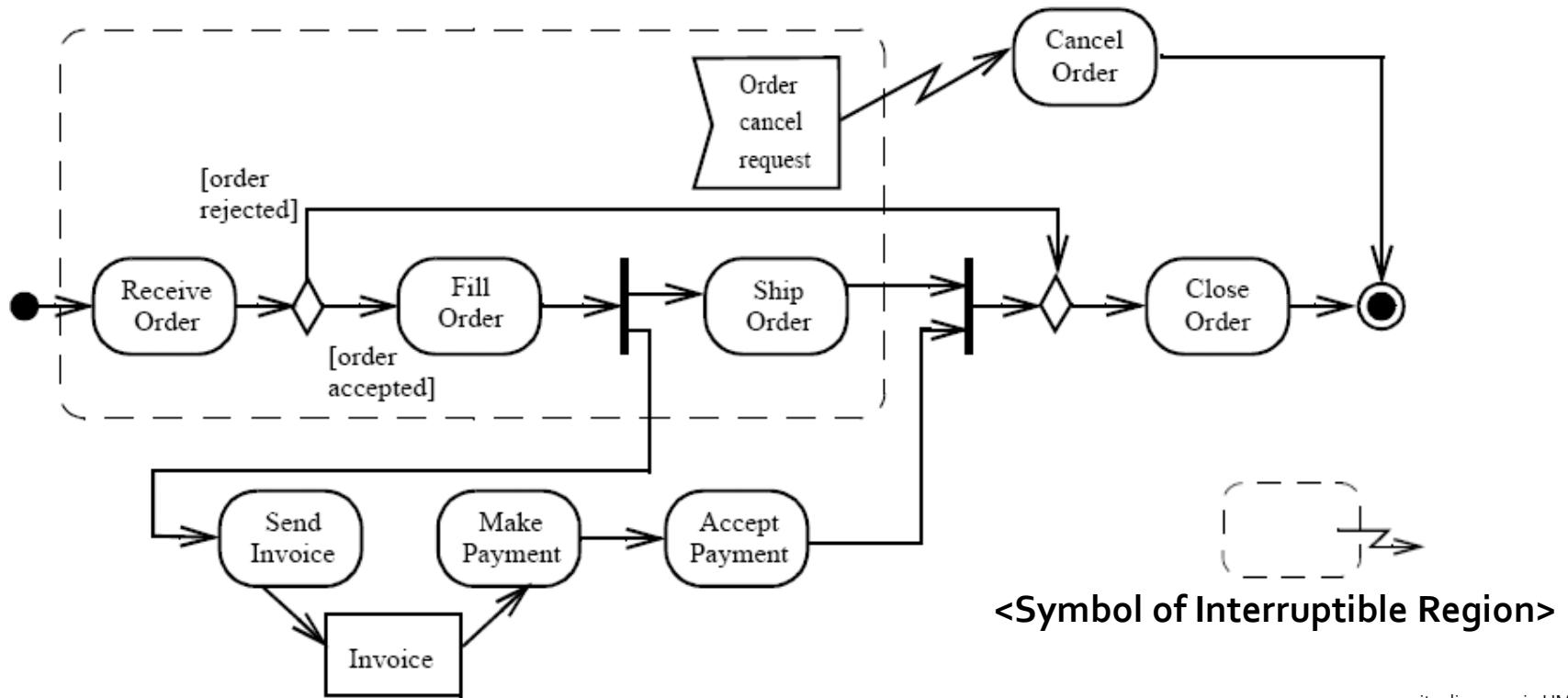
(Source: Debenedetti Emanuele, Activity diagrams in UML 2.0)

Interruptible Region
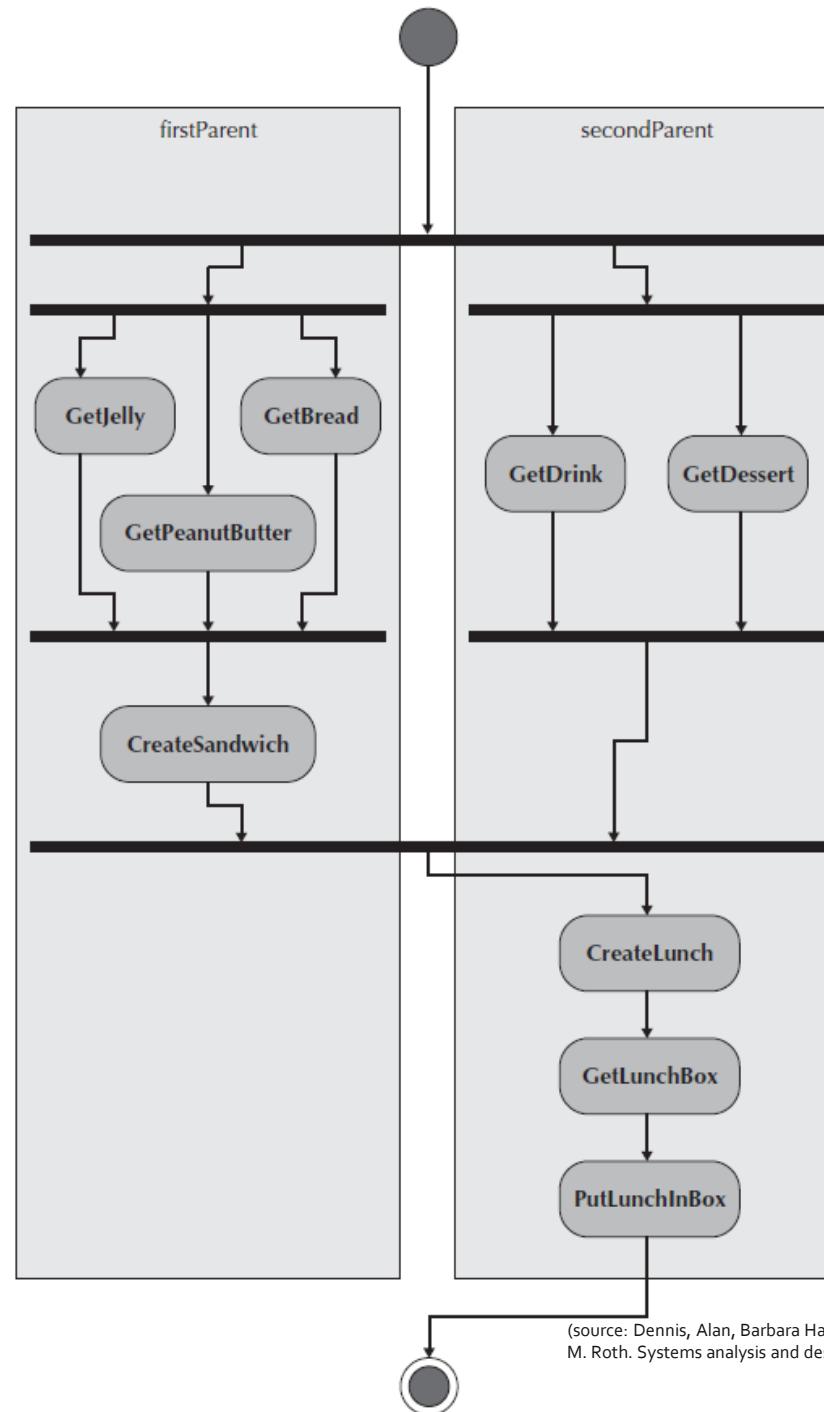
❖ **Is an activity group (sets of nodes and edges) that supports termination of control flow**

❖ **When receiving a signal to interrupt any activity within the interruptible region, all behaviors in the region are terminated**

<Symbol of Interruptible Region>

(Source: Debenedetti Emanuele, Activity diagrams in UML 2.0)

Swimlanes

❖ **Making a Lunch-Box**

❖ **showing Swimlanes**

(source: Dennis, Alan, Barbara Haley Wixom, and Roberta M. Roth. Systems analysis and design. John wiley & sons.)

Creating Activity Diagrams

1. Since an activity diagram can be used to model any kind of process, you should set the context or scope of the activity being modeled. Once you have **determined the scope**, you should give the diagram an appropriate title.

2. You must identify the activities, control flows, and object flows that occur between the activities.

3. You should identify any decisions that are part of the process being modeled.

4. You should attempt to identify any prospects for parallelism in the process.

5. You should draw the activity diagram.

Activity Diagram : Drawing Considerations

❖ **Minimize line crossings to enhance the readability**

❖ **Lay out the activities on the diagram in a left to right and/or top to bottom order**

❖ **Swimlanes**
  - Used to simplify the understanding
  - Minimizing the number of flows crossing the different swimlane
  - Horizontal orientation vs. Vertical orientation

❖ **No inflows or No outflows**

Your Turn – Activity

❖ **Create an activity diagram using the following business process**
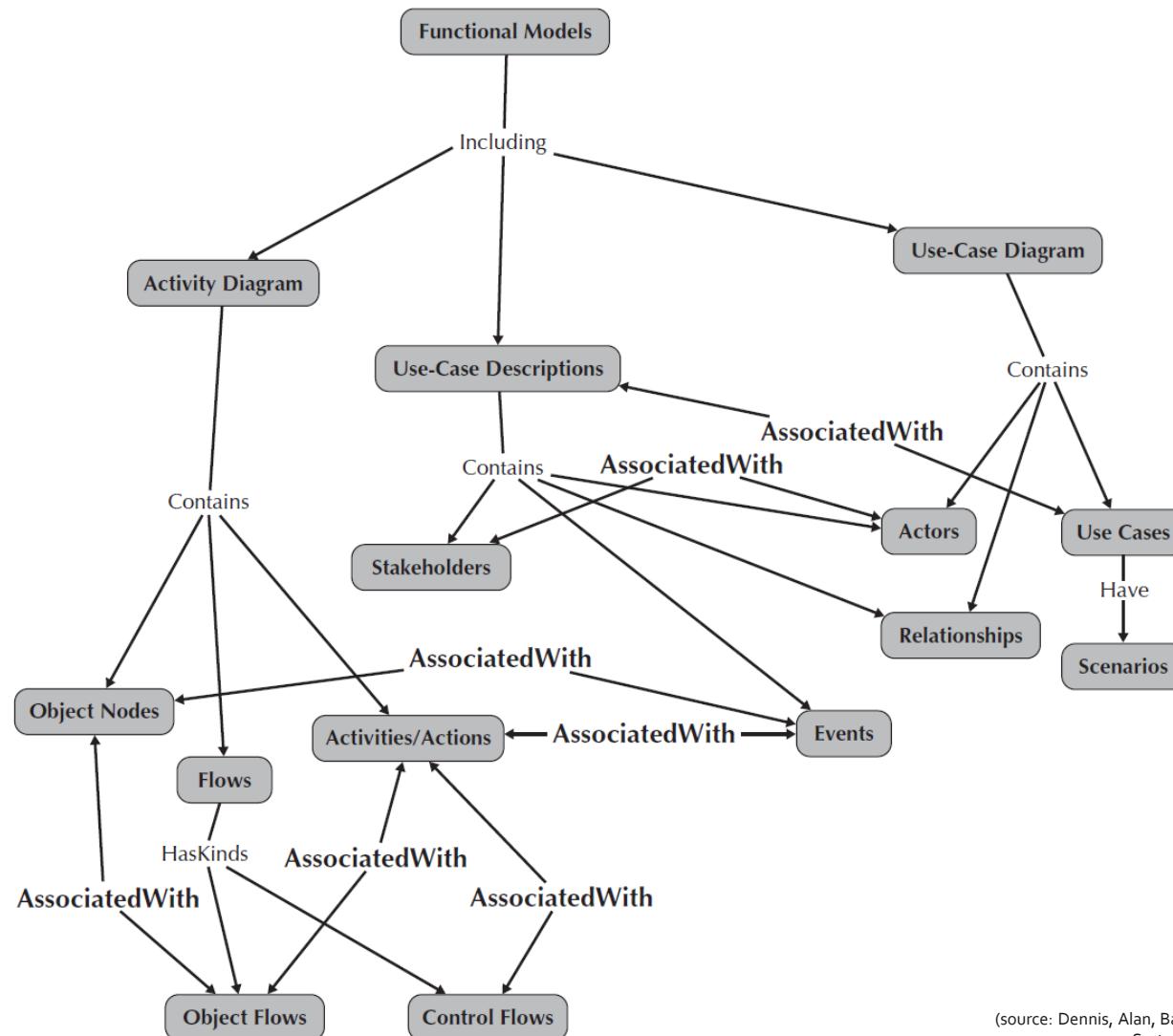
The first step is to see an eye doctor who will give a prescription.

Once you have a prescription, you go to a glasses store, where select

your frames and place the order for your glasses.

Once the glasses have been made, you return to the store for a fitting

and pay for the glasses.

Interrelationship among Functional Models

(source: Dennis, Alan, Barbara Haley Wixom, and Roberta M. Roth. Systems analysis and design. John wiley & sons.)

Summary and Discussion

❖ **Extended Symbols of Activity diagram from UML 1.5**
- Signals : SendSignal / ReceiveSignal
- Interruptible Region
- Exception, ...

❖ **What is the purpose of an activity diagram ?**

❖ **Why do we take Activity diagram to represent the business functionality (instead of flowchart, SDL diagram, etc) ?**