

# Systems Engineering: Design and Development

**ENGR 387** 



### Agenda

- Introduction
- SysML Behaviors and Behavioral Diagrams
- Purpose and Use of Activities and Activity Diagrams
- Purpose and Use of Object Nodes
- Purpose and Use of Control Nodes
- Purpose and Use of Actions
- Purpose and Use of Token Flow
- Purpose and Use of Object Flows
- Purpose and Use of Control Flows
- When Does an Action Start?
- When Does an Activity End?
- Review Questions



### Introduction

- Objectives:
  - Understand the purpose and use of the Activity Diagram
  - Understand the purpose and use of all types of Nodes and Edges
  - Understand Token Flow
  - Understand when an Action starts



### **SysML Behaviors and Behavior Diagrams**

- A behavior is an element of definition
- 3 kinds of behaviors:
  - Activity
  - State Machine
  - Interaction
- Each kind of behavior has its own diagram
- 3 types of behavioral diagrams:
  - Activity Diagram (act)- specifies an Activity
  - State Machine Diagram (stm)- specifies a State Machine
  - Sequence Diagram (sd)- specifies an Interaction



### Purpose and Use of Activities and Activity Diagrams

- Activity: behavior that specifies the transformation of inputs to outputs through a controlled sequence of actions
- Activity diagram: defines the actions in an activity along with the flow of input/output and control between them
  - Express *order* of actions
  - CAN express which structure *performs* each action
  - Does NOT express which structure *invokes* each action
- Activity diagram frame:
  - diagramKind abbreviation: act

act [Activity] Make Coffee [ Make Coffee ]

- modelElementType: activity
- Model Element Name: namespace of model element represented by the diagram



### Purpose and Use of Activities and Activity Diagrams Cont.

- 2 kinds of elements an activity can own:
  - Node
  - Edge
- 3 kinds of nodes:
  - Object Node
  - Control Node
  - Action
- 2 kinds of edges:
  - Object Flow
  - Control Flow
- An activity is a set of nodes connected by edges



## **Purpose and Use of Object Nodes**

- 3 kinds of object nodes:
  - Basic object node
  - Pin
  - Activity Parameter



### **Basic Object Node**

- Input or output of a single action within an activity
- Attach a pin to an action
- Notation:
  - Small square attached to the boundary on the outside of an action with name string:
  - <pin name>:<type>[<multiplicity>]
    - Pin name is user defined
    - Type is the name of a block, value type, or signal defined within the model
    - Multiplicity is the number of object tokens the pin can hold at any given moment/the number of object tokens required by an action
- Optional pin:
  - Lower multiplicity of 0
  - Optional input pin: Action can start with no object token at pin
  - Optional output pin: Action can execute and produce no object token at pin



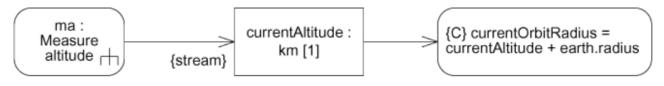


Figure 6.4: An object node between 2 actions from: SysML Distilled: A Brief Guide to the Systems Modeling Language by Lenny Delligatti

#### Pin

- Models the flow of object tokens through an activity
- Most often appears between 2 actions to convey that the first action produces object tokens as outputs, and the second action consumes those object tokens as inputs
- Notation:
  - Rectangle with name string:

<object node name>:<type>[<multiplicity>]

- Object node name is user defined
- Type is the name of a block, value type, or signal defined within the model
- Multiplicity is the number of object tokens the object node can hold at any given moment (default is 1:1)

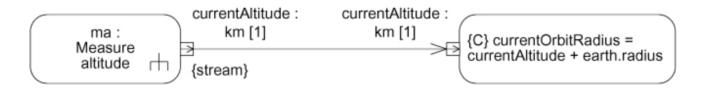


Figure 6.5: Actions with pins: SysML Distilled: A Brief Guide to the Systems Modeling Language by Lenny Delligatti



### **Activity Parameter**

- Input or output of the activity as a whole
- Attach an activity parameter to the frame of an activity diagram
- Notation:
  - Rectangle sitting on the frame of an activity diagram with name string:
  - <activity parameter name>:<type>[<multiplicity>]
    - Activity parameter name is user defined
    - Type is the name of a block, value type, or signal defined within the model
    - Multiplicity is the number of object tokens the activity parameter can hold at any given moment/the number of object tokens required by an activity
- Optional activity parameter:
  - Lower multiplicity of 0
  - Optional input activity parameter: Activity can start with no object token at activity parameter
  - Optional output pin: Activity can execute and produce no object token at activity parameter

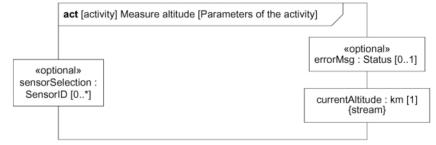




Figure 6.7: An activity diagram frame with activity parameters attached: SysML Distilled: A Brief Guide to the Systems Modeling Language by Lenny Delligatti

### Nonstreaming vs Streaming with Object Nodes

- Nonstreaming
  - · Input object tokens consumed by actions and activities only at the moment they begin executing
  - · Output object tokens delivered only at the moment they finish executing
- Streaming
  - Input object tokens consumed by actions and activities while behavior is executing
  - Output object tokens delivered while behavior is executing
  - Model continuous behavior
  - Notation: {stream} at the end of the name string for a pin or an activity parameter
- An output pin and the input pin it is connected to do NOT have to both be streaming or nonstreaming, they are independent design decisions
- Object nodes are described as streaming or nonstreaming, NOT actions



Figure 6.8: An action with nonstreaming pins: SysML Distilled: A Brief Guide to the Systems Modeling Language by Lenny Delligatti

Figure 6.9: An action with a streaming pins: SysML Distilled: A Brief Guide to the Systems Modeling Language by Lenny Delligatti



## **Purpose and Use of Control Nodes**

- 7 kinds of control nodes:
  - Initial node
  - Activity Final node
  - Flow Final node
  - Decision node
  - Merge node
  - Fork node
  - Join node



#### **Initial Node**

- Marks the starting point within an activity
- Marks a place in the activity where the flow of a control token begins
- A single activity can have multiple initial nodes
- An activity does not need to have an initial node
- Notation: small, filled-in circle

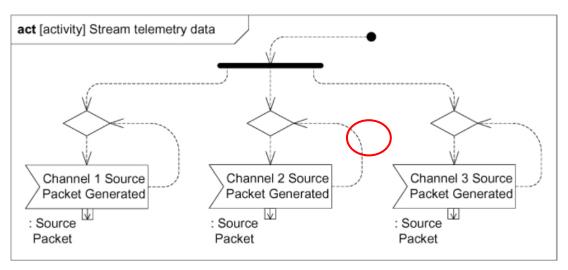
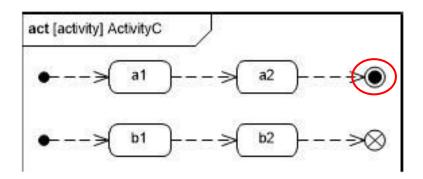


Figure 6.20: An activity fragment containing an initial node: SysML Distilled: A Brief Guide to the Systems Modeling Language by Lenny Delligatti



## **Activity Final Node**

- When a control token arrives at an activity final node, the entire activity terminates
- If multiple edges are going into the activity final node, the first token to arrive terminates the activity
- If an activity has multiple activity final nodes, whichever node has a token arrive first terminates the activity
- Notation: circle containing a smaller filled in circle

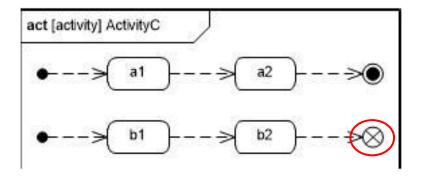


From Question 6-4 in OCSMP Sample Questions from the OCSMP Accelerator Course by Lenny Delligatti



### **Flow Final Node**

- When a control token arrives at flow final node, only that token is destroyed
- Terminates a sequence of actions without terminating the activity
- Notation: hollow circle containing a X



From Question 6-4 in OCSMP Sample Questions from the OCSMP Accelerator Course by Lenny Delligatti



#### **Decision Node**

- Marks the start of alternative sequences in an activity
- Has a single incoming edge and 2 or more outgoing edges
- Each outgoing edge has a Boolean expression called a guard
  - Guard notation: string between square brackets
- When a token arrives at a decision node, all the guards are evaluated simultaneously
- Token is passed to the edge whose guard evaluates to true at that moment
- Token is only passed to one outgoing edge, NEVER multiple outgoing edges
- Notation: hollow diamond

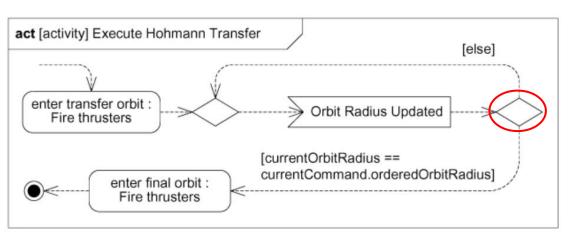




Figure 6.22: Using a merge node to model a loop: SysML Distilled: A Brief Guide to the Systems Modeling Language by Lenny Delligatti

### Merge Node

- Marks the end of alternative sequences in an activity
- Has 2 or more incoming edges and a single outgoing edge
- Token is immediately passed to the outgoing edge when it arrives on ANY of the incoming edges
- Essential to model a loop correctly
- Notation: hollow diamond

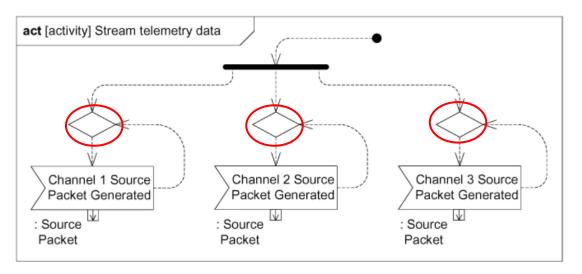


Figure 6.20: An activity fragment containing an initial node: SysML Distilled: A Brief Guide to the Systems Modeling Language by Lenny Delligatti



#### **Fork Node**

- Marks the start of concurrent sequences in an activity
- Has a single incoming edge and 2 or more outgoing edges
- When a token arrives at a fork node, it is duplicated and passed to each outgoing edge
- Each copy of original token represents an independent, concurrent flow
- Notation: line segment

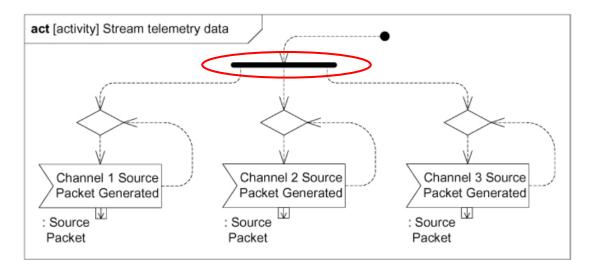
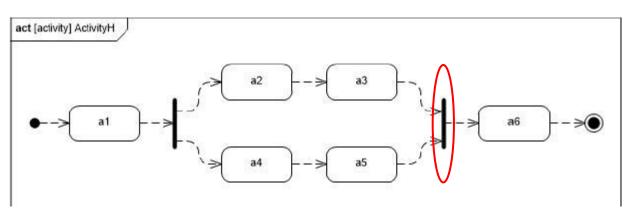


Figure 6.20: An activity fragment containing an initial node: SysML Distilled: A Brief Guide to the Systems Modeling Language by Lenny Delligatti



#### **Joint Node**

- Marks the end of concurrent sequences in an activity
- Has 2 or more incoming edges and a single outgoing edge
- All incoming edges are control flows:
  - A single control token is passed to the outgoing edge when a token arrives on EACH of the incoming edges
- All incoming edges are object flows:
  - ALL object tokens are passed to the outgoing edge when a token arrives on EACH of the incoming edges
- Incoming edges are a mix of control and object flows:
  - ALL object tokens and NO control tokens are passed to the outgoing edge when a token arrives on EACH of the incoming edges
- Notation: line segment





From Question 6-6 in OCSMP Sample Questions from the OCSMP Accelerator Course by Lenny Delligatti

### **Purpose and Use of Actions**

- 4 kinds of actions:
  - Call behavior action
  - Send signal action
  - Accept event action
  - Wait time action
- An action is an element of usage, a behavior is an element of definition
- An action represents the execution of a defined behavior



### **Purpose and Use of Actions**

- 4 kinds of actions:
  - Call behavior action
  - Send signal action
  - Accept event action
  - Wait time action
- An action is an element of usage, a behavior is an element of definition
- An action represents the execution of a defined behavior



### Synchronous vs. Asynchronous

- Synchronous
  - Invoke behavior and sender waits for reply
- Asynchronous
  - Invoke behavior and sender does NOT wait for reply



#### **Call Behavior Action**

- · Specialized action that invokes another behavior when it becomes enabled
  - Can call any of the 3 kinds of behavior
- Can represent behavioral decomposition on an activity diagram
- Notation: rectangle with rounded corners with name string

<action name>:<Behavior name>

- Action name is user defined
- Behavior name MUST match the name of an interaction, a state machine, or an activity defined somewhere in the model
- Rake symbol in lower right corner conveys the behavior being called is an activity
- If a call behavior action invokes another activity, pins of call behavior action MUST match the activity parameters of called activity





Figure 6.14: A call behavior action that invokes the behavior Stream telemetry data: SysML Distilled: A Brief Guide to the Systems Modeling Language by Lenny Delligatti

### **Send Signal Action**

- Specialized kind of action that asynchronously generates and sends a signal instance to a target when it becomes enabled
- Send signal action has to have an input pin for each property of the signal being sent
- Signal can be sent to a block with behavior assigned or to a behavior
- Notation: convex pentagon shaped like a signpost
  - String inside MUST match the name of a signal defined somewhere in the model

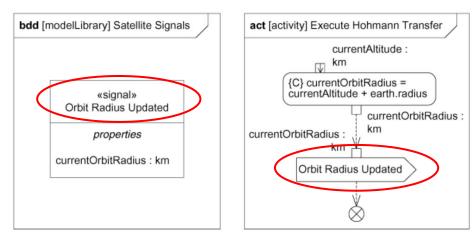


Figure 6.16: A signal and send signal action: SysML Distilled: A Brief Guide to the Systems Modeling Language by Lenny Delligatti



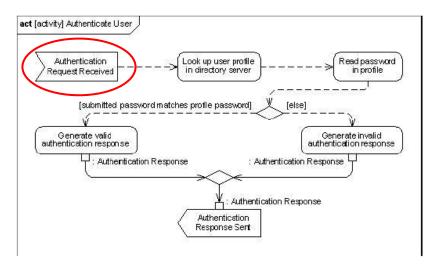
### **Accept Event Action (1 of 2)**

- Used in an activity to convey that the activity must wait for an asynchronous event occurrence before it can continue execution
- 4 kinds of events:
  - Signal event- the arrival of a signal instance at a target
  - Time event- indicates that a given time interval has passed since current state entered (relative) or that given instance in time has arrived (absolute)
  - Call event- receipt of a request to invoke a behavior or operation of a block
  - Change event- Boolean expression that changes from false to true



### **Accept Event Action (1 of 2)**

- Event has NOT occurred:
  - When the event occurs, accept event action executes
- Event has occurred:
  - Accept event action execution proceeds immediately
- Event occurrences are buffered until consumed by a behavior that cares
- Accept event action does NOT need any incoming edges
  - Starts executing as soon as the activity begins executing
  - Remains enabled
- Accept event action waiting for a signal has to have an output pin for each property of the signal being sent
- Notation: concave pentagon shaped like a pennant
  - String inside OFTEN matches the name of a signal defined somewhere in the model



From Question 6-2 in OCSMP Sample Questions from the OCSMP Accelerator Course by Lenny Delligatti



#### **Wait Time Actions**

- An accept event action that waits for a time event occurrence
- Absolute time event: begins with the word "at"
- Relative time event: begins with the word "after"
- Becomes enabled when a control token arrives on its incoming control flow
  - If absolute time event has already occurred:
    - Wait time action completes immediately
  - If absolute time event has NOT occurred:
    - Wait time action waits for that time event to occur
  - Clock for the relative time event begins when the wait time action becomes enabled
- Wait time action does NOT need any incoming edges
  - Starts executing as soon as the activity begins executing
  - Remains enabled
- Notation: hourglass symbol with a time expression string beneath it

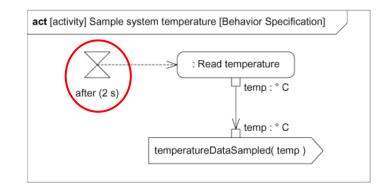


Figure 6.19: A wait time action with a relative time expression: SysML Distilled: A Brief Guide to the Systems Modeling Language by Lenny Delligatti

### **Purpose and Use of Token Flows**

- Tokens are not model elements; token flow is an abstract concept
- 2 kinds of tokens:
  - Object tokens
  - Control tokens
- Object token
  - Represents an instance of a block, value type, or signal that you've created somewhere in your model hierarchy
  - Represents an instance of matter, energy, or data that flows through an activity
  - Can represent an input or an output of the activity as a whole
  - Can represent an input or an output of an action within the activity
- Control token
  - Represents the flow of control
  - Does NOT represent anything physical; has no type
- No notion of time associated with token flow



### **Purpose and Use of Object Flows**

- Kind of edge that transports object tokens
- Used to convey that instance of matter, energy, or data through an activity from one node to another when the activity executes during system operation
- Must ensure object nodes at ends of object flow have compatible types
  - Types can be identical
  - Upstream type can be a subtype of downstream type
- Notation: solid line with an open arrowhead

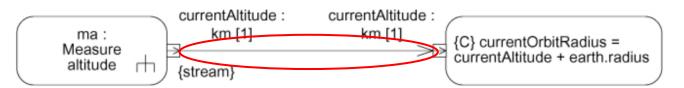


Figure 6.5: Actions with pins: SysML Distilled: A Brief Guide to the Systems Modeling Language by Lenny Delligatti



### **Purpose and Use of Control Flows**

- Kind of edge that transports control tokens
- Arrival of a control token enables an action requiring one
- Used to convey sequencing constraints among a set of action when object flows alone are not sufficient
- Notation: dashed line with open arrowhead or solid line with open arrowhead

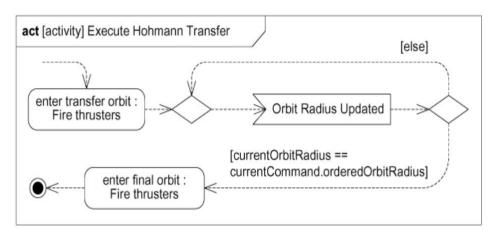


Figure 6.22: Using a merge node to model a loop: SysML Distilled: A Brief Guide to the Systems Modeling Language by Lenny Delligatti



#### When Does Action Start?

- 3 conditions MUST be satisfied for an action to start:
  - The activity that owns the action is currently executing
  - A control token arrives on *each* of the incoming control flows
  - A sufficient number of object tokens arrive on each of the incoming object flows to satisfy the lower multiplicity of the respective input pin
- Multiple incoming edges represents an *and* condition NOT an *or* condition
  - Tokens do NOT need to arrive at the same time, but token MUST be present on all incoming edges for the
    action to start
- An action does NOT need any incoming edges
  - Action begins to execute as soon as the owning activity begins to execute



### When Does an Activity End?

- When a token arrives at an activity final node
- When all the actions inside the activity come to an end
  - If any action continues executing, the only way to terminate activity is an activity final node



# Questions





# **Summary**



### References

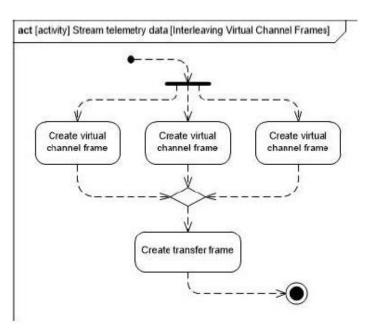
- Additional information can be obtained by reviewing:
  - SysML Distilled (Delligatti)
    - Chapter 6: Activity Diagrams
  - A Practical Guide to SysML (Friedenthal)
    - Chapter 9:Modeling Flow-Based Behavior with Activities

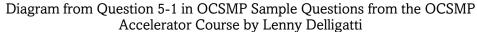




What is the Model Element Name?

- a. Interleaving Virtual Channel Frame
- b. act
- c. activity
- d. Stream telemetry data

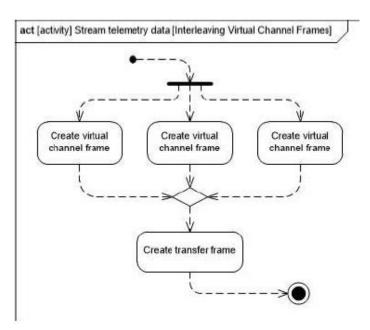


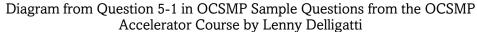




What is the Model Element Name?

- a. Interleaving Virtual Channel Frame
- b. act
- c. activity
- d. Stream telemetry data







How many control nodes does the diagram contain?

- a. 3
- b. 4
- c. 7
- d. 9

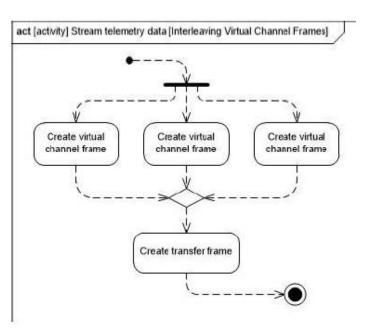


Diagram from Question 5-1 in OCSMP Sample Questions from the OCSMP Accelerator Course by Lenny Delligatti



How many control nodes does the diagram contain?

- a. 3
- b. 4
- c. 7
- d. 9

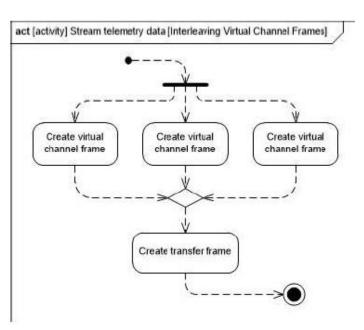


Diagram from Question 5-1 in OCSMP Sample Questions from the OCSMP Accelerator Course by Lenny Delligatti



#### What is a2?

- a. Behavior
- b. Activity Parameter
- c. Call Behavior Action
- d. Block

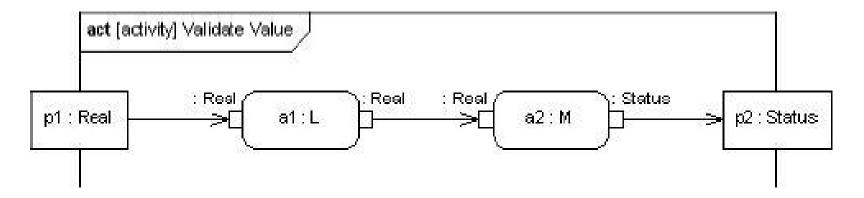


Diagram from Question 6-1 in OCSMP Sample Questions from the OCSMP Accelerator Course by Lenny Delligatti



#### What is a2?

- a. Behavior
- b. Activity Parameter
- c. Call Behavior Action
- d. Block

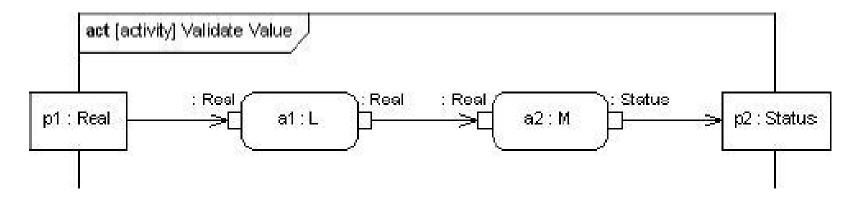
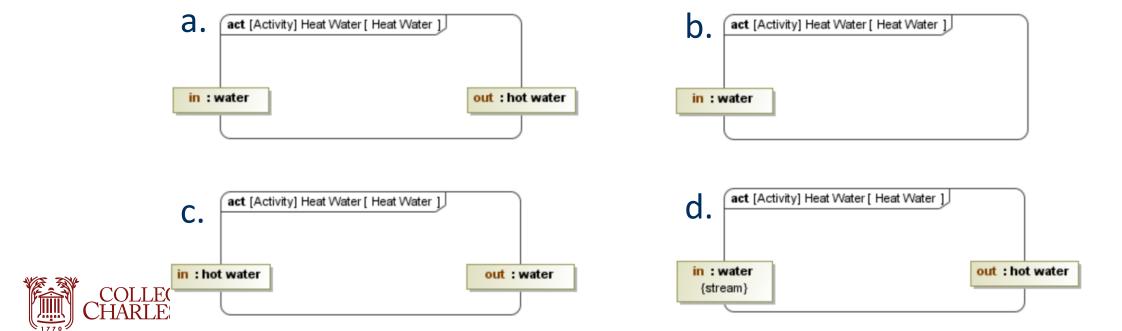


Diagram from Question 6-1 in OCSMP Sample Questions from the OCSMP Accelerator Course by Lenny Delligatti



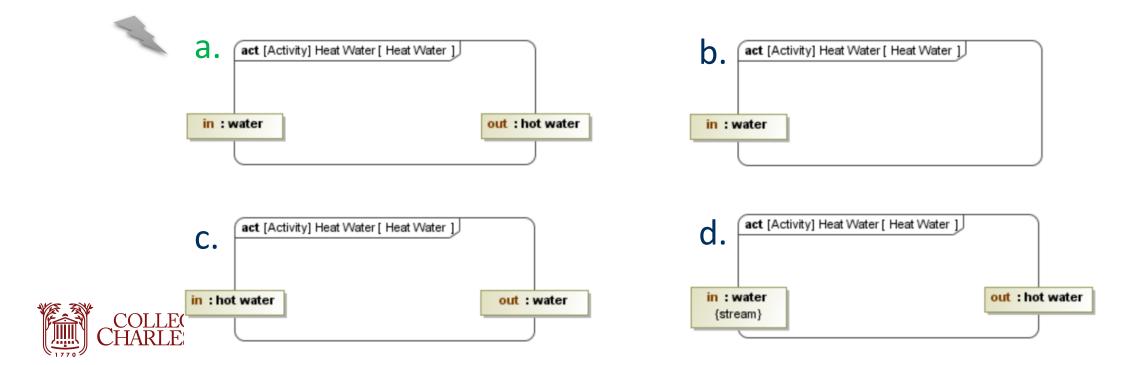


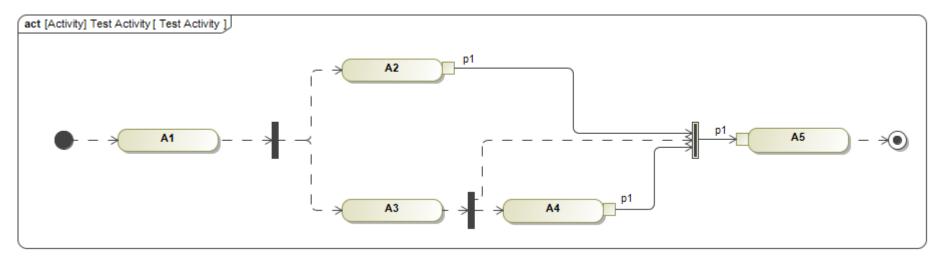
Which of the following is the proper depiction of the activity invoked in the call behavior action above?





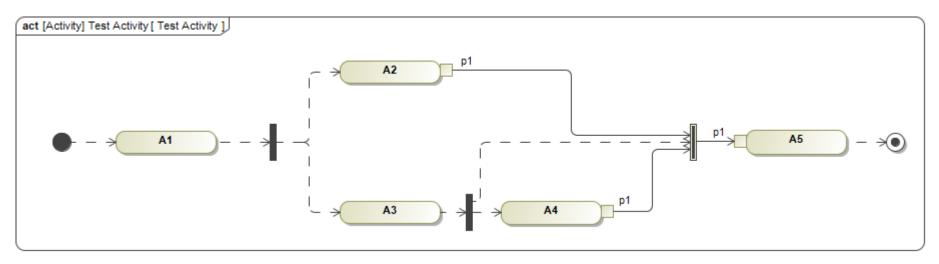
Which of the following is the proper depiction of the activity invoked in the call behavior action above?





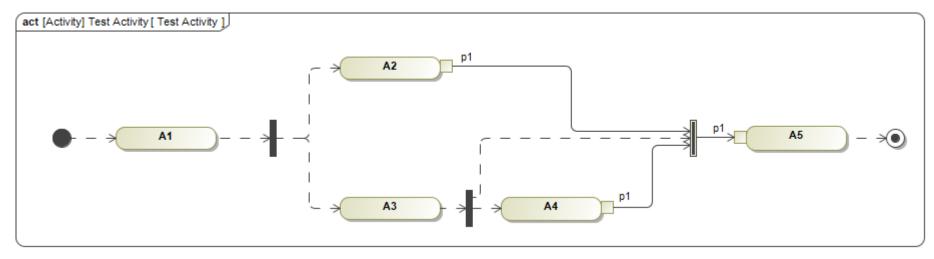
How many control tokens and how many object tokens are on the incoming edge for A5?

- a. 1 control token, 2 object tokens
- b. 1 control token, 1 object token
- c. 0 control token, 2 object tokens
- d. 0 control token, 1 object token



How many control tokens and how many object tokens are on the incoming edge for A5?

- a. 1 control token, 2 object tokens
- b. 1 control token, 1 object token
- c. 0 control token, 2 object tokens
- d. 0 control token, 1 object token



How many control tokens and how many object tokens are on the incoming edge for A5?

- a. 1 control token, 2 object tokens
- b. 1 control token, 1 object token
- c. 0 control token, 2 object tokens
- d. 0 control token, 1 object token