

Processes Modelling with UML Activity Diagram

Erik Perjons



Question to answer

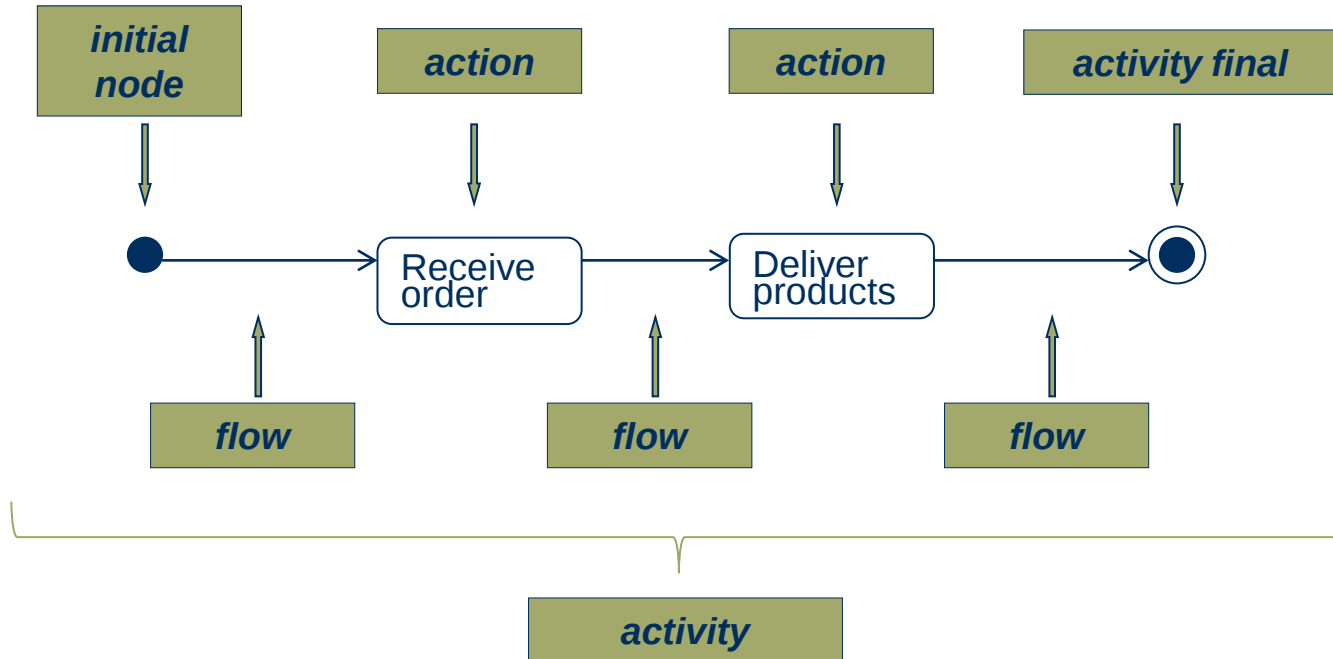
- How do you model business processes using a formal business process modelling language, in this case UML Activity Diagram?



UML Activity Diagram



UML Activity Diagram



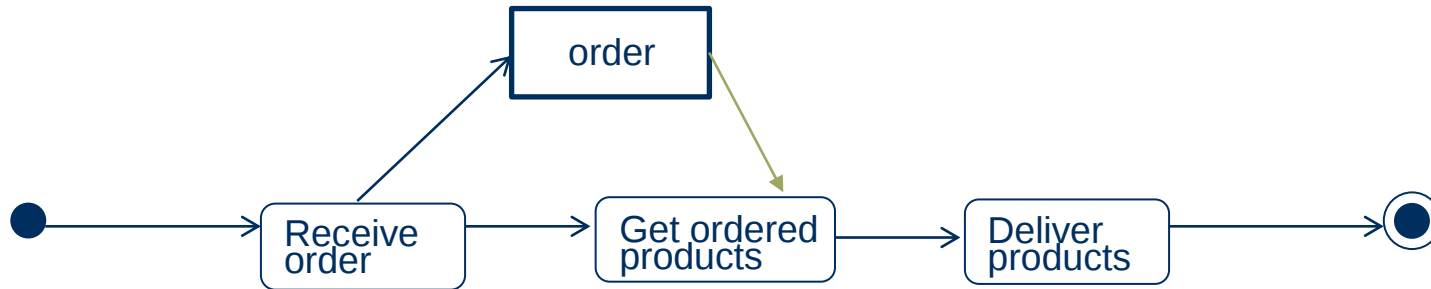
The Control Flow

- The flow symbol (i.e. the arrow) represents the order of actions. For example, the action "Receive order" should be carried out before the action "Get ordered products"
- The order of actions in an activity (or process) is sometimes called the control flow (which differs from the data or information flow)

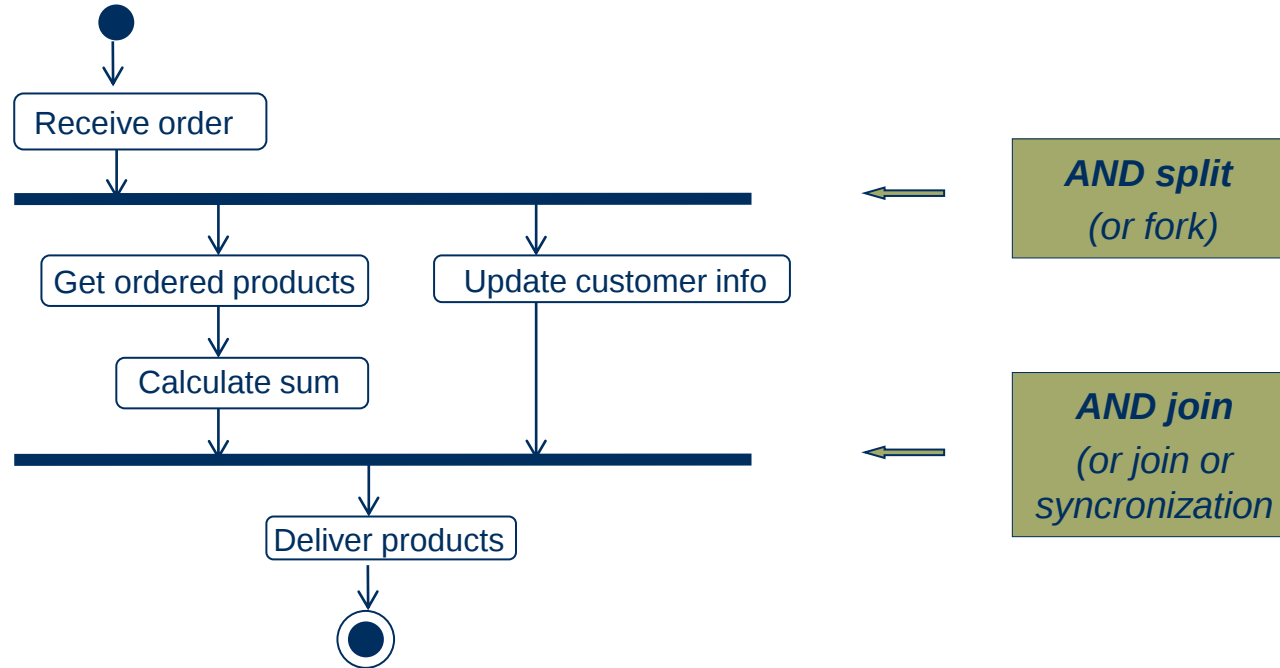


The Information Flow

- It is also possible in UML Activity Diagram to model the flow of data or information
- The flow of data or information is shown as a labeled box, and it model explicitly the input to and output from actions

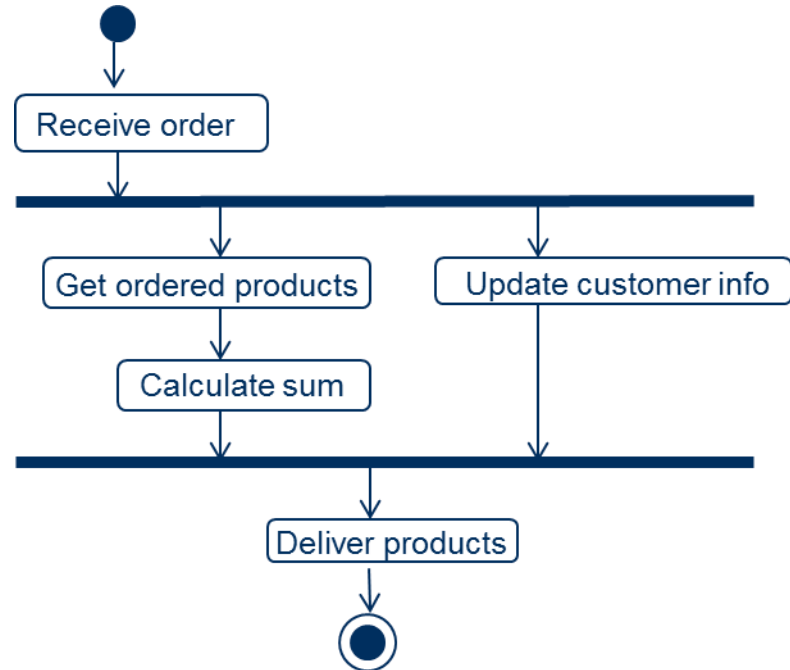


Control flow: Parallel behaviour



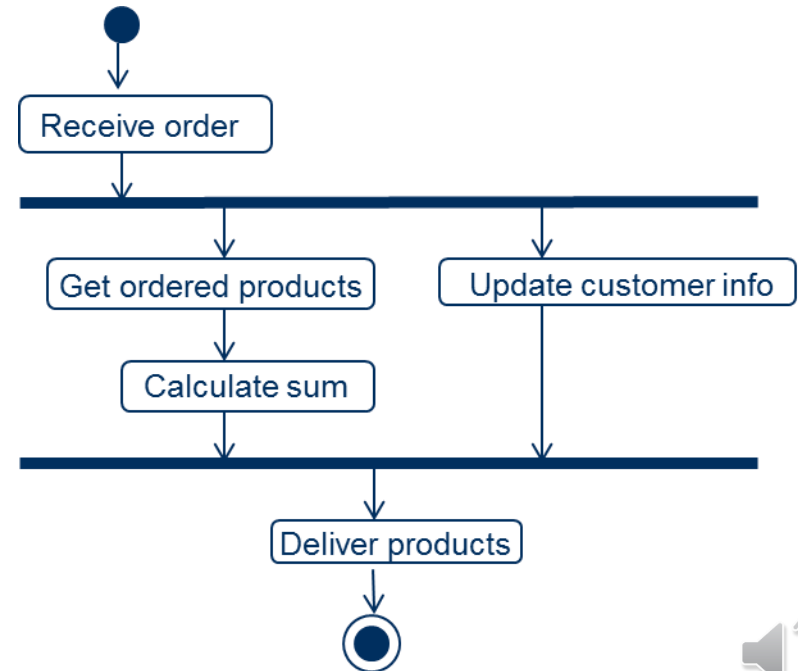
Control flow: Parallel behaviour

- An **AND split** has one flow in and several flows out
- An **AND join** has several flows in and one flow out

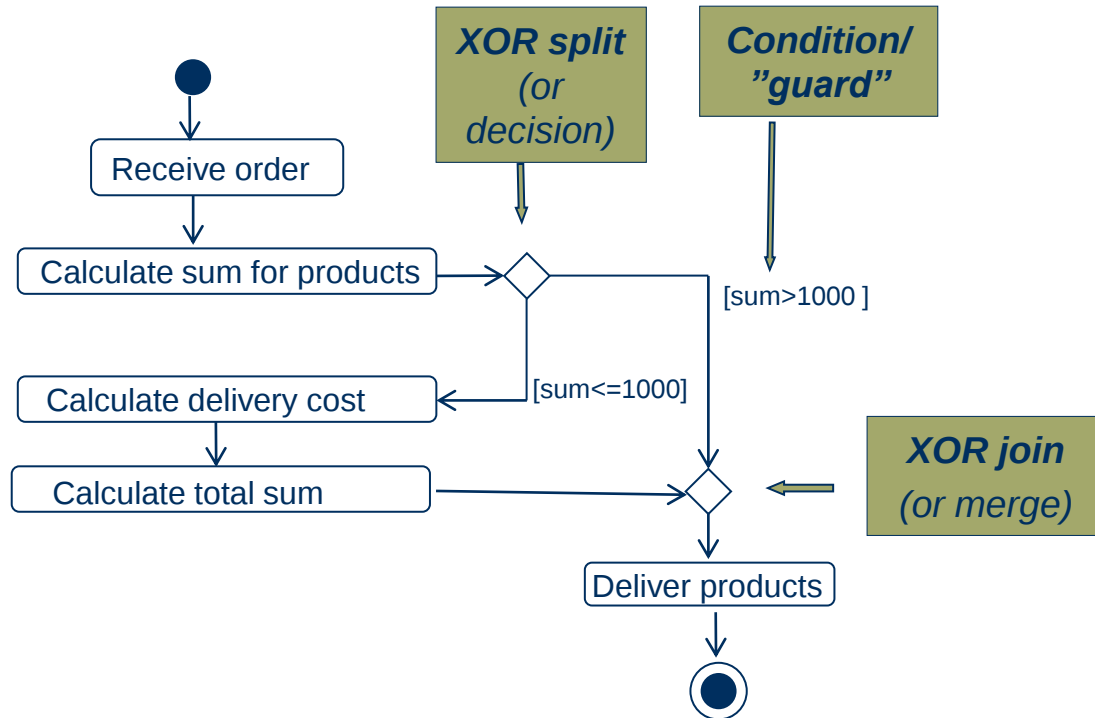


Control flow: Parallel behaviour

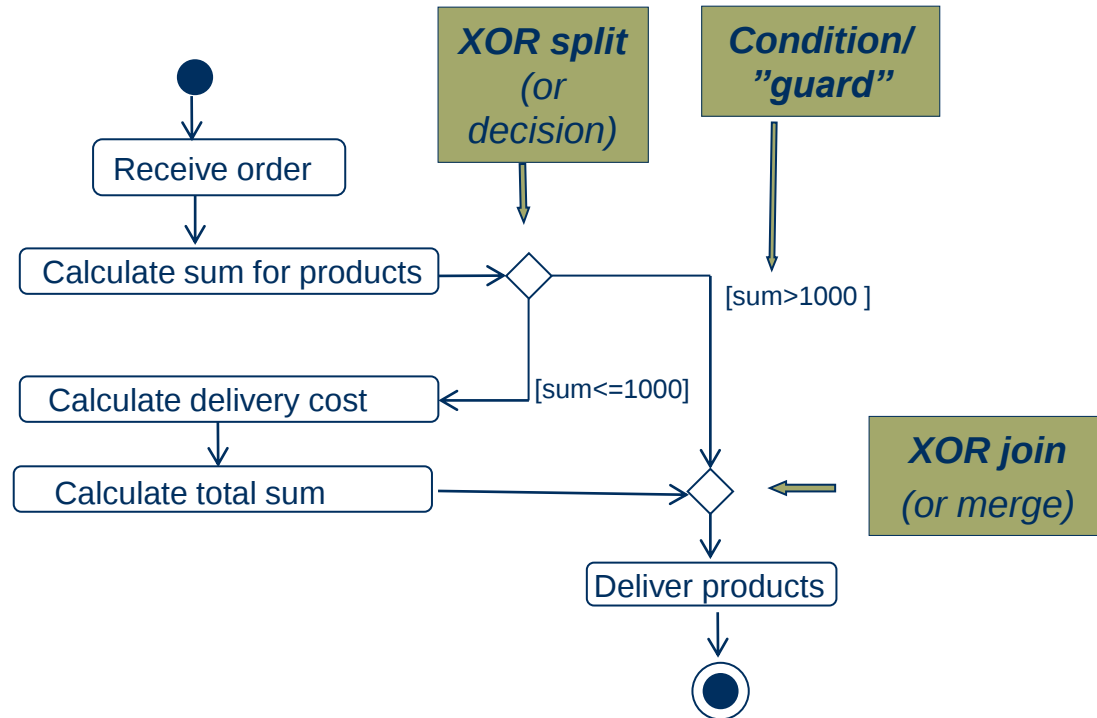
- **Guideline:** Parallel behaviour can be used in two situations:
 - Situation 1: Two actions occure in the same time
 - Situation 2: Two actions can occure in any order which is not specified



Control flow: Conditional behaviour

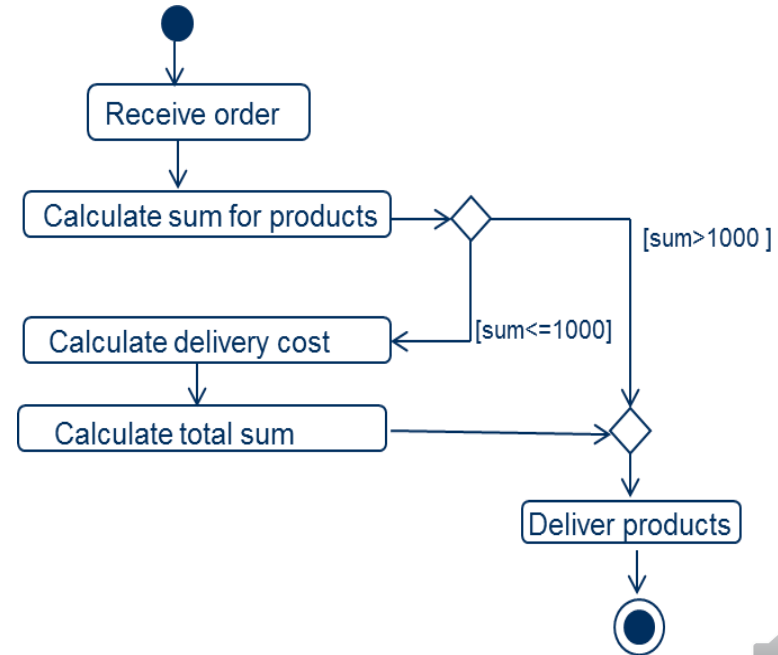


Control flow: Conditional behaviour



Control flow: Conditional behaviour

- An **XOR split** has one flow in and several flows out. The condition/guard must be true for a process instance to choose a certain path
- An **XOR join** has several flows (edges) in and one flow out



Guidelines

- **Guideline:** UML does not use the terms AND split and AND join, and XOR split and XOR join (UML use the terms forks, join, decision and merge), but I recommend the use of these terms, because they are easy to remember and can be used in different business process languages
- **Guideline:** Each AND split should be followed by an AND join later in the activity (process)
- **Guideline:** If possible, each XOR split should be followed by an XOR join later in the activity (process)

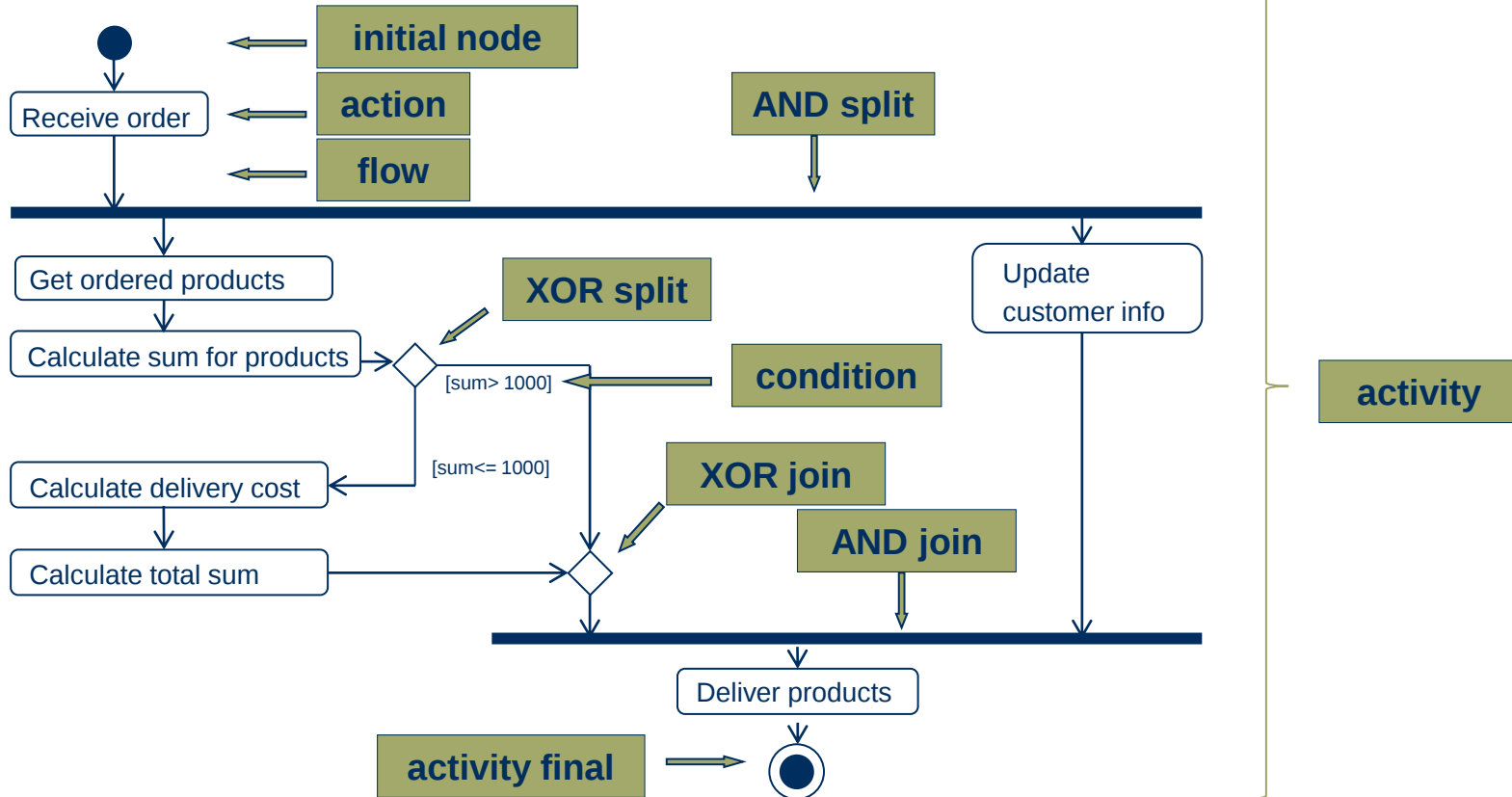


More Guidelines

- **Guidelines:** Start an activity with one initial node (although UML Activity Diagram does not forbid to have several initial nodes in an activity) – since it a common requirement for business processes models to have one start node
- **Guideline:** Several activity final nodes can be used



Completed example

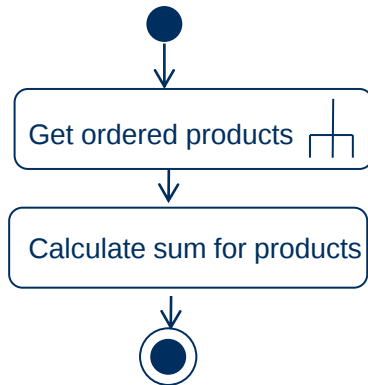


More about UML Activity Diagram



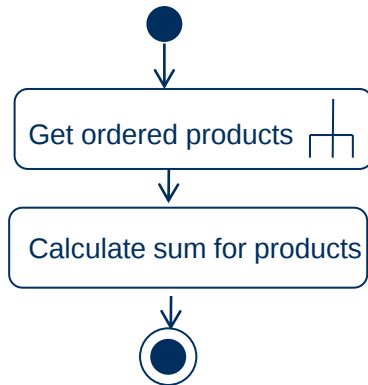
Action vs. Activity

- An **UML Activity Diagram** can include both **activities** and **actions**
- **Action** is a named element which represents a single atomic step within an activity, that is, an action cannot be further decomposed
- **Activity** represents a behavior that is composed of activities and/or actions



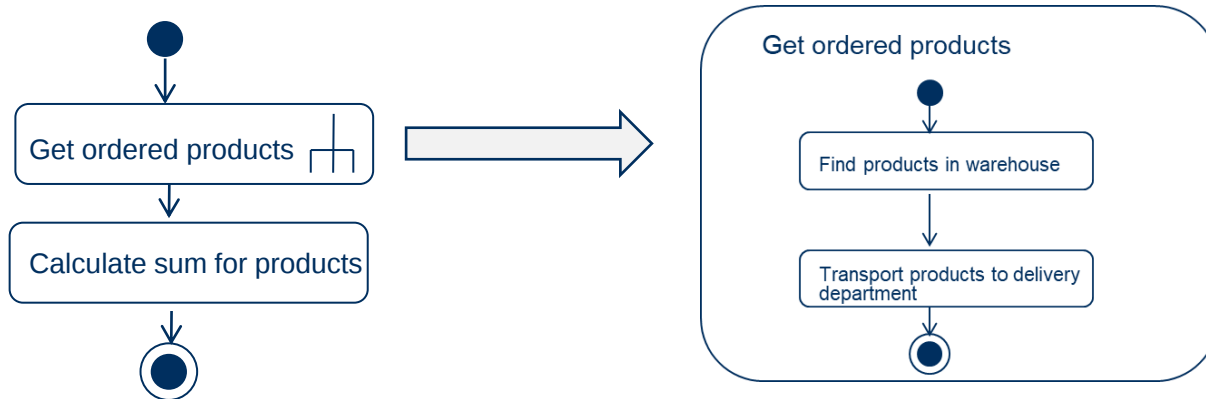
Action vs. Activity

- An **UML Activity Diagram** can include both **activities** and **actions**
- **Action** is a named element which represents a single atomic step within an activity, that is, an action cannot be further decomposed
- **Activity** represents a behavior that is composed of activities and/or actions



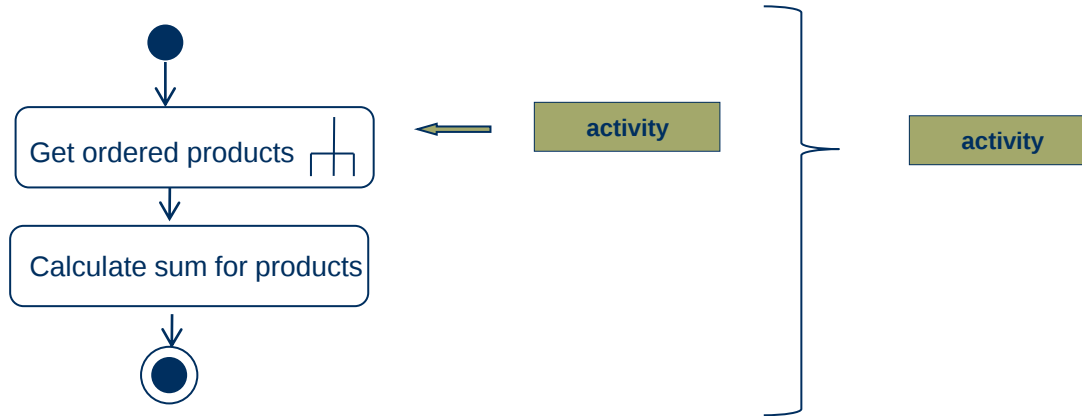
Action vs. Activity

- An **UML Activity Diagram** can include both **activities** and **actions**
- **Action** is a named element which represents a single atomic step within activity, that is, an action cannot be further decomposed
- **Activity** represents a behavior that is composed of activities and actions



Action vs. Activity

- **Activity** represents a behavior that is composed of individual elements that are actions and/or activities



Token

- The modelling element "**token**" is shorthand for **control and data values** that **flow through an activity**
- **A token could represent many things:** an specific order, a case, a patient which can flow through an activity

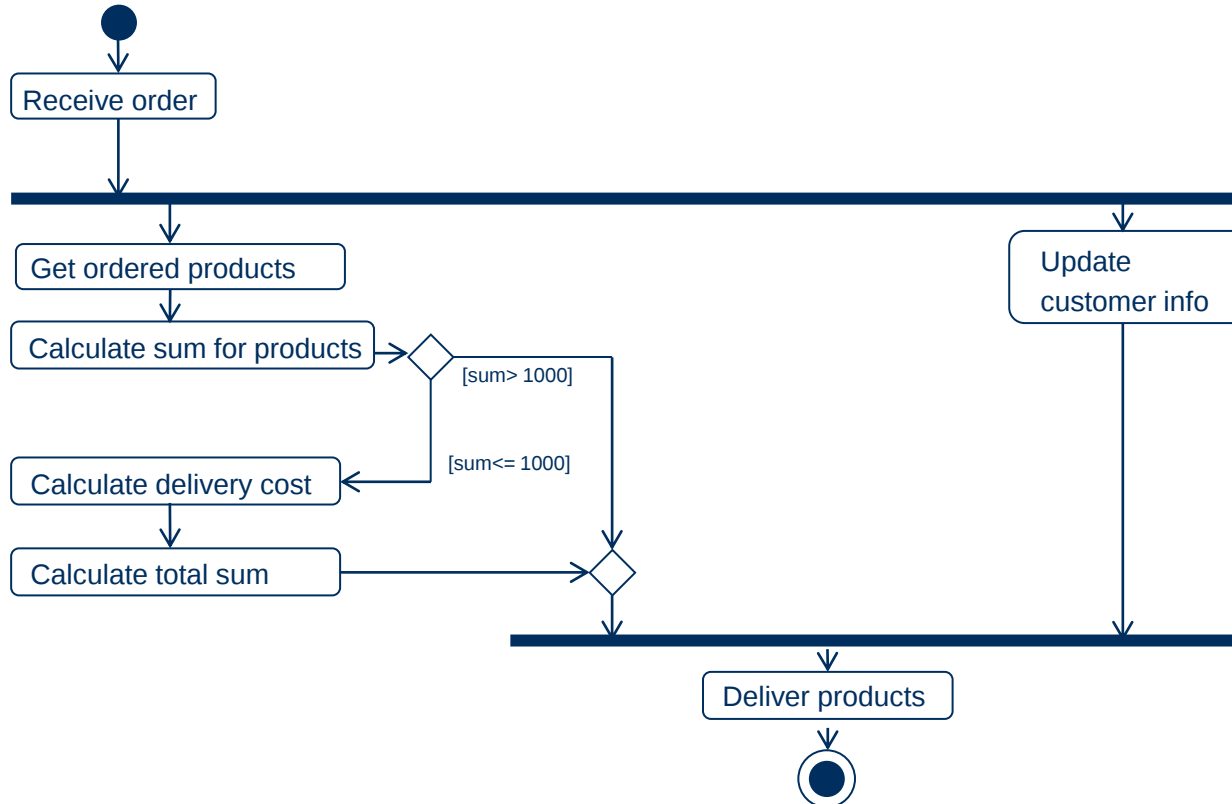


Token

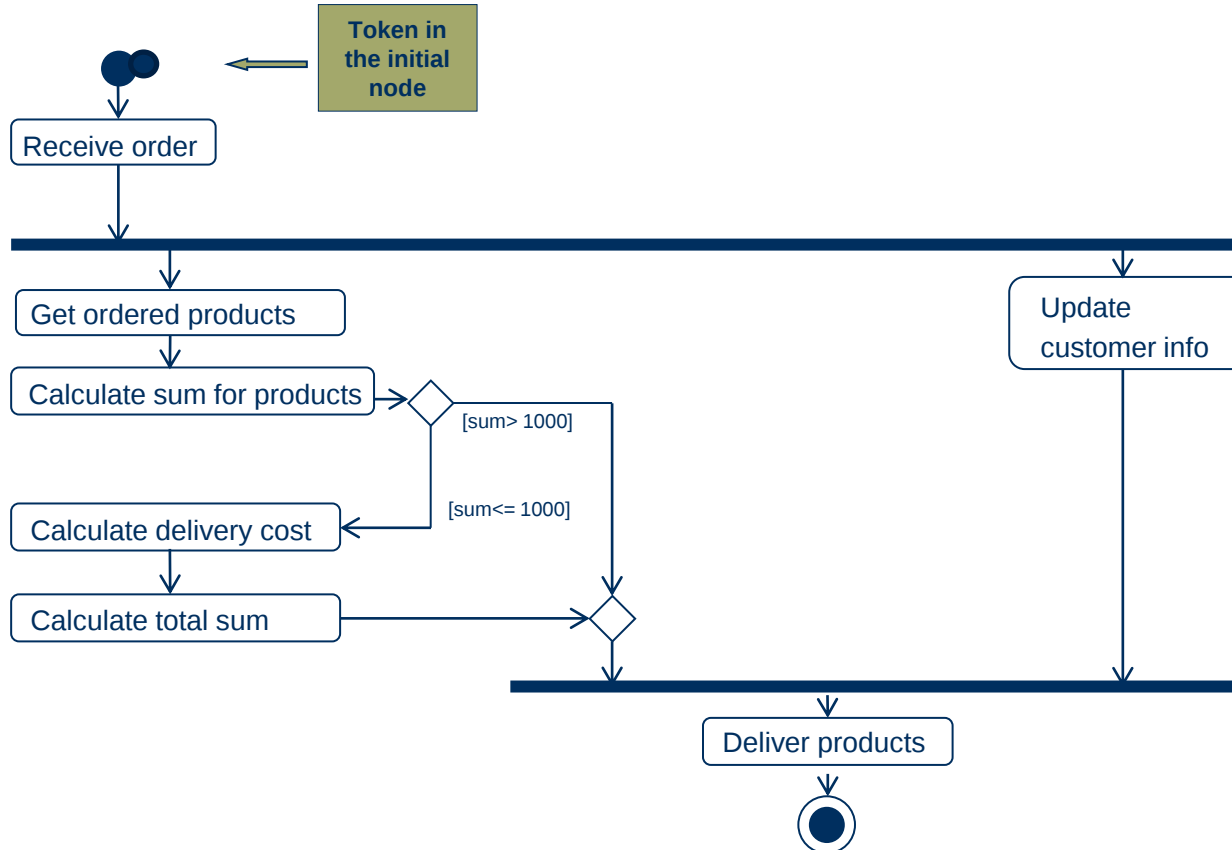
- **Token is consumed and produced in each actions/activity** in the diagrams.
- **Guideline:** Use tokens to see if the activity diagram is modelled correctly



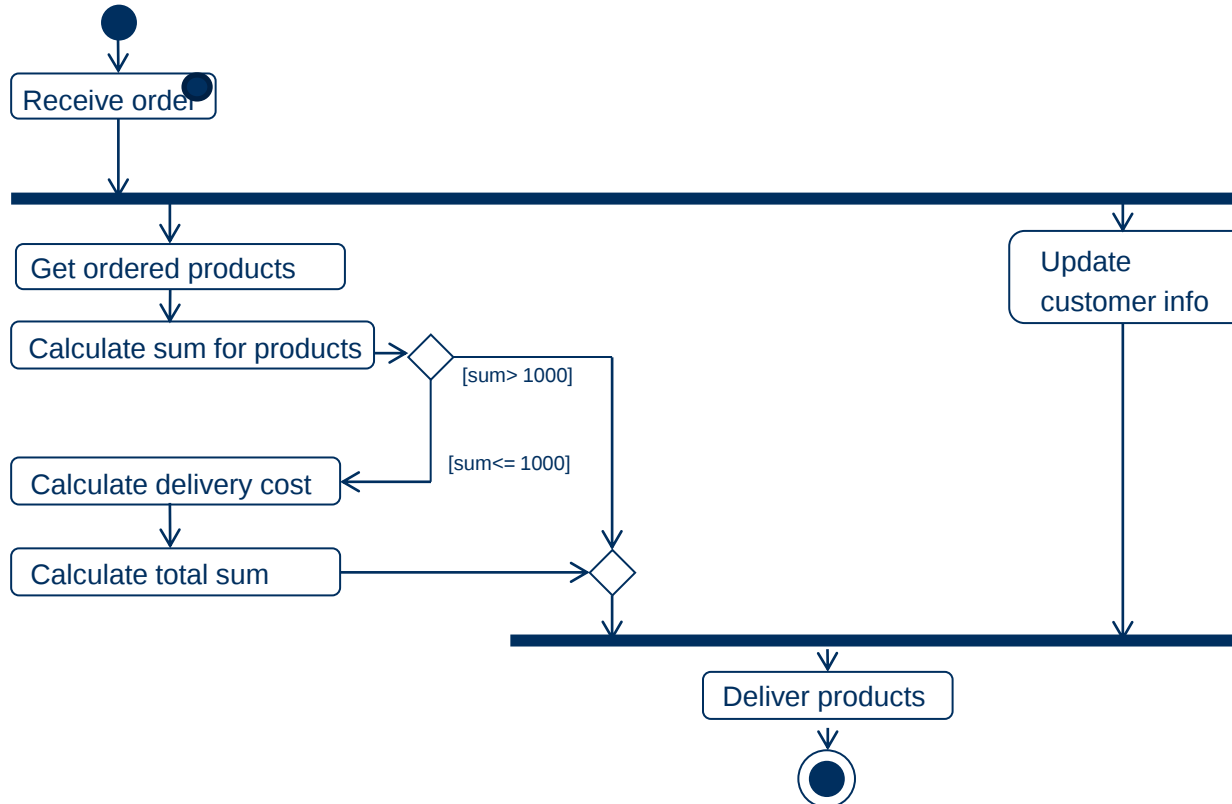
Token



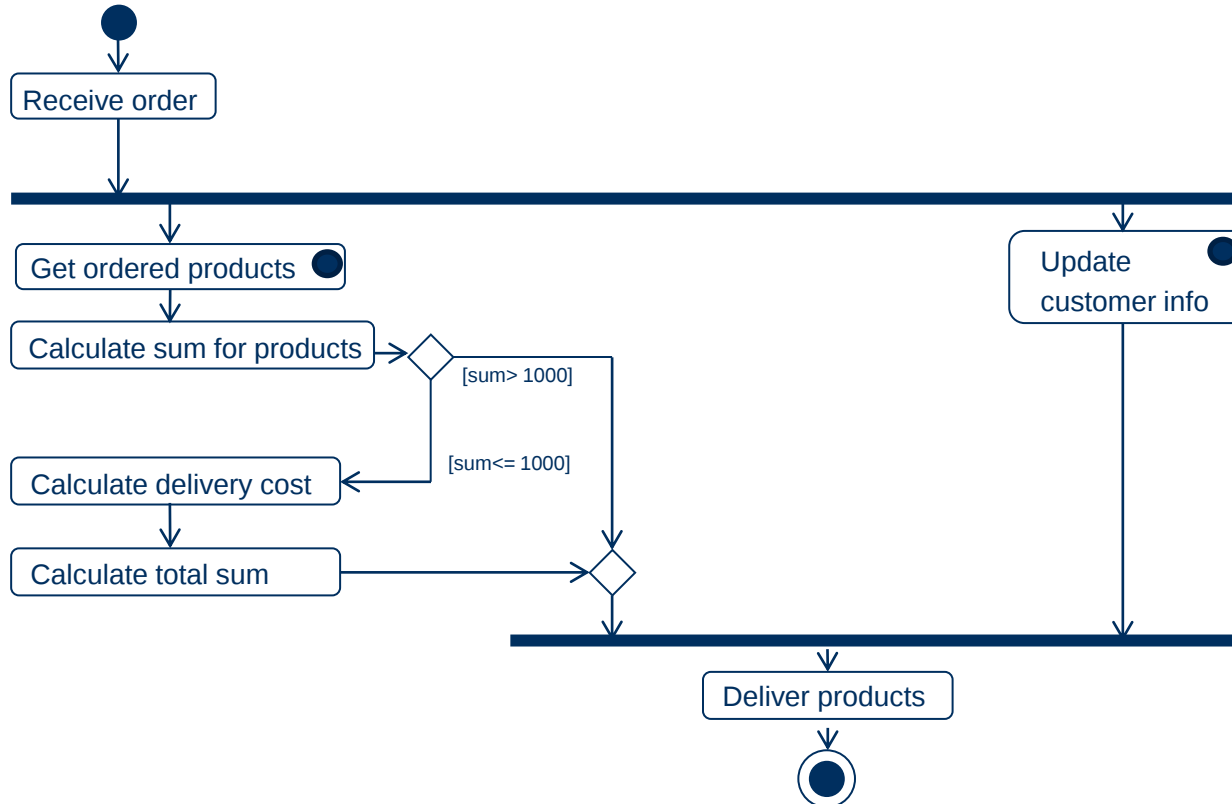
Token



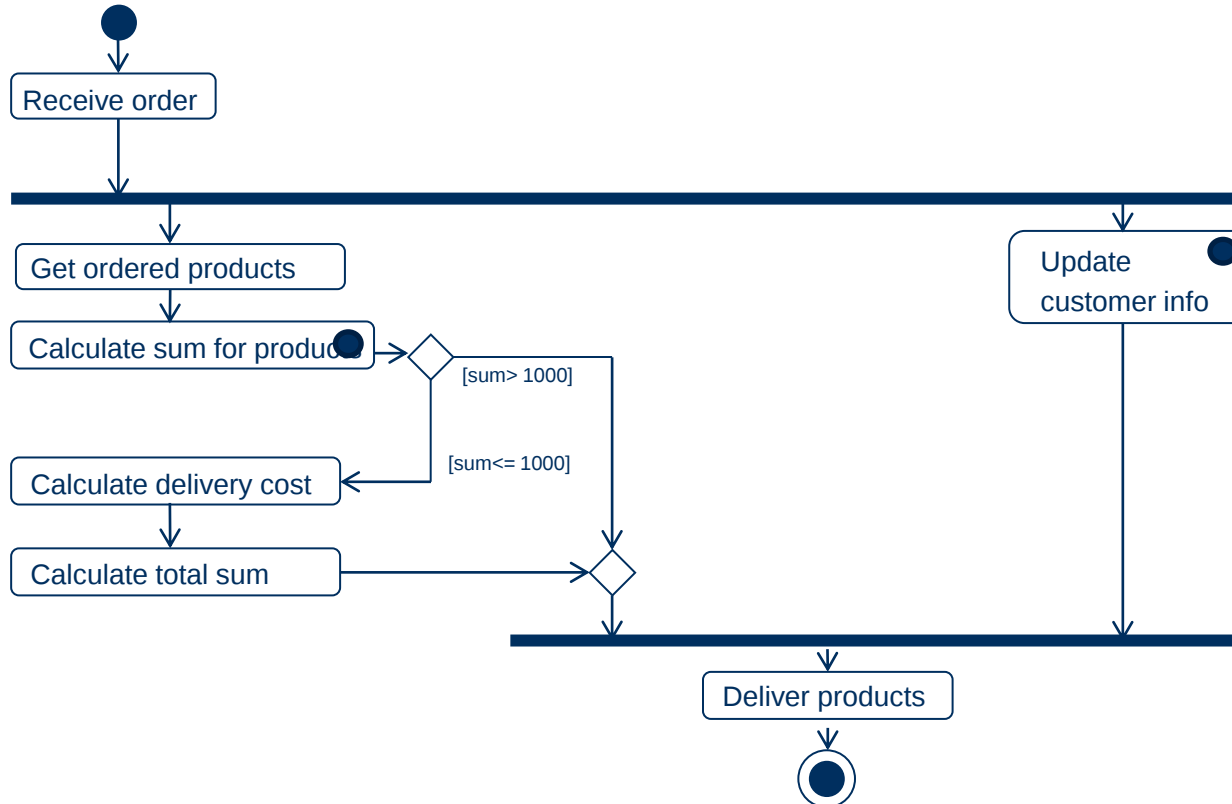
Token



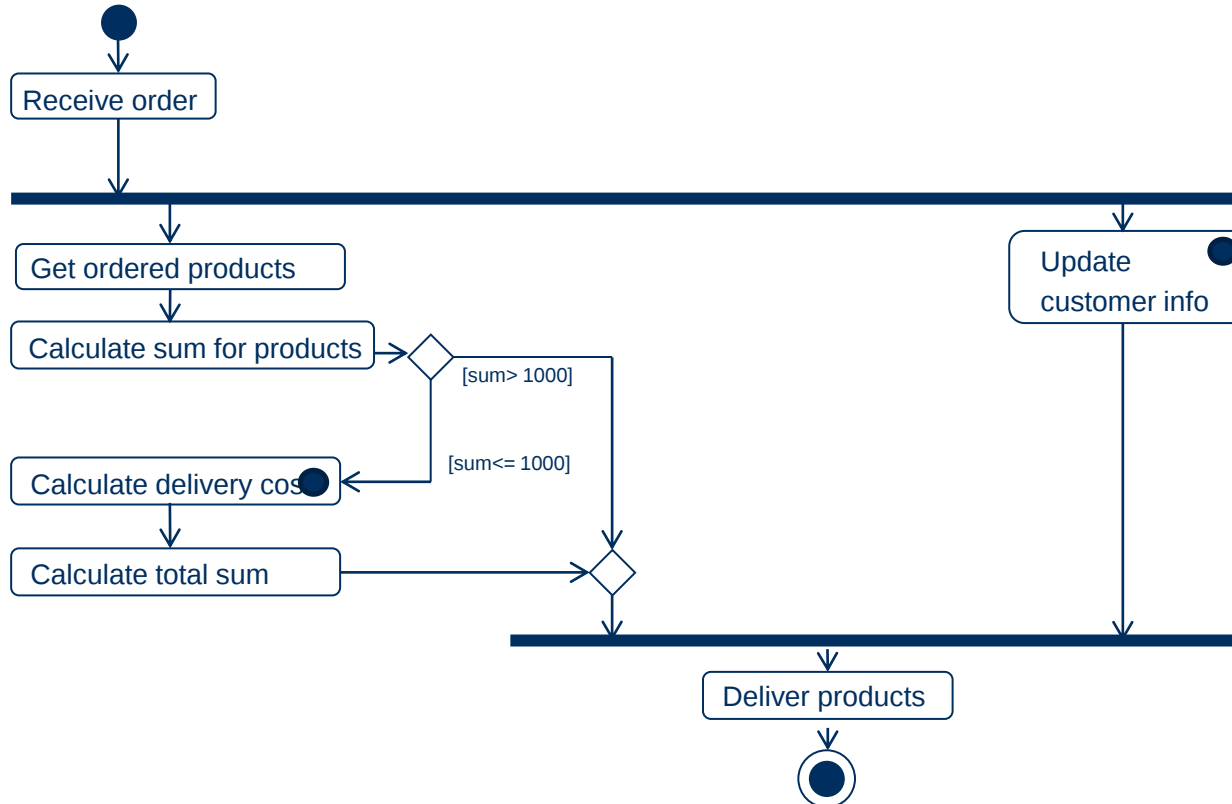
Token



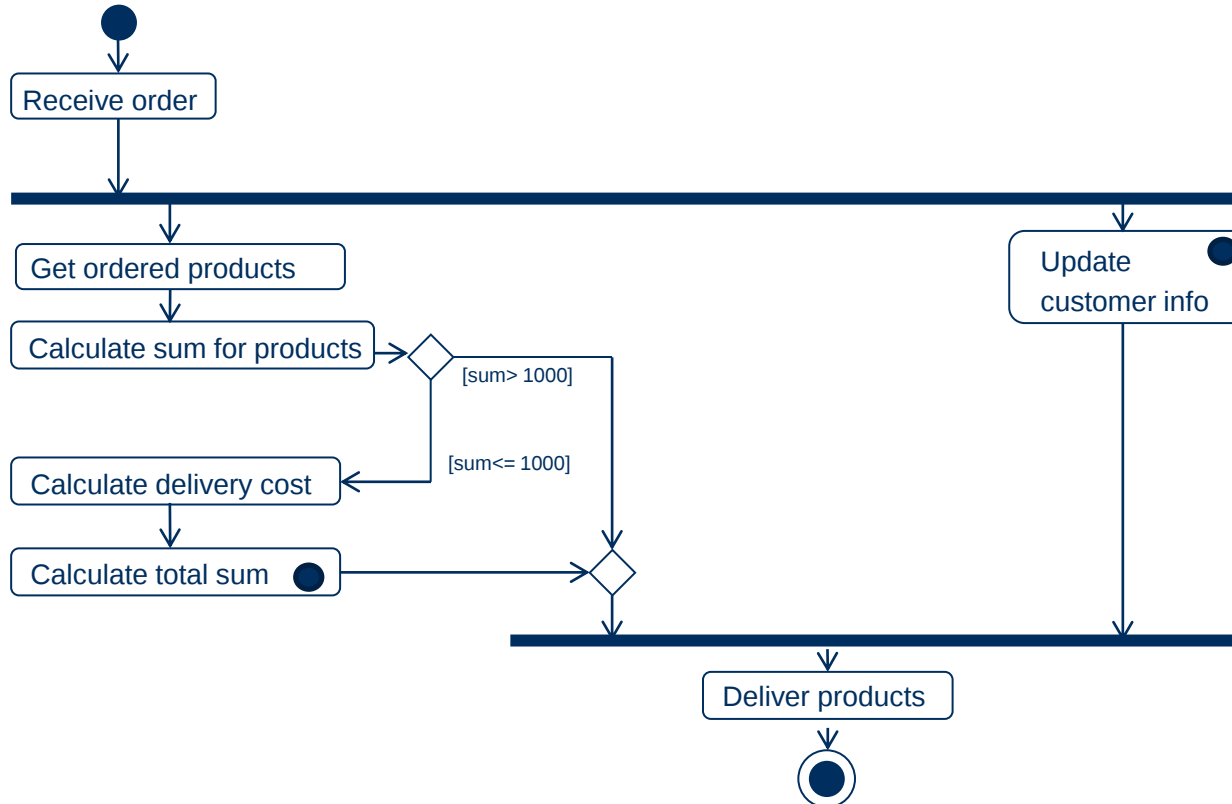
Token



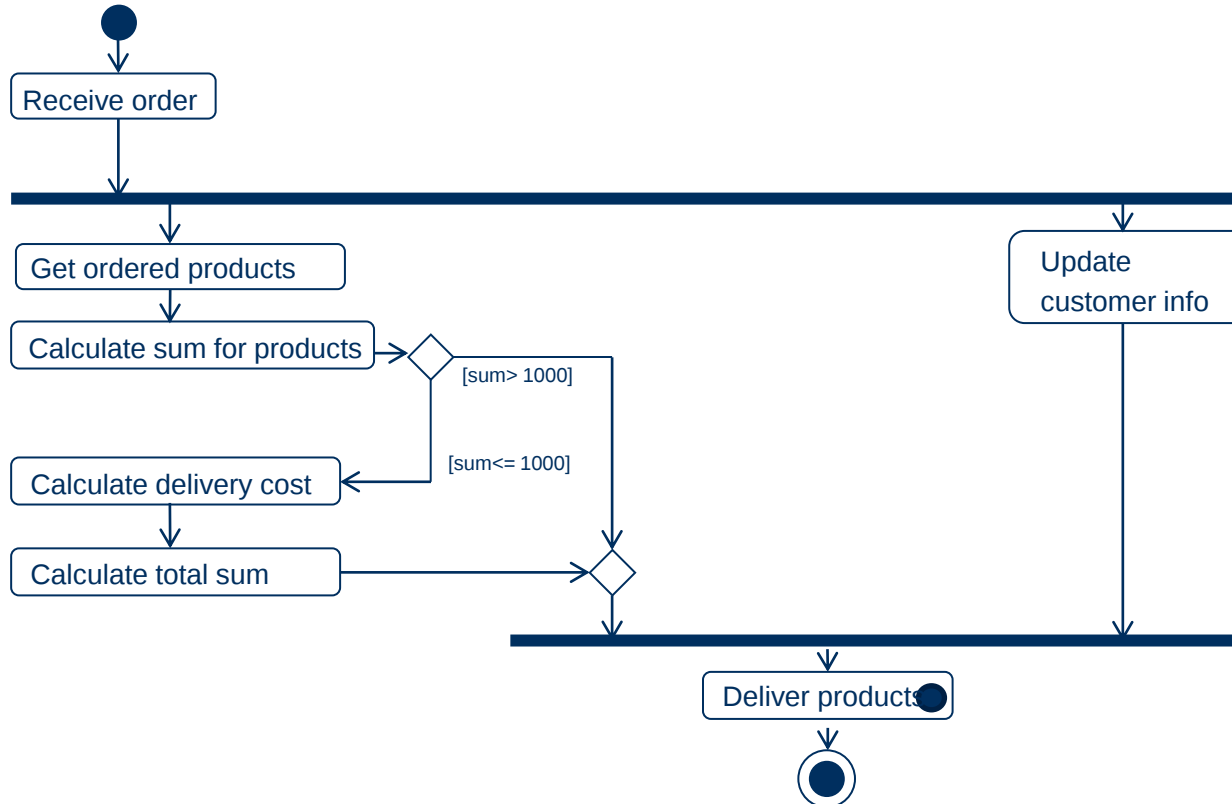
Token



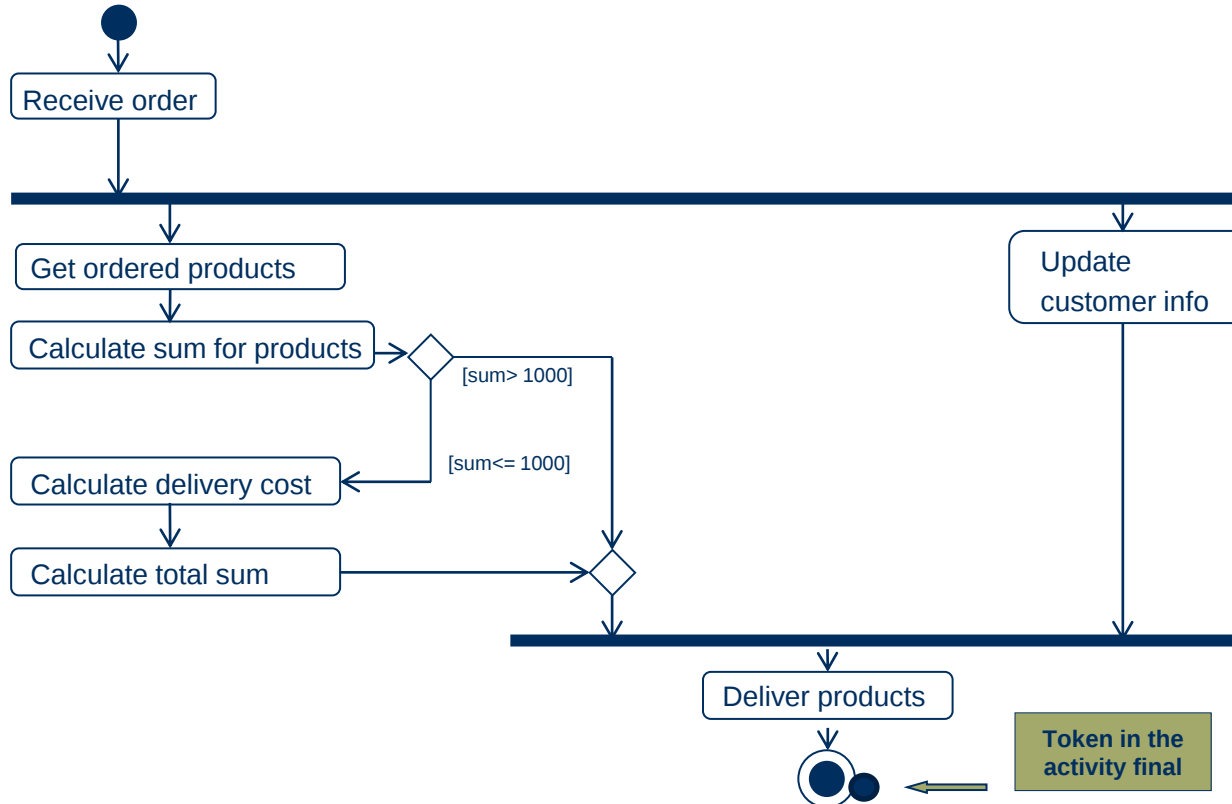
Token



Token

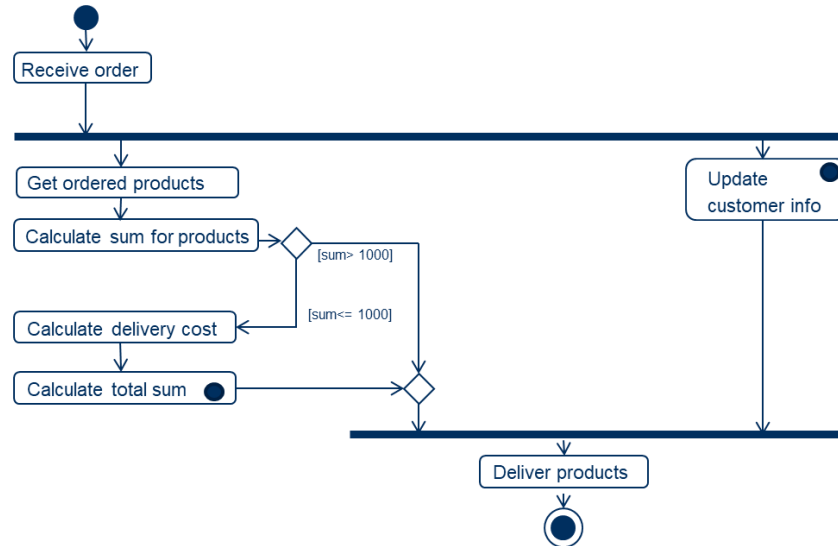


Token



Process instance revisited

- A token could make the concept of process instance more understandable
- The token can be used for describing the performance of a certain process instance

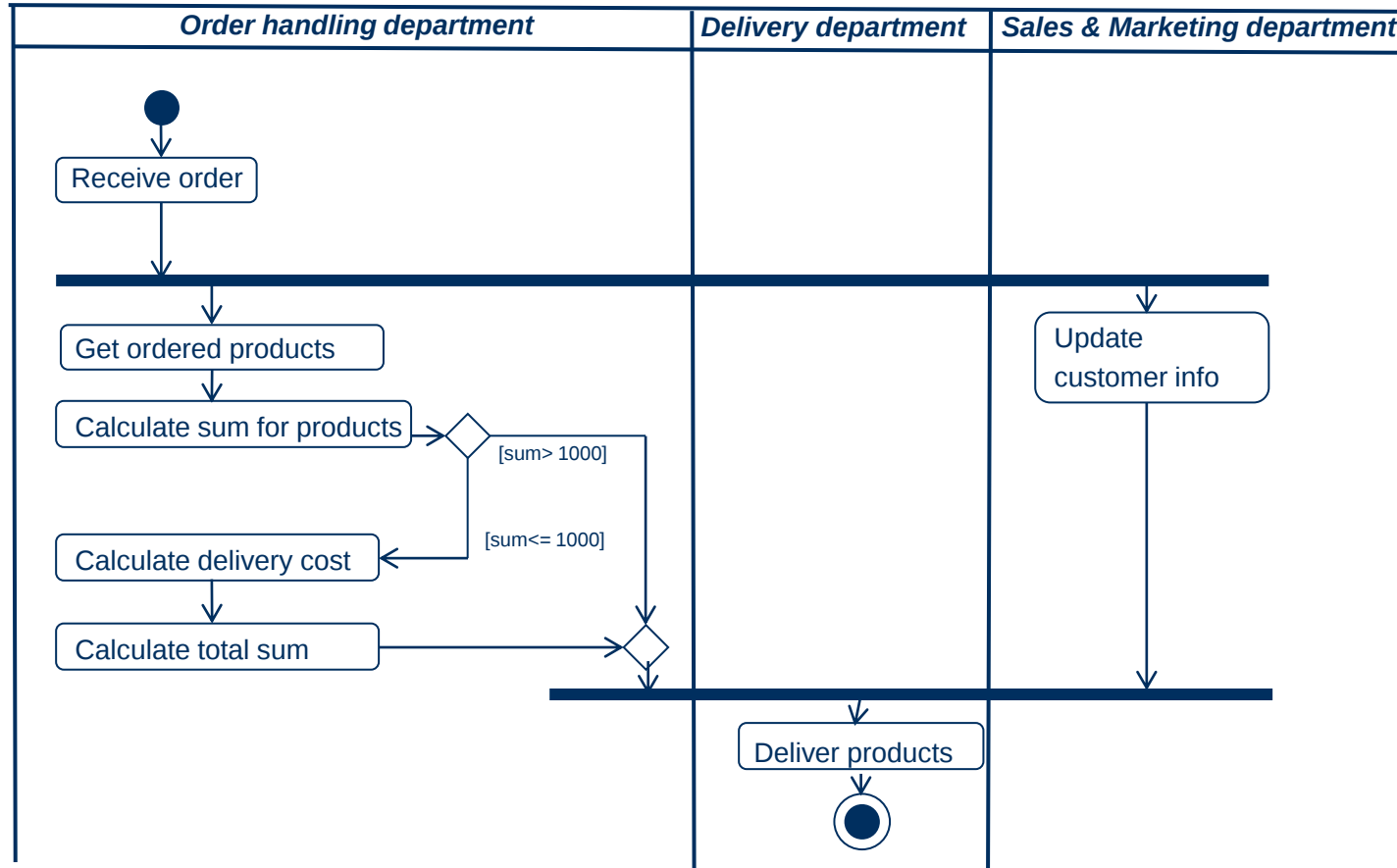


Process instance revisited

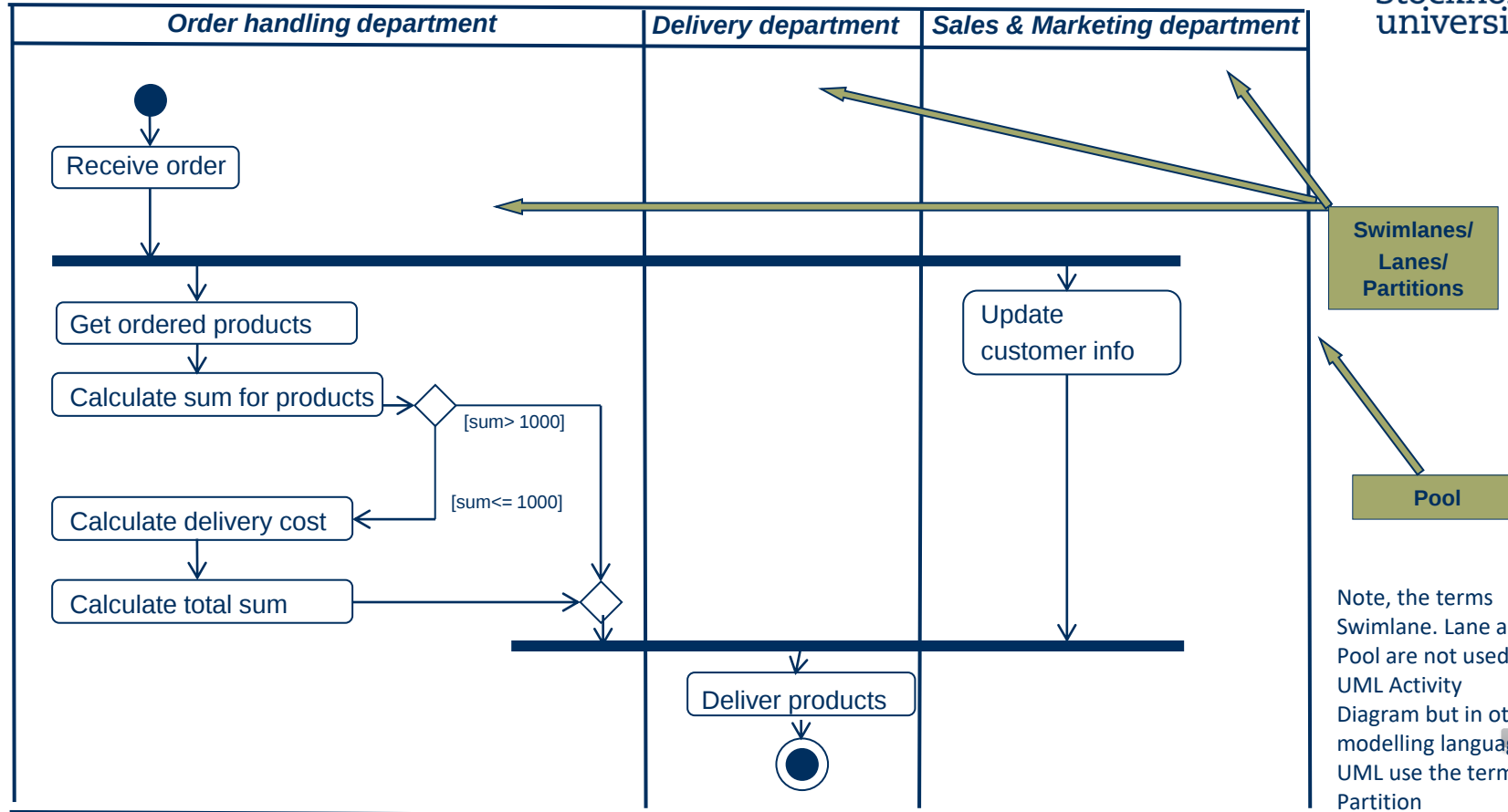
- Another way to describe a process instance could be by showing the order of actions and activities carried out by the instance, and which date and time they were carried out, as well as other data related to the process instance, see figure right



Swimlanes/Lanes/Partitions

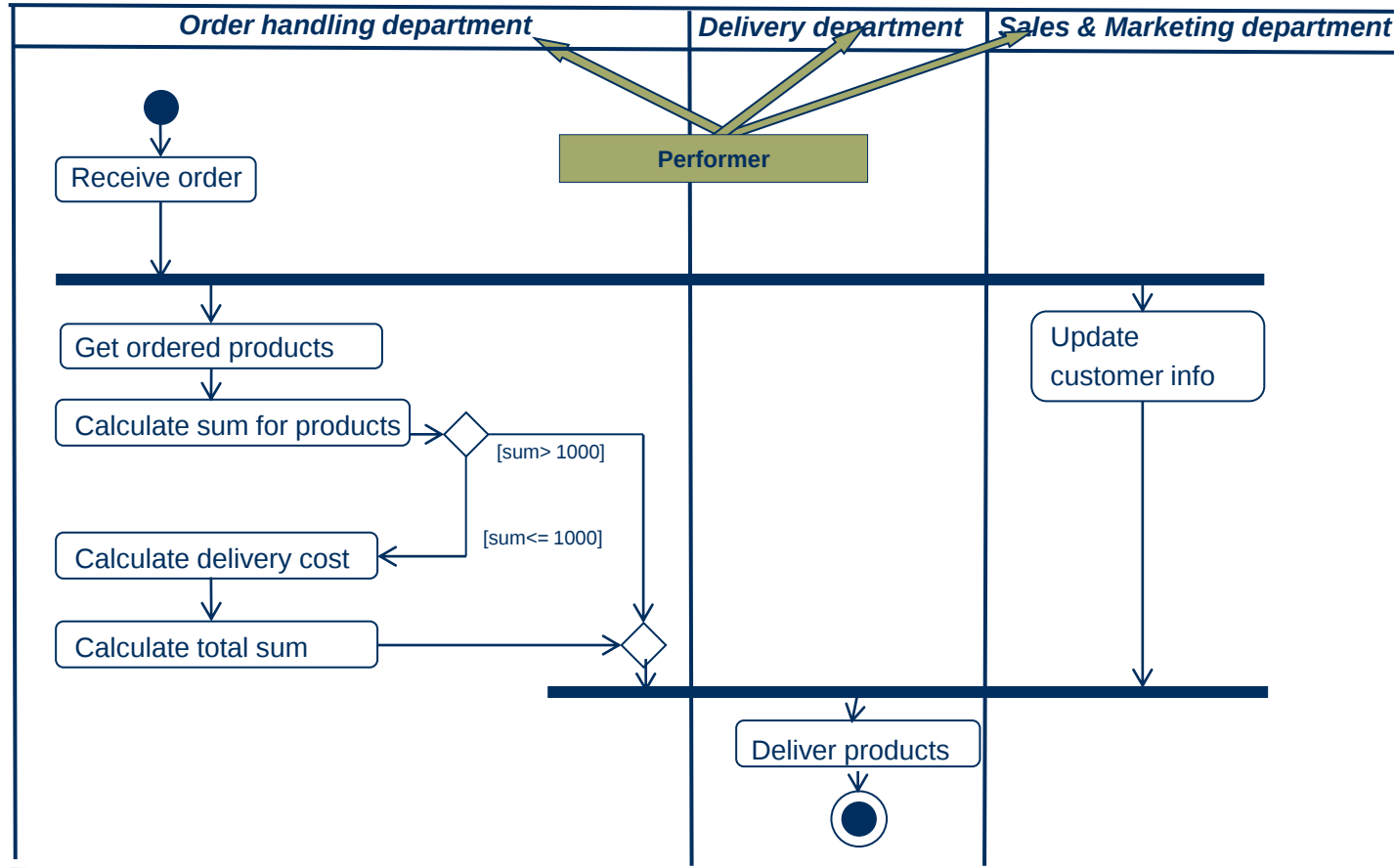


Swimlanes/Lanes/Partitions

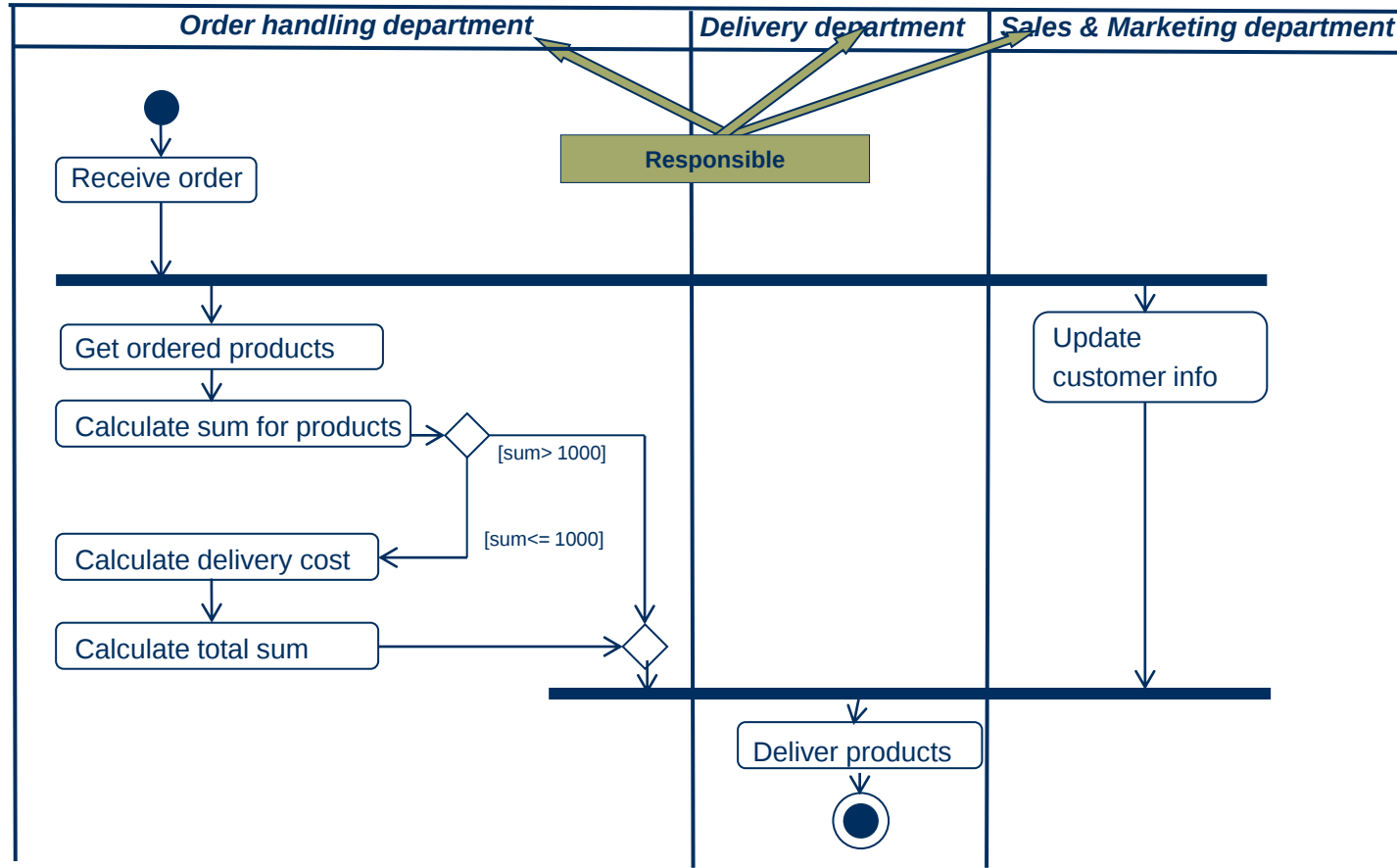


Note, the terms Swimlane. Lane and Pool are not used in UML Activity Diagram but in other modelling languages. UML use the term Partition

Swimlanes/Lanes Partitions



Swimlanes/Lanes Partitions



Question to answer

- How do you model business processes using a formal business process modelling language, in this case UML Activity Diagram?

