Activity Diagram

Activity diagram

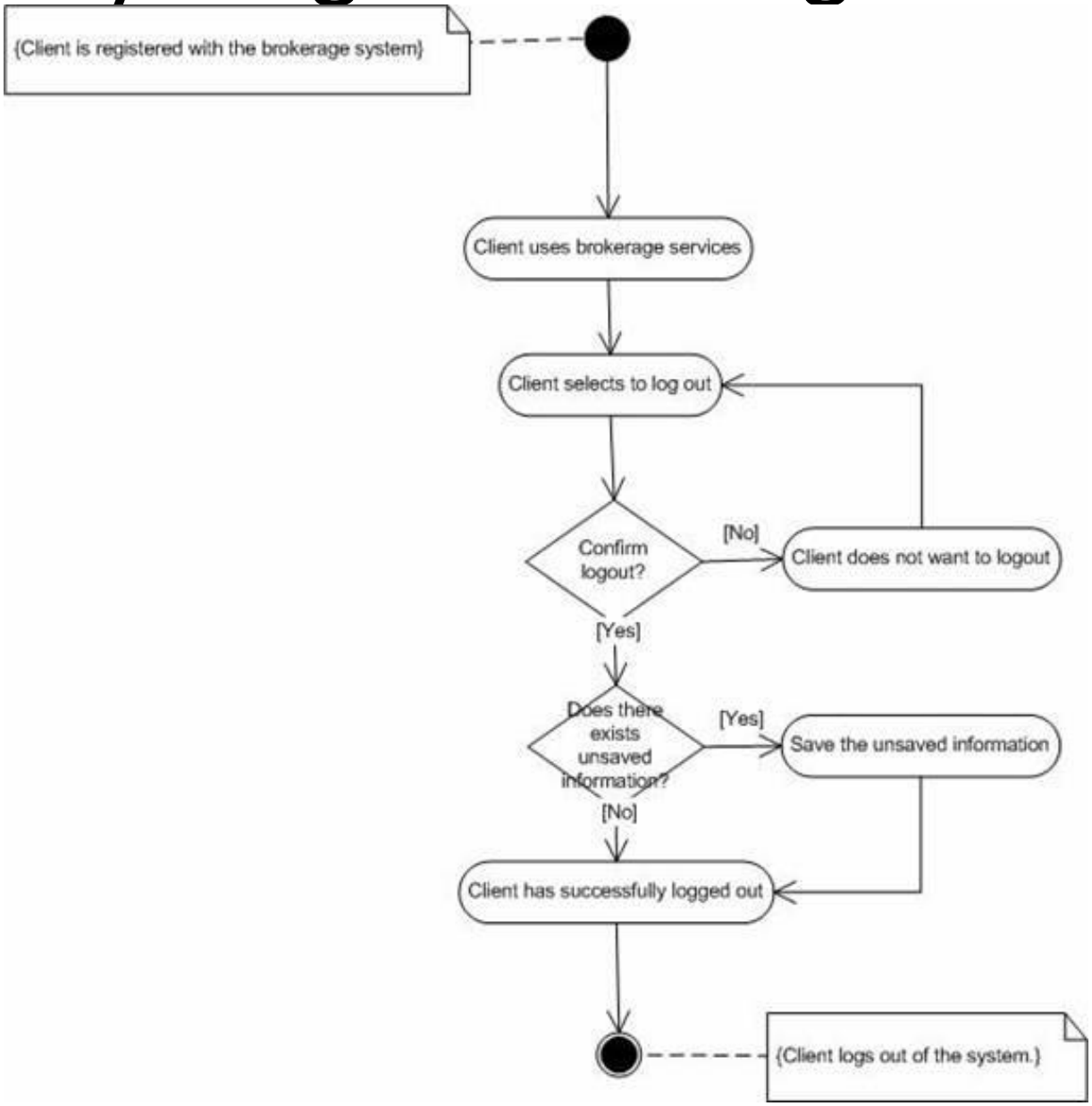
- **Activity diagram** is another important **diagram** in UML to describe the dynamic aspects of the system. **Activity diagram** is basically a flowchart to represent the flow from one **activity** to another **activity**. The **activity** can be described as an operation of the system. The control flow is drawn from one operation to another.

Activity Diagram

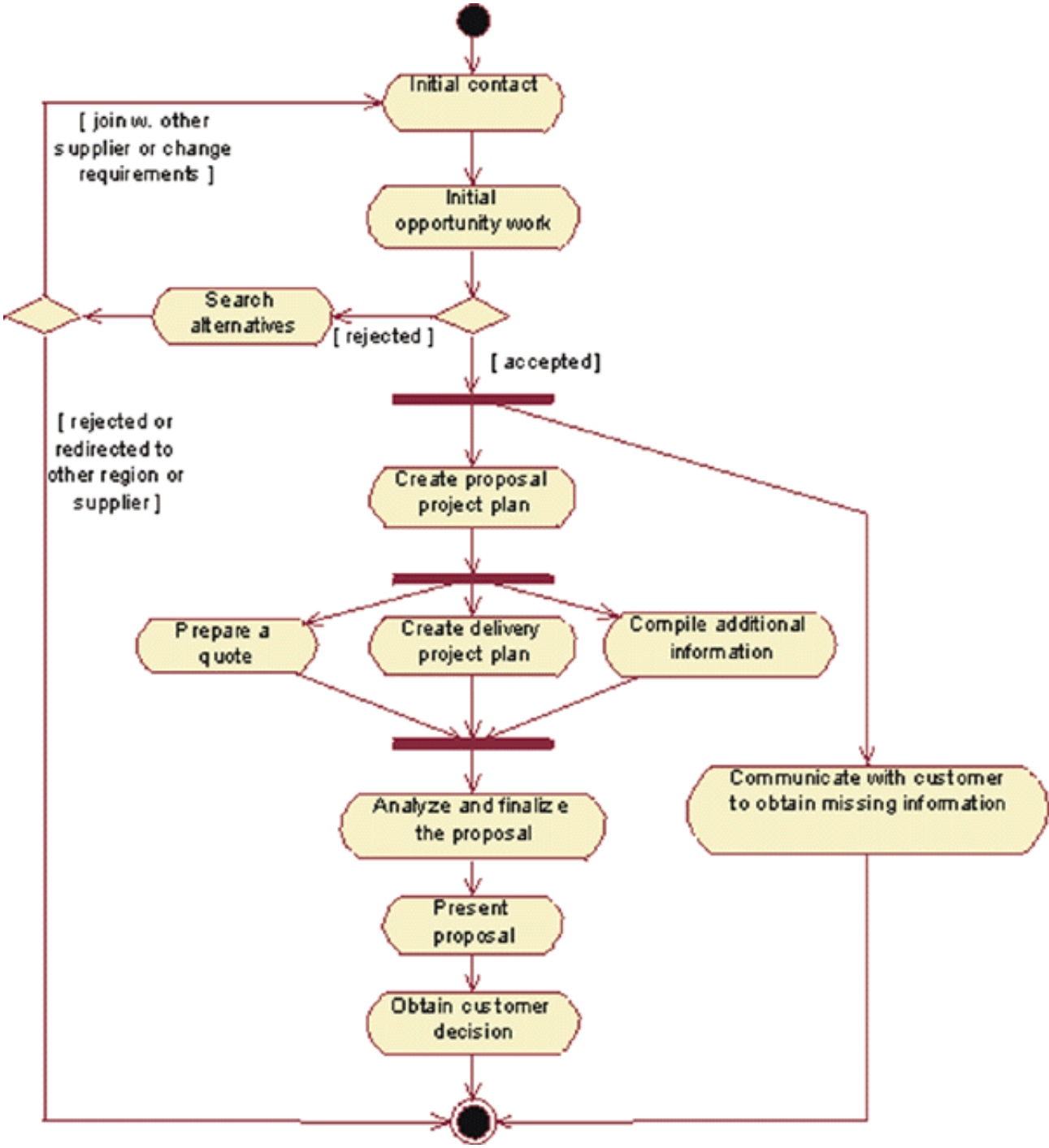
Activity diagrams are constructed from a limited number of shapes. The most important shape types:

- *rounded rectangles* represent *actions*;
- *diamonds* represent *decisions*;
- *bars* represent the start (*split*) or end (*join*) of concurrent activities;
- a *black circle* represents the start (*initial state*) of the workflow;
- an *encircled black circle* represents the end (*final state*).
- Arrows run from the start towards the end and represent the order in which activities happen.

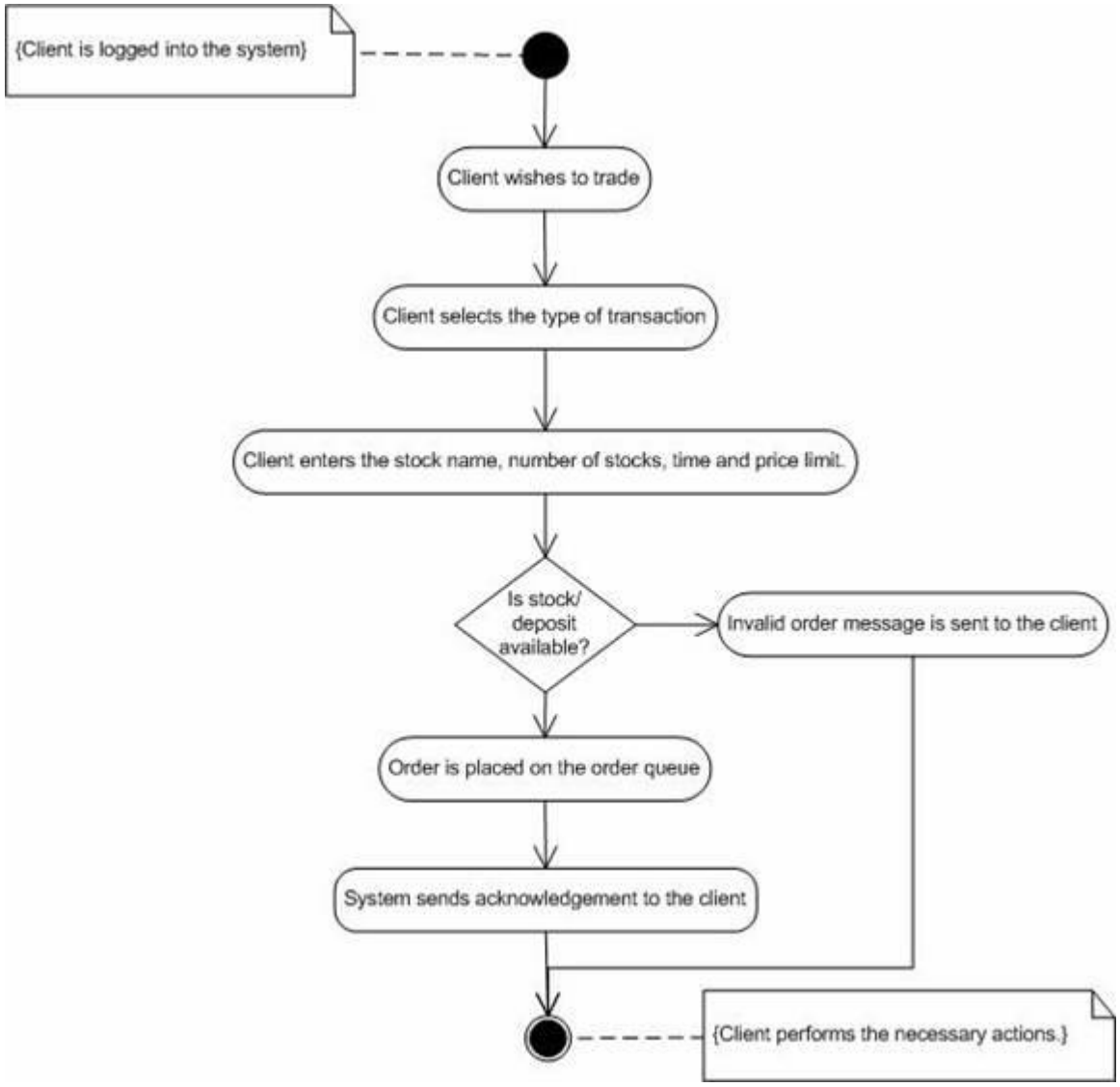
Activity Diagram for Logout



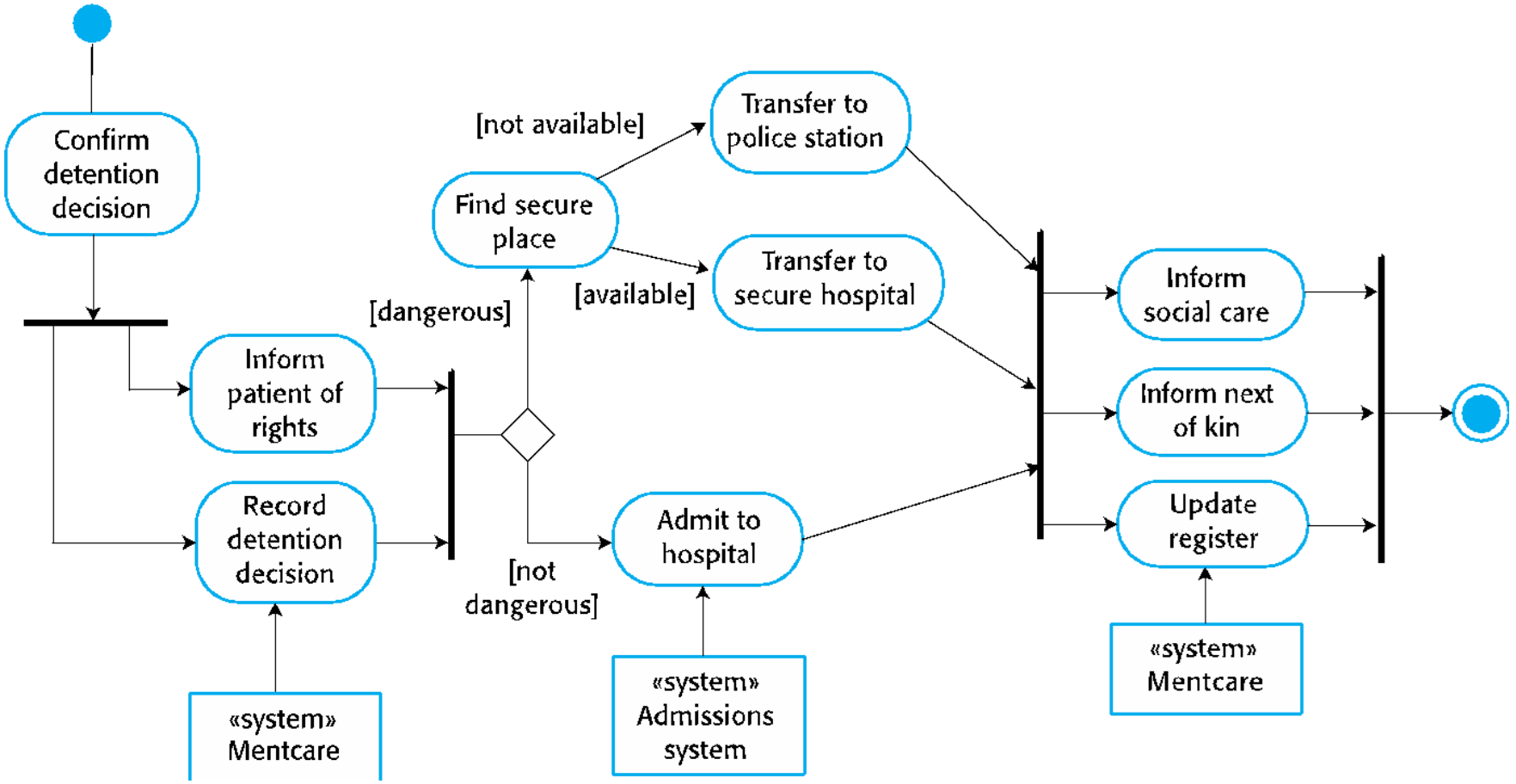
Activity Diagram for getting Project Order



Activity Diagram for Placing an Order



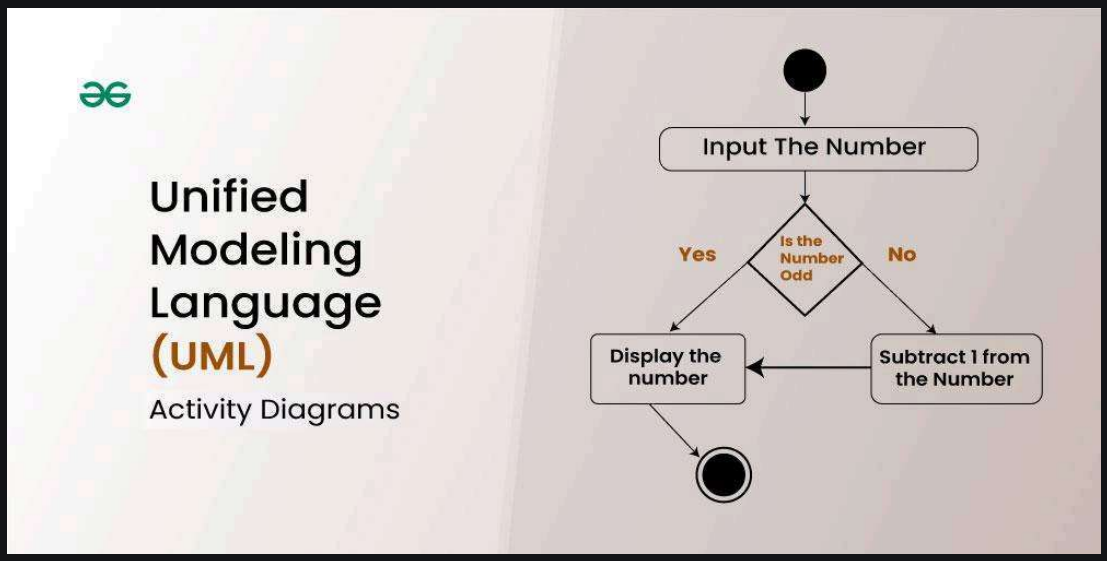
Process model (/ activity diagram) of involuntary detention



Activity Diagrams | Unified Modeling Language (UML)

Last Updated : 15 Jan, 2024

Activity Diagrams are used to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. It is a type of behavioral diagram and we can depict both sequential processing and concurrent processing of activities using an activity diagram ie an activity diagram focuses on the condition of flow and the sequence in which it happens.



Important Topics for the Activity Diagrams

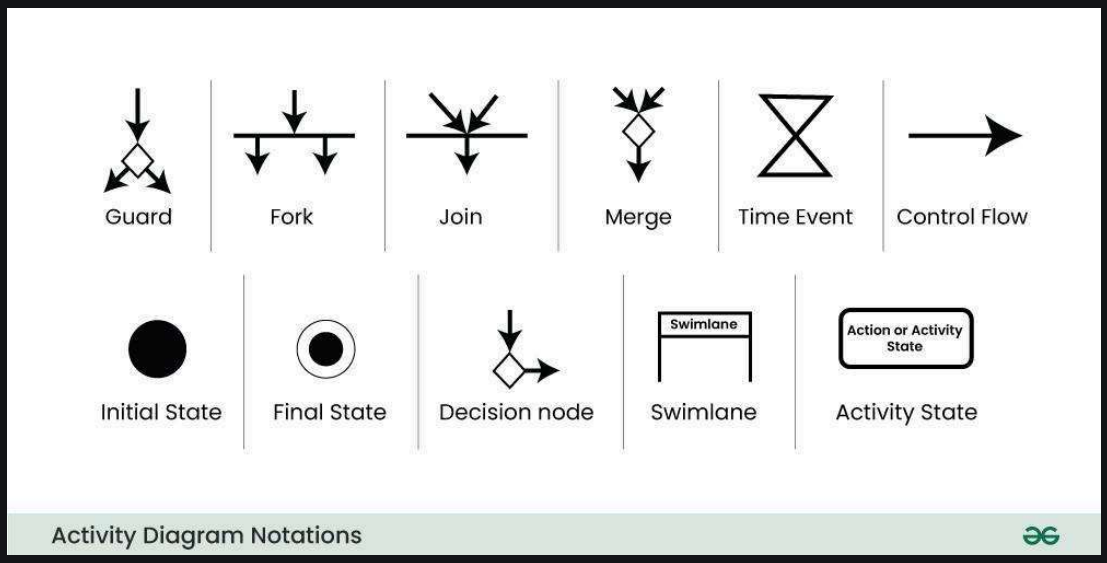
- What is an Activity Diagram?
- Activity Diagram Notations
- How to Draw an Activity Diagram in UML?
- What are Activity Diagrams used for?
- What are the Differences between an Activity diagram and a Flowchart?

1. What is an Activity Diagram?

Activity Diagrams are used to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We can depict both sequential processing and concurrent processing of activities using an activity diagram ie an activity diagram focuses on the condition of flow and the sequence in which it happens.

- We describe what causes a particular event using an activity diagram.
- An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.
- They are used in business and process modeling where their primary use is to depict the dynamic aspects of a system.

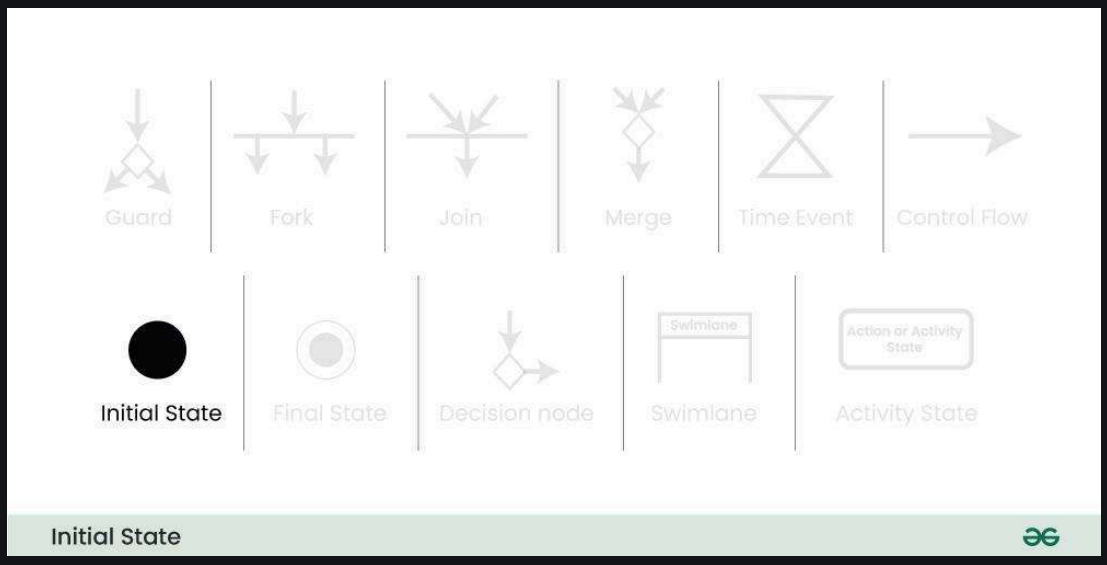
2. Activity Diagram Notations



In activity diagrams, the notations are like visual symbols that help represent different elements and actions in a simple way.

2.1. Initial State

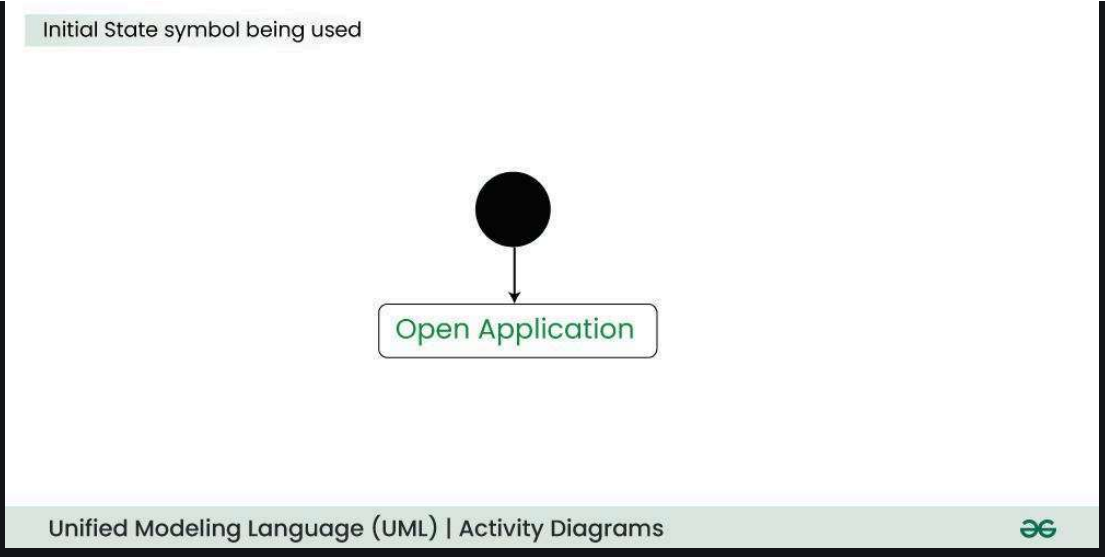
The starting state before an activity takes place is depicted using the initial state.



A process can have only one initial state unless we are depicting nested activities. We use a black filled circle to depict the initial state of a system. For objects, this is the state when they are instantiated. The Initial State from the UML Activity Diagram marks the entry point and the initial Activity State.

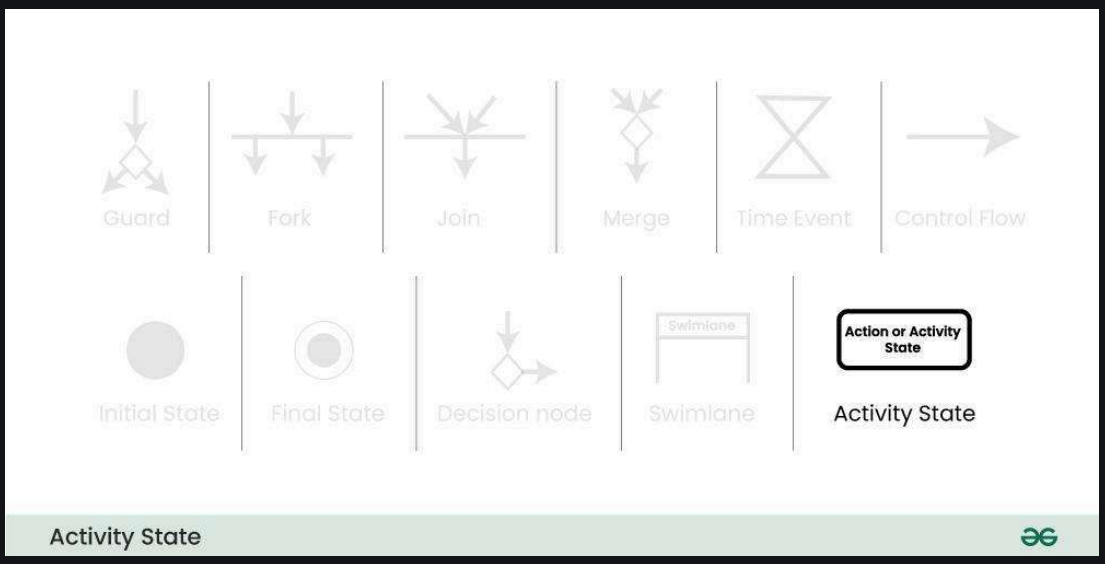
For example:

Here the initial state of the system before the application is opened.



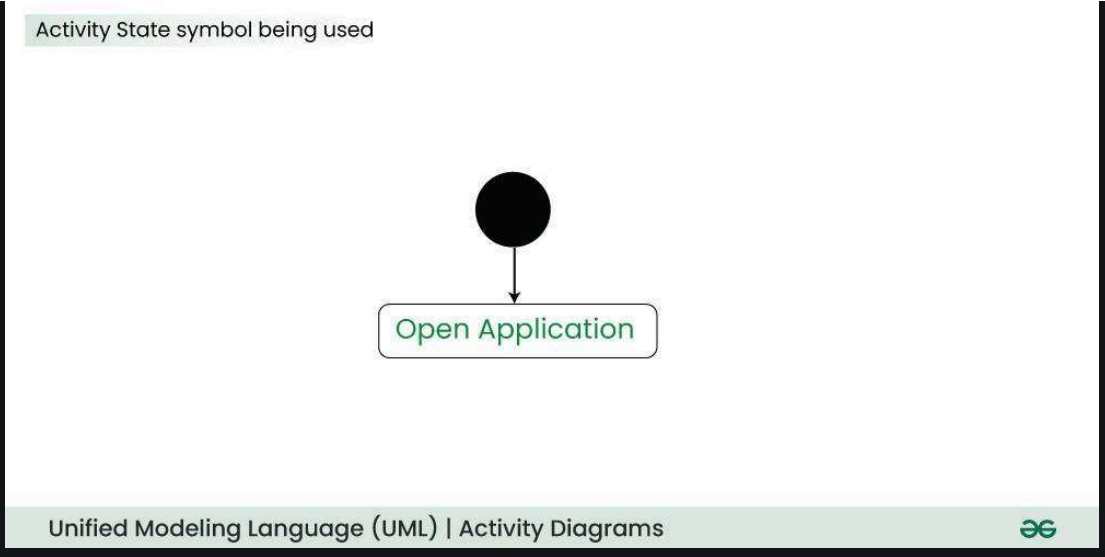
2.2. Action or Activity State

An activity represents execution of an action on objects or by objects. We represent an activity using a rectangle with rounded corners. Basically any action or event that takes place is represented using an activity.



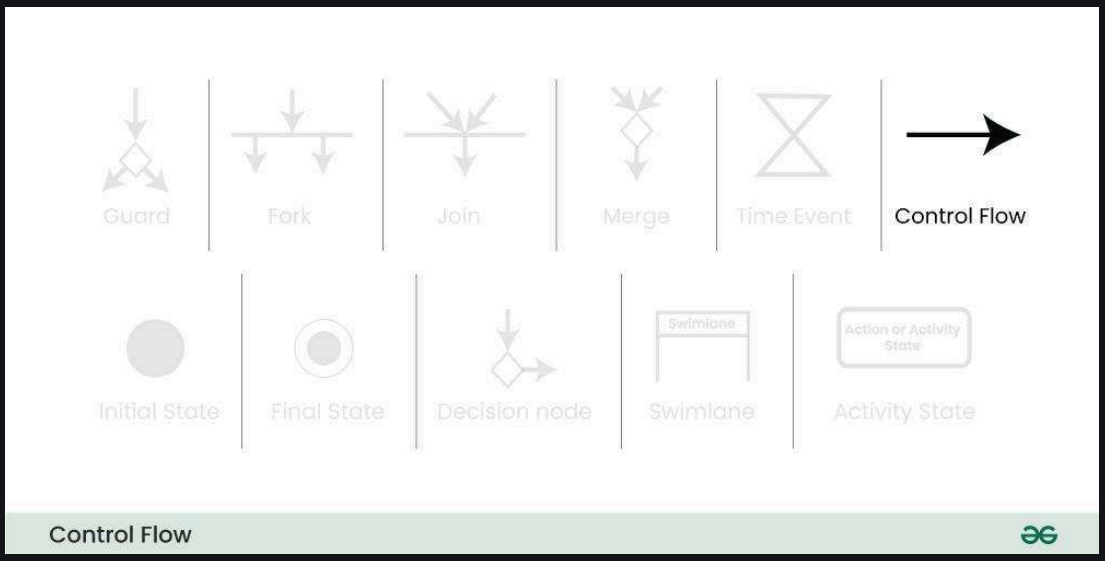
For example:

> *Consider the previous example of opening an application, opening the application is an activity state in the activity diagram.*



2.3. Action Flow or Control flows

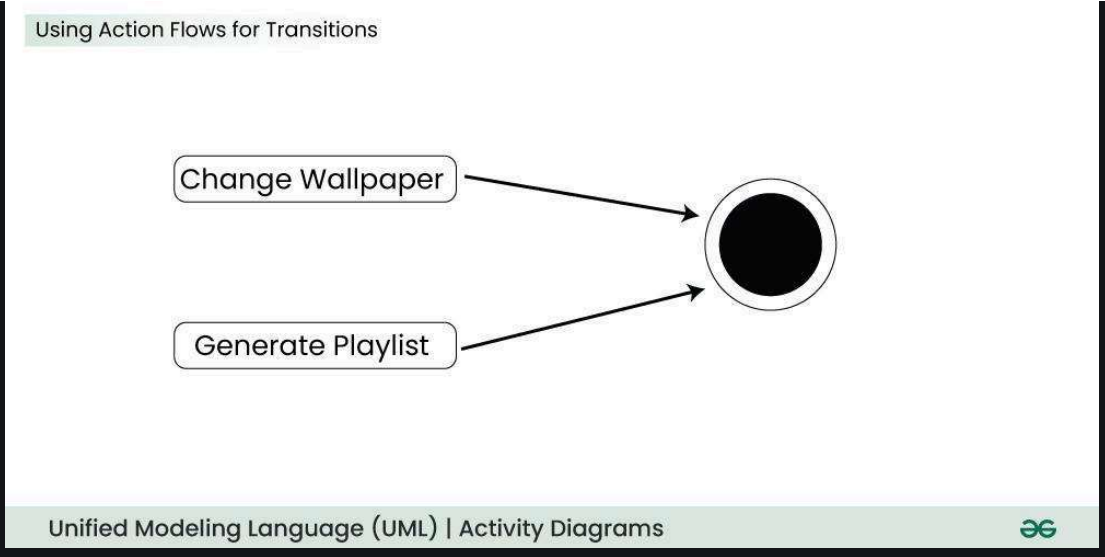
Action flows or Control flows are also referred to as paths and edges. They are used to show the transition from one activity state to another activity state.



An activity state can have multiple incoming and outgoing action flows. We use a line with an arrow head to depict a Control Flow. If there is a constraint to be adhered to while making the transition it is mentioned on the arrow.

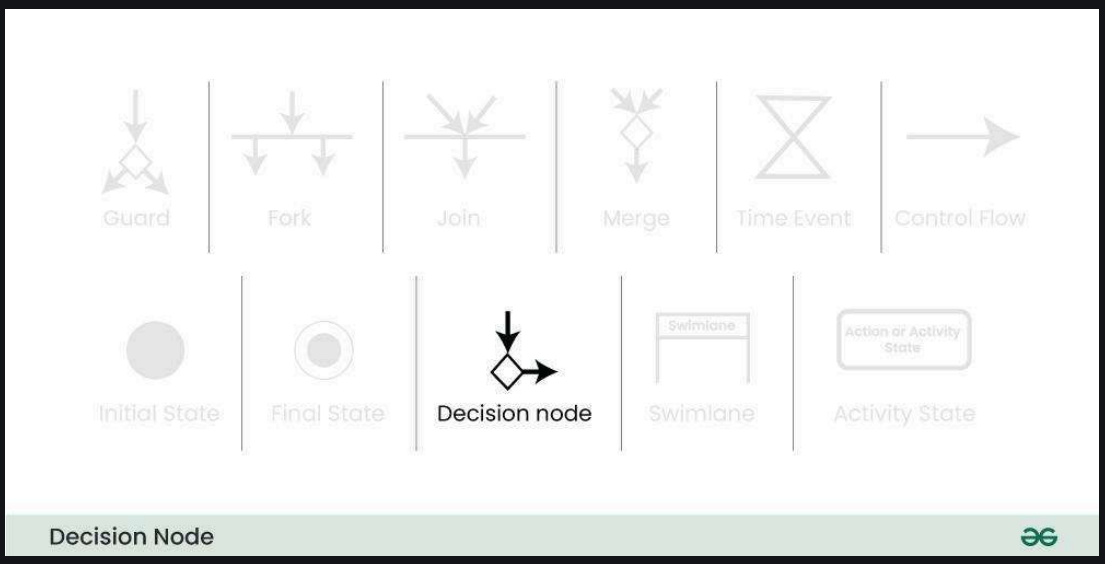
For example:

Here both the states transit into one final state using action flow symbols i.e. arrows.



2.4. Decision node and Branching

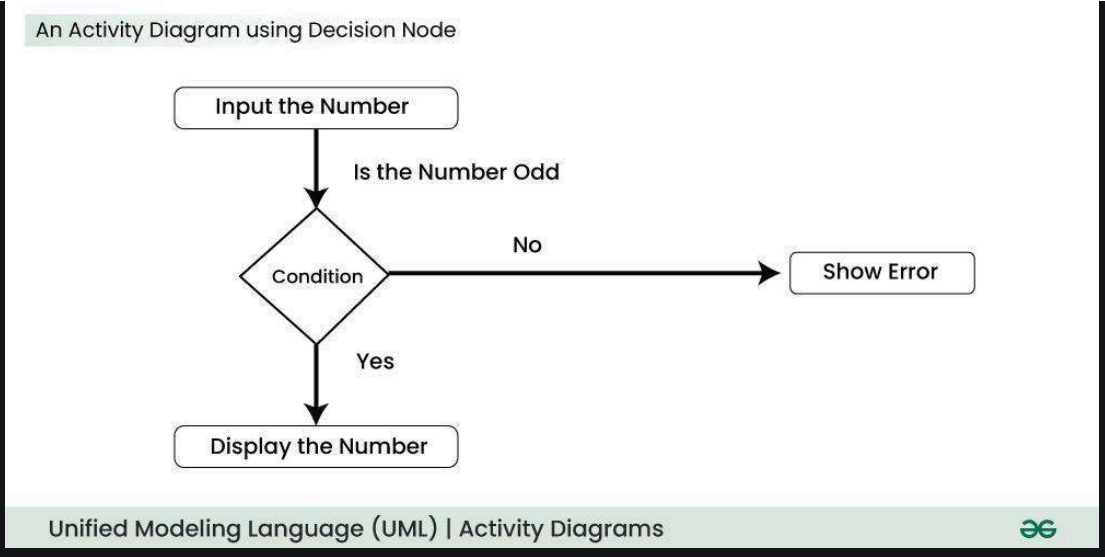
When we need to make a decision before deciding the flow of control, we use the decision node. The outgoing arrows from the decision node can be labelled with conditions or guard expressions. It always includes two or more output arrows.



For example:

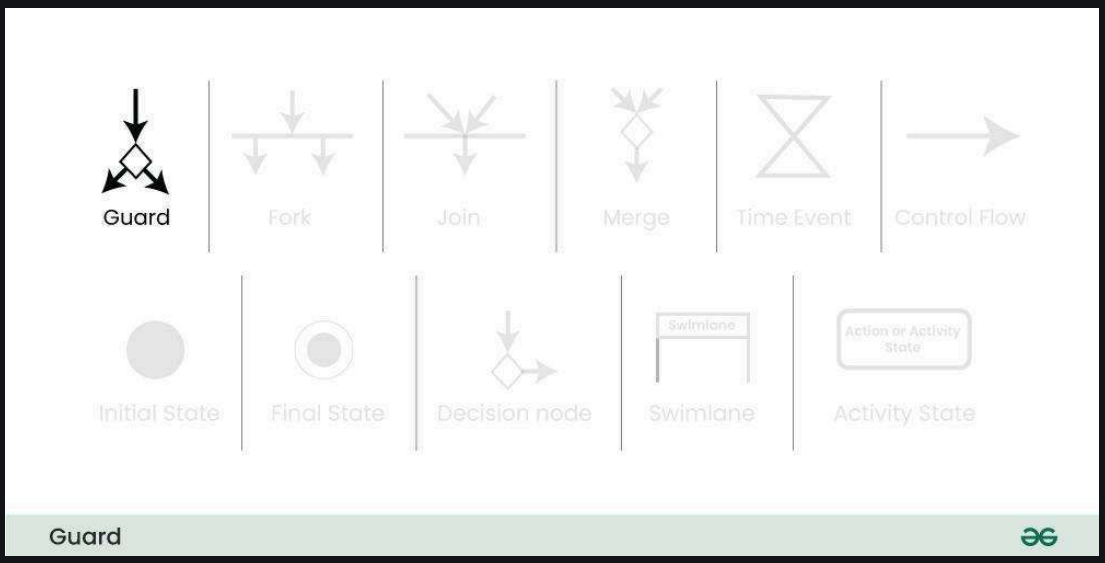
We apply the conditions on input number to display the result :

- *If number is odd then display the number.*
- *If number if even then display the error.*



2.5. Guard

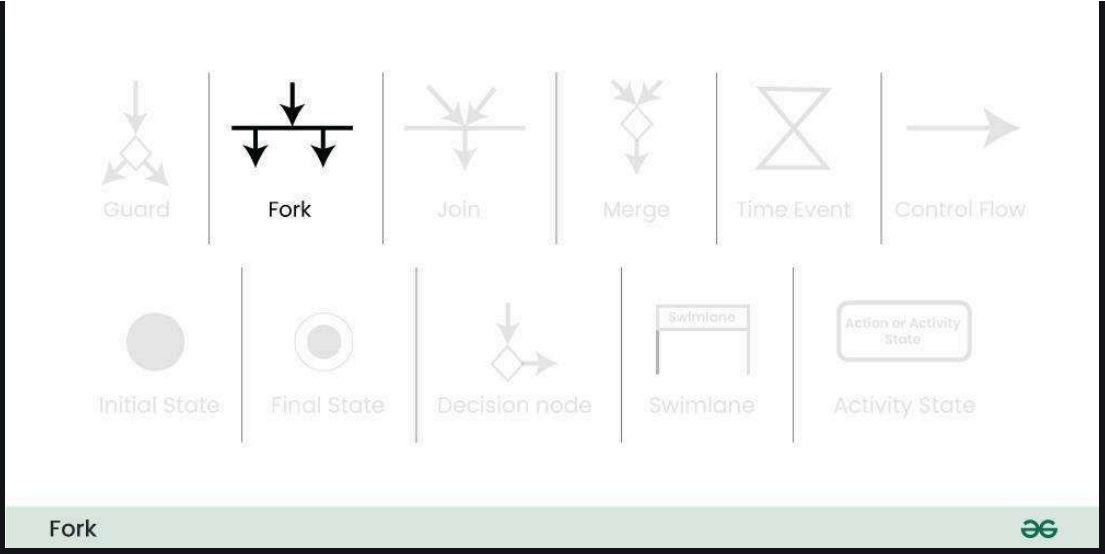
A Guard refers to a statement written next to a decision node on an arrow sometimes within square brackets.



The statement must be true for the control to shift along a particular direction. Guards help us know the constraints and conditions which determine the flow of a process.

2.6. Fork

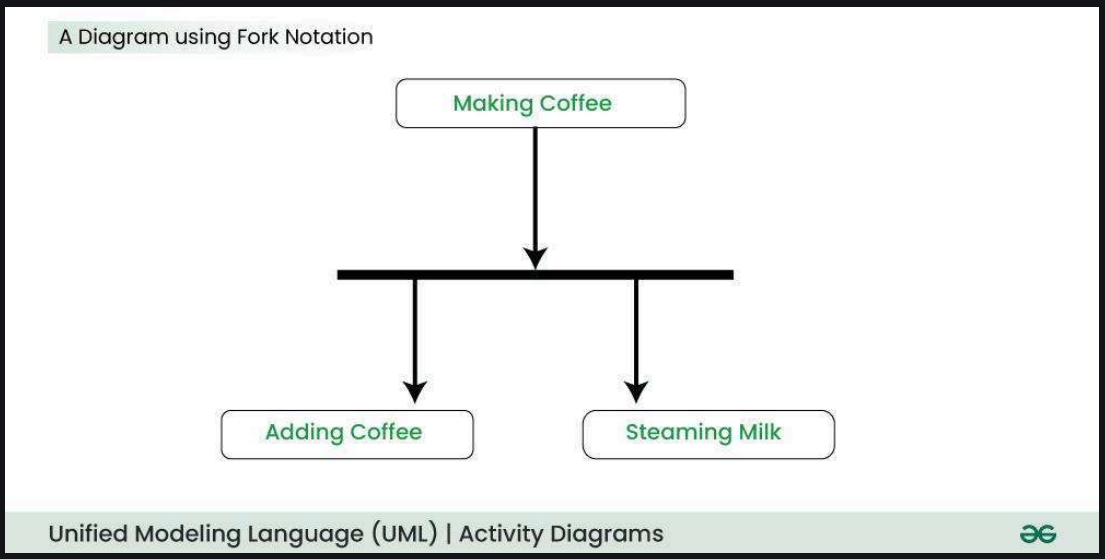
Fork nodes are used to support concurrent activities. When we use a fork node when both the activities get executed concurrently i.e. no decision is made before splitting the activity into two parts. Both parts need to be executed in case of a fork statement. We use a rounded solid rectangular bar to represent a Fork notation with incoming arrow from the parent activity state and outgoing arrows towards the newly created activities.



Fork

For example:

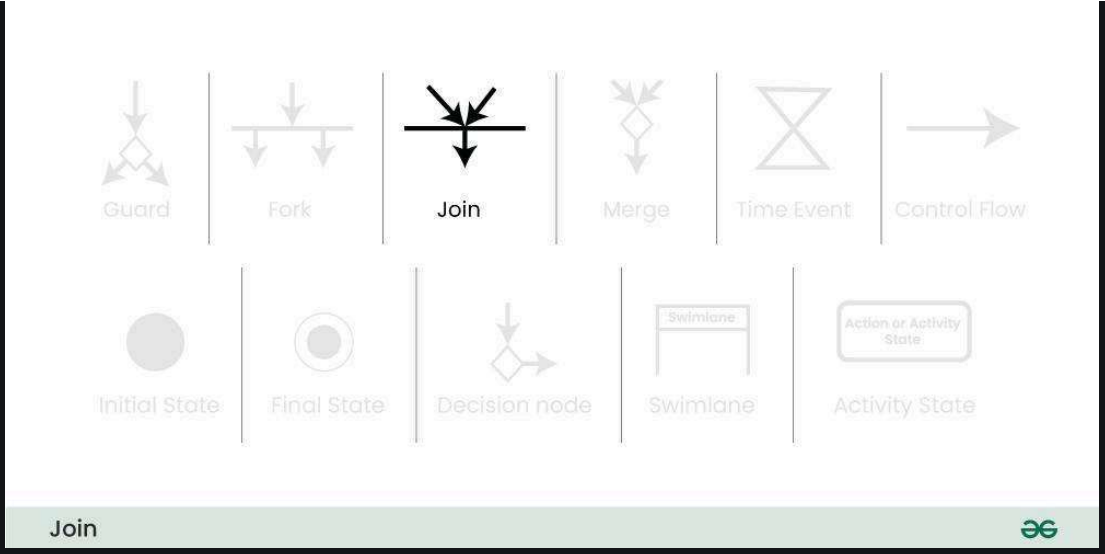
In the example below, the activity of making coffee can be split into two concurrent activities and hence we use the fork notation.



Unified Modeling Language (UML) | Activity Diagrams

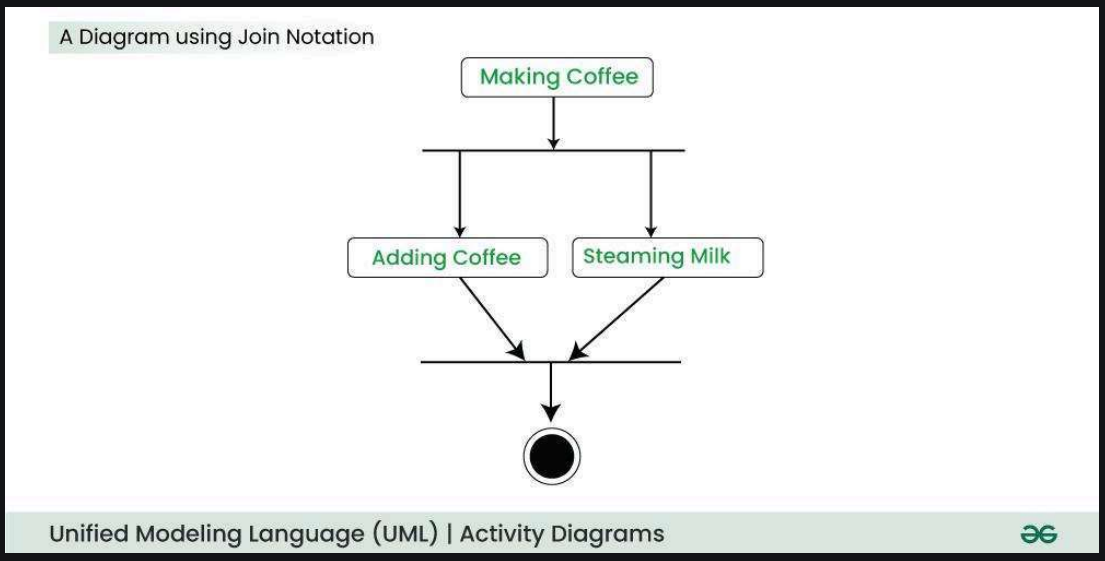
2.7. Join

Join nodes are used to support concurrent activities converging into one. For join notations we have two or more incoming edges and one outgoing edge.



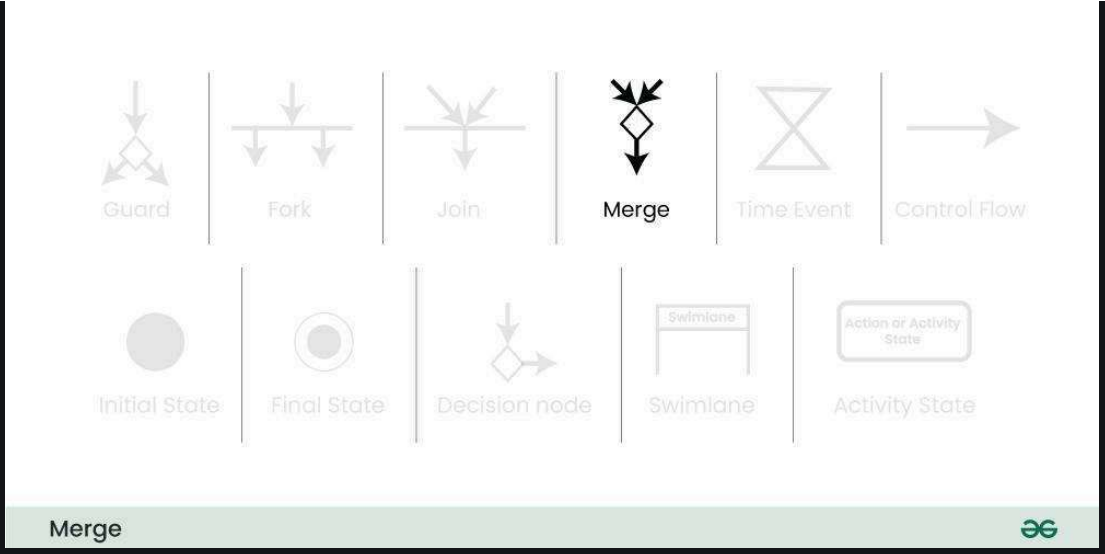
For example:

When both activities i.e. steaming the milk and adding coffee get completed, we converge them into one final activity.



2.8. Merge or Merge Event

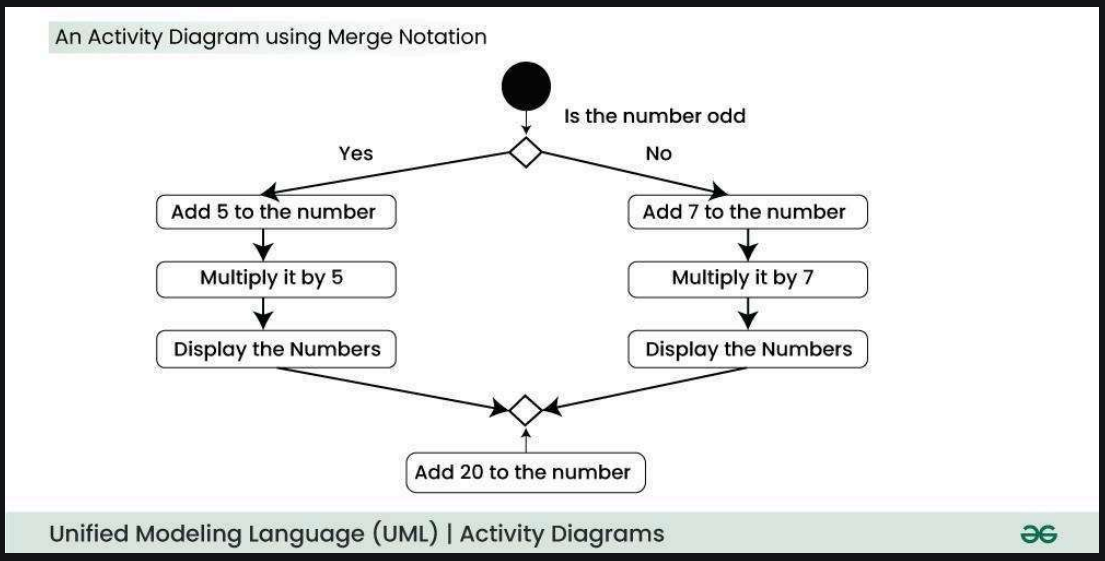
Scenarios arise when activities which are not being executed concurrently have to be merged. We use the merge notation for such scenarios. We can merge two or more activities into one if the control proceeds onto the next activity irrespective of the path chosen.



Merge

For example:

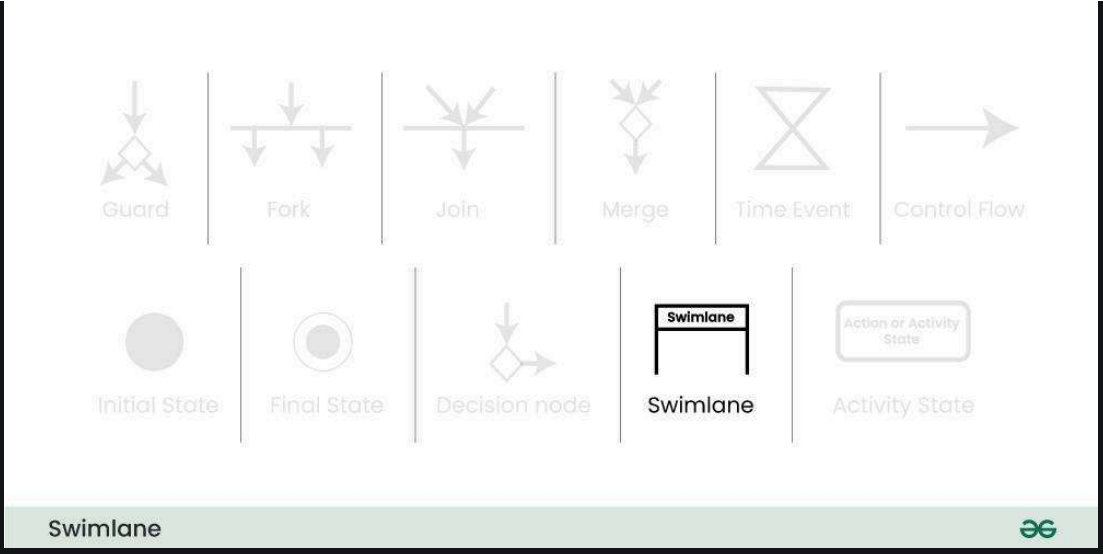
In the diagram below: we can't have both sides executing concurrently, but they finally merge into one. A number can't be both odd and even at the same time.



Unified Modeling Language (UML) | Activity Diagrams

2.9. Swimlanes

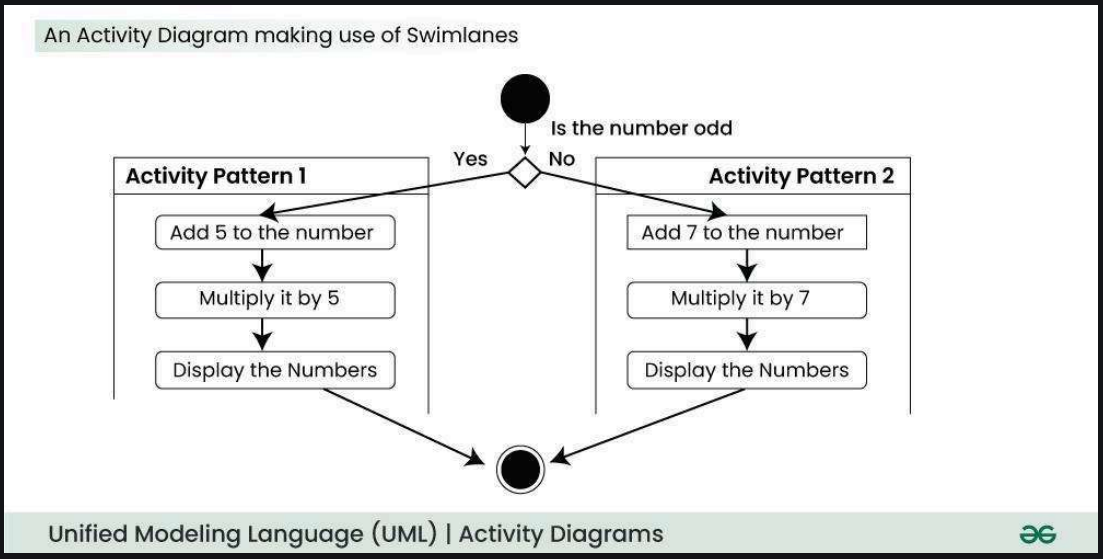
We use Swimlanes for grouping related activities in one column. Swimlanes group related activities into one column or one row. Swimlanes can be vertical and horizontal. Swimlanes are used to add modularity to the activity diagram. It is not mandatory to use swimlanes. They usually give more clarity to the activity diagram. It's similar to creating a function in a program. It's not mandatory to do so, but, it is a recommended practice.



We use a rectangular column to represent a swimlane as shown in the figure above.

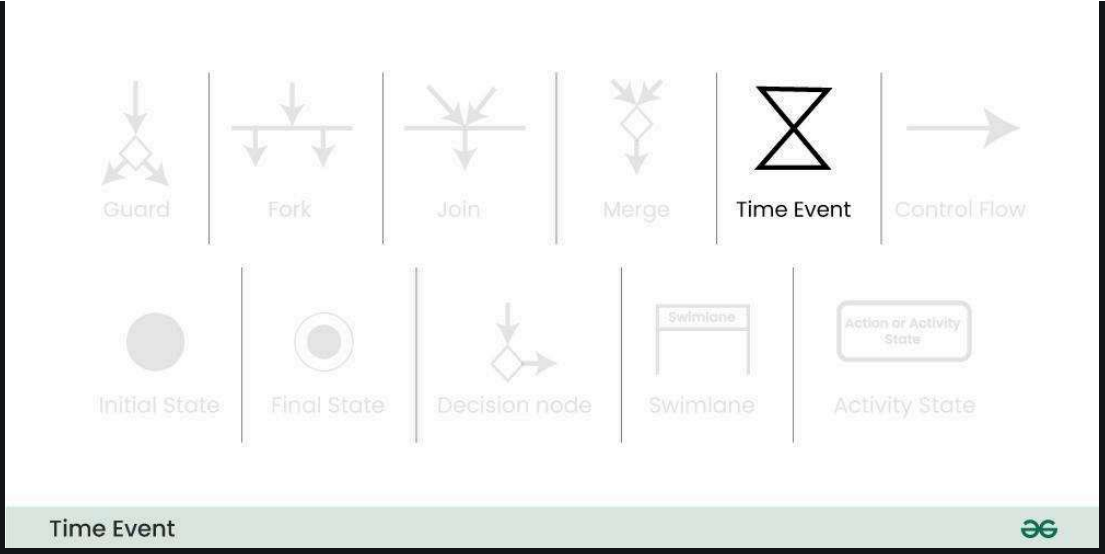
For example:

Here different set of activities are executed based on if the number is odd or even. These activities are grouped into a swimlane.



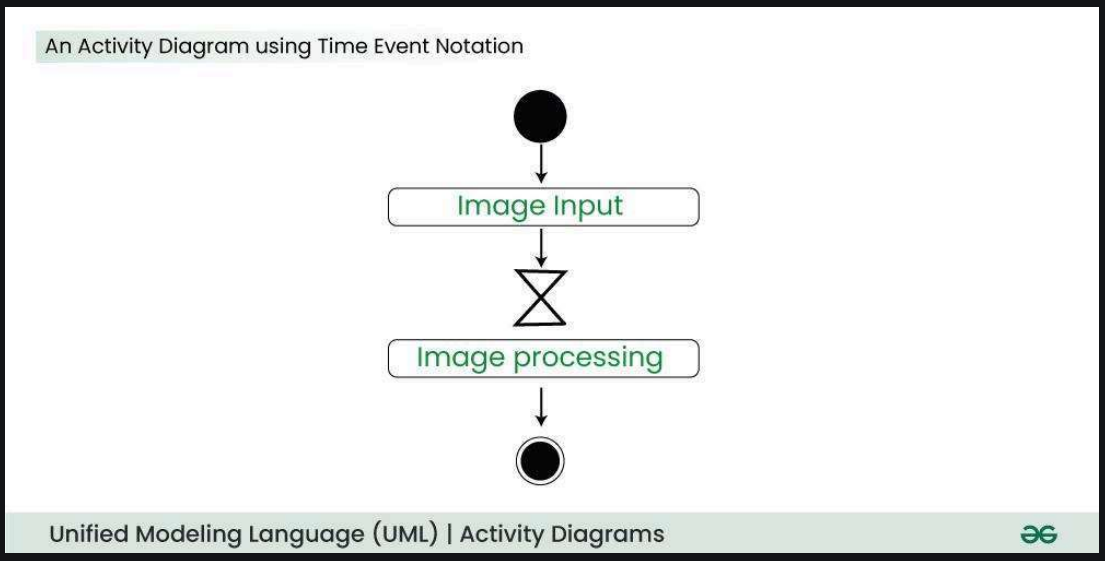
2.10. Time Event

This refers to an event that stops the flow for a time; an hourglass depicts it. We can have a scenario where an event takes some time to completed.



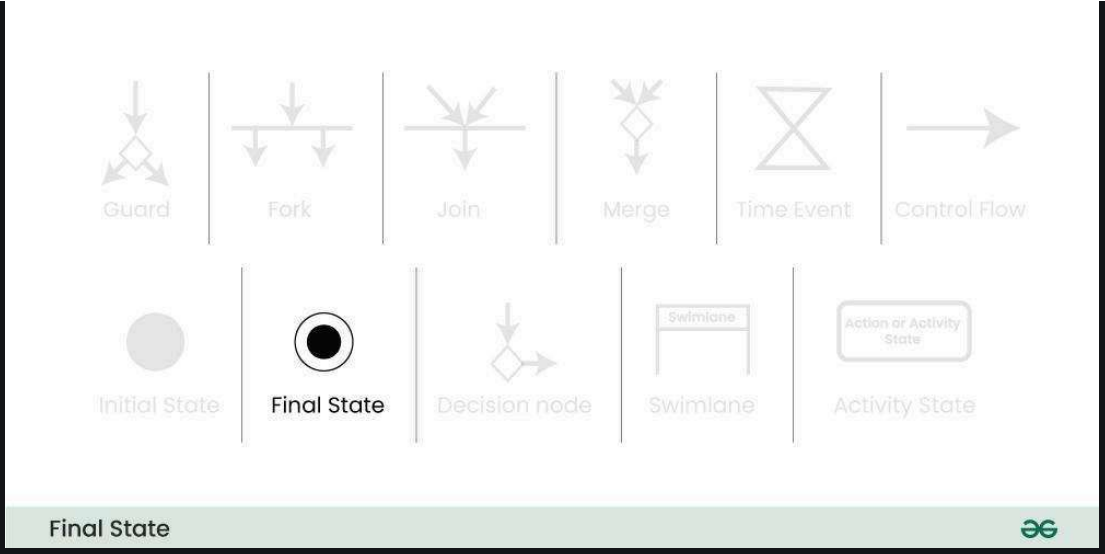
For example:

Let us assume that the processing of an image takes a lot of time. Then it can be represented as shown below.



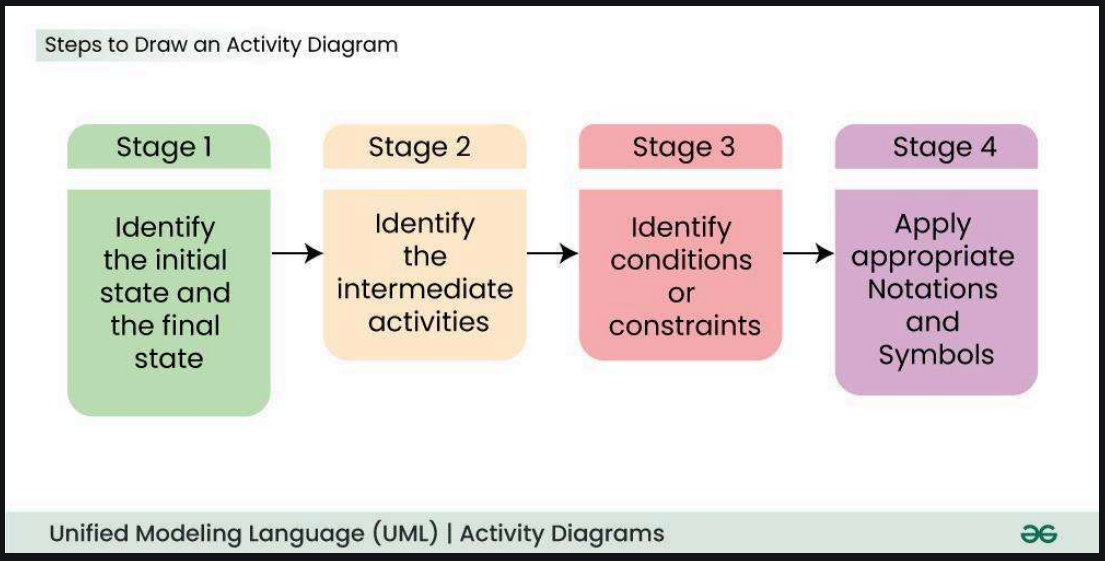
2.11. Final State or End State

The state which the system reaches when a particular process or activity ends is known as a Final State or End State. We use a filled circle within a circle notation to represent the final state in a state machine diagram. A system or a process can have multiple final states.



Final State

3. How to Draw an Activity Diagram in UML?



Unified Modeling Language (UML) | Activity Diagrams

Below are the steps of how to draw the Activity Diagram in UML:

Step 1. Identify the Initial State and Final States:

- This is like setting the starting point and ending point of a journey.
- Identify where your process begins (initial state) and where it concludes (final states).
- For example, if you are modelling a process for making a cup of tea, the initial state could be “No tea prepared,” and the final state could be “Tea ready.”

Step 2. Identify the Intermediate Activities Needed:

- Think of the steps or actions required to go from the starting point to the ending point.
- These are the activities or tasks that need to be performed.
- Continuing with the tea-making , intermediate activities could include “Boil water,” “Pour tea into a cup”.

Step 3. Identify the Conditions or Constraints:

- Consider the conditions or circumstances that might influence the flow of your process.
- These are the factors that determine when you move from one activity to another.
- Using the tea-making scenario, a condition could be “Water is boiled,” which triggers the transition to the next activity.

Step 4. Draw the Diagram with Appropriate Notations:

- Now, represent the identified states, activities, and conditions visually using the appropriate symbols and notations in an activity diagram. This diagram serves as a visual map of your process, showing the flow from one state to another.

4. What are Activity Diagrams used for?

Activity diagrams are used in software development and system design to model and visualize the dynamic aspects of a system. Here are some common uses of activity diagrams:

- Dynamic modelling of the system or a process.
- Illustrate the various steps involved in a UML use case.
- Model software elements like methods,operations and functions.
- We can use Activity diagrams to depict concurrent activities easily.
- Show the constraints, conditions and logic behind algorithms.
- During the requirements analysis phase, activity diagrams assist in capturing and documenting the dynamic aspects of user interactions.

5. What are the Differences between an Activity diagram and a Flowchart?

An activity diagram is very similar to a flowchart. So let us understand if activity diagrams or flowcharts are any different.

What is a Flow Chart?

An algorithm is like a set of clear instructions to solve a problem, and a flowchart is a picture that shows those instructions.

- When we’re writing computer programs, a flowchart helps us map out the steps of the algorithm to solve the problem.
- Non programmers use Flow charts to model workflows.
- We can call a flowchart a primitive version of an activity diagram.
- Business processes where decision making is involved is expressed using a flow chart.

Example:

A manufacturer uses a flow chart to explain and illustrate how a particular product is manufactured.

What are the differences?

Activity Diagram	Flow Chart
An activity diagram is associated with the UML(Unified Modelling Language)	A Flow Chart is associated with the programming.
An activity diagram is used to model the dynamic aspects of a system and also illustrates the workflow of activities within a use case or business process.	Depicts a diagrammatic representation illustrating a solution model to a given problem and a flow chart converges into being an activity diagram if complex decisions are being made.
Commonly used in software engineering within the UML for modeling and designing software systems on high level.	Widely used in software engineering for representing algorithms, decision structures, and program flows.

Do we need to use both the diagrams and the textual documentation?

Let's understand this with the help of an example:

- Different individuals have different preferences in which they understand something.
- To understand a concept, some people might prefer a written tutorial with images while others would prefer a video lecture.
- So we generally use both the diagram and the textual documentation to make our system description as clear as possible.

6. Conclusion

In conclusion, Activity Diagrams serve as invaluable tools in system design and analysis, offering a visual representation of dynamic processes within organizations. They are widely utilized to model business processes, illustrate user interactions, and guide software system design. By providing a clear and concise overview of activities, decision points, and interactions, activity diagrams enhance communication among project stakeholders and contribute to effective documentation.

Want to be a Software Architect or grow as a working professional? Knowing both Low and High-Level System Design is highly necessary. As such, our course fits you perfectly: Mastering System Design: From Low-Level to High-Level Solutions. Get in-depth into **System Design** with hands-on projects and **Real-World Examples**. Learn how to come up with systems that are scalable, efficient, and robust—ones that impress. Ready to elevate your tech skills? Enrol now and build the future!

Previous Article

State Machine Diagrams | Unified Modeling Language (UML)

Next Article

Use Case Diagrams | Unified Modeling Language (UML)

Similar Reads

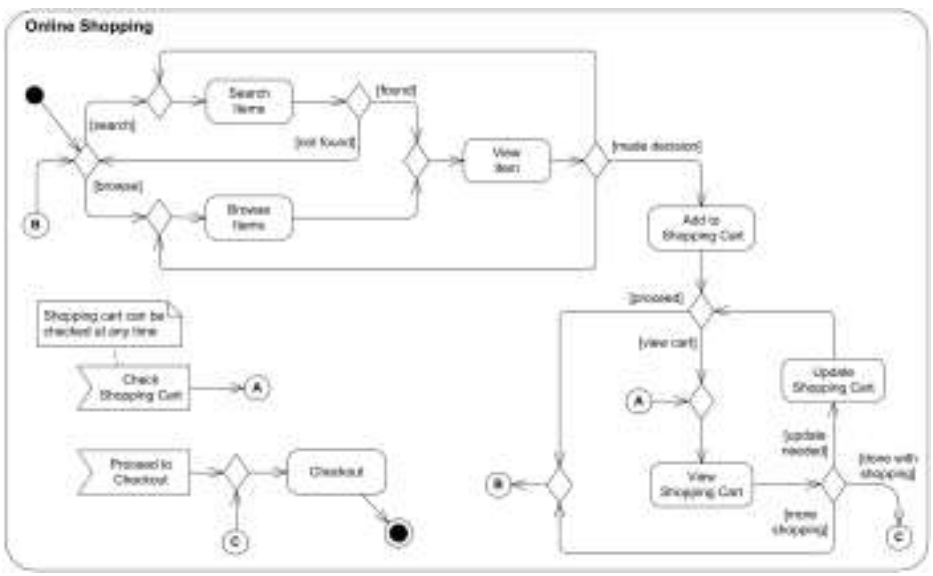
UML Activity Diagram Examples

Here we provide several examples of UML **activity diagrams**:

Online shopping UML activity diagram

***Purpose**: An example of activity diagram for online shopping.*

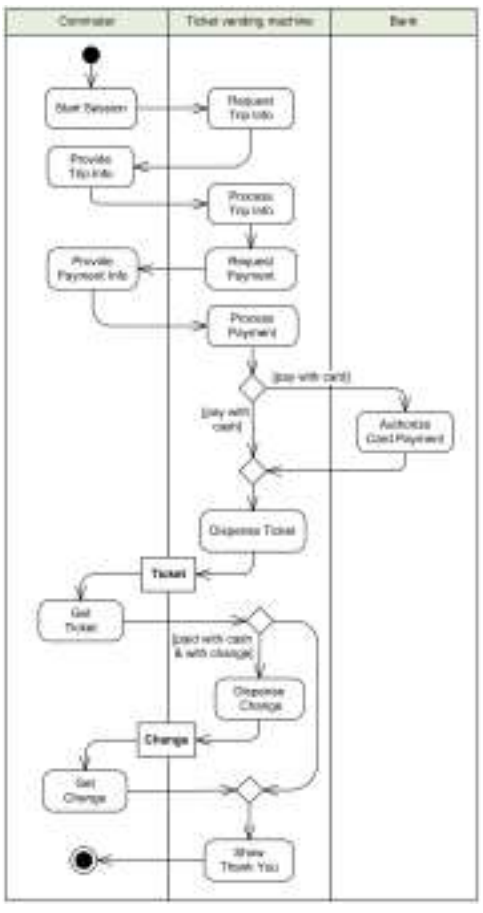
***Summary**: Online customer can browse or search items, view specific item, add it to shopping cart, view and update shopping cart, do checkout. User can view shopping cart at any time.*



Ticket vending machine

***Purpose**: An example of UML activity diagram describing behavior of the **Purchase Ticket** **use case** for a Ticket vending machine.*

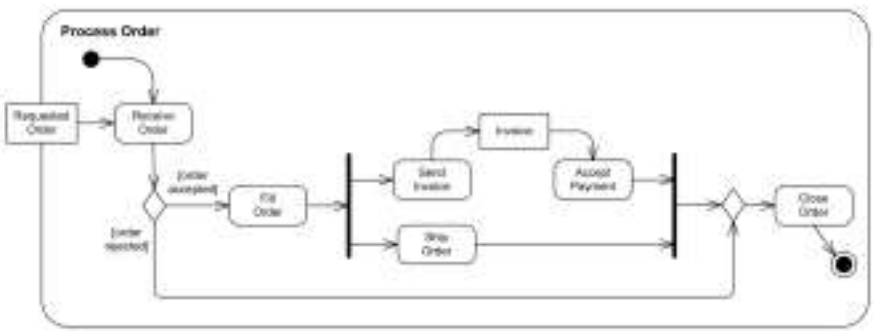
***Summary**: Activity is started by Commuter actor who needs to buy a ticket. Ticket vending machine will request trip information from Commuter. Based on the info machine will calculate payment due and request payment options. After payment is complete, ticket is dispensed to the Commuter.*



Business flow - Process order

Purpose: *An example of business flow UML activity diagram to process purchase order.*

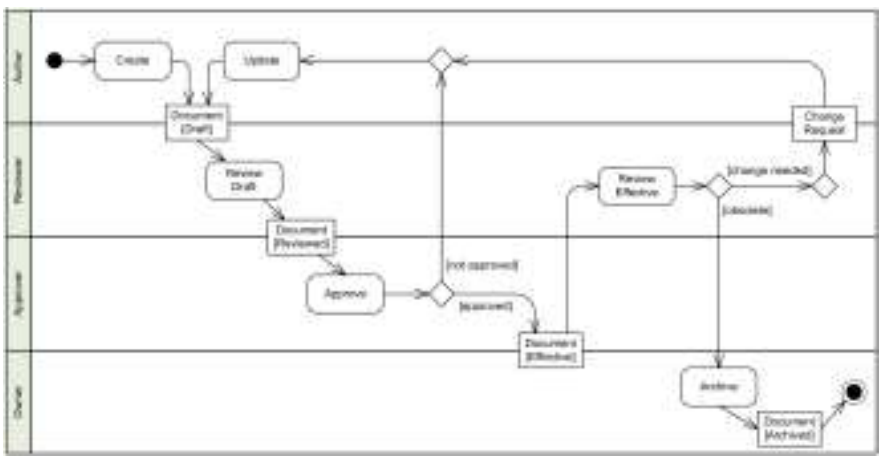
Summary: *Requested order is input parameter of the activity. After order is accepted and all required information is filled in, payment is accepted and order is shipped.*



Business flow - Document management process

Purpose: *An example of UML activity diagram describing a Document Management Process. Some kind of formal and properly communicated document management process is usually required in any major corporation especially under a regulatory compliance.*

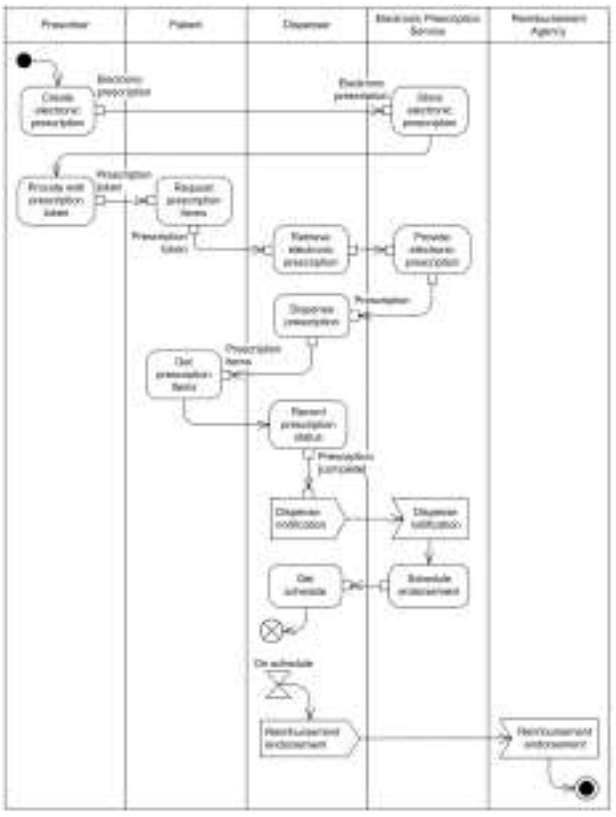
Summary: *Document is created, reviewed, updated, approved, and at some point archived. This activity diagram example shows responsibilities of different roles and a flow of document changes. Partitions represent different roles participating in the activity - Author, Reviewer, Approver, and Owner.*



Electronic prescription service

Purpose: *Electronic prescriptions UML activity diagram example is based on documentation for the Electronic Prescription Service (EPS) R2 developed by the NHS Connecting for Health (NHS CFH) in England.*

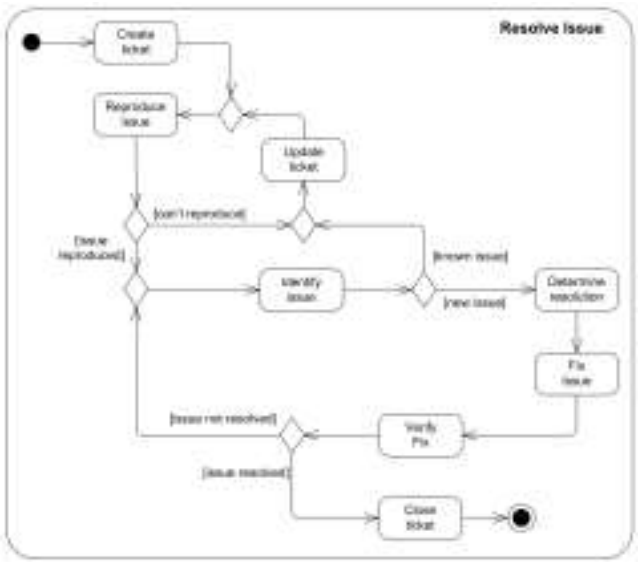
Summary: *Prescribers could send prescriptions electronically to a pharmacy of the patient's choice where patient can pick it up. Dispenser retrieves electronic prescriptions from the EPS.*



Software design - Resolve issue

Purpose: *An example of UML activity diagram to resolve an issue in software design.*

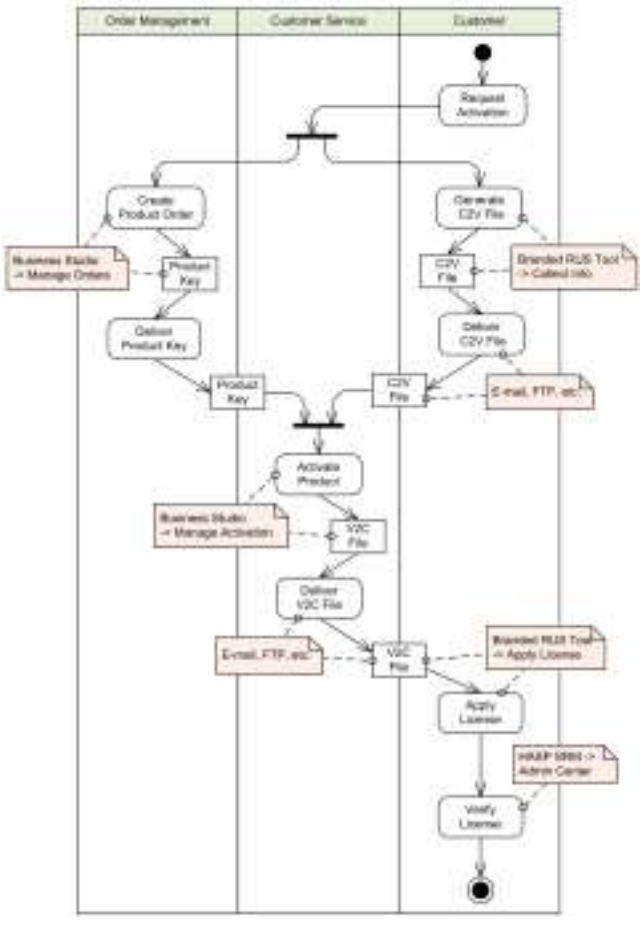
Summary: *After issue or error ticket is created by some authority and the issue was reproduced, issue is identified, resolution is determined, issue is fixed and verified, and ticket is closed, if issue was resolved.*



Activation of Sentinel HASP SL provisional product UML activity diagram example

Purpose: *An example of activity diagram describing manual activation of trial (provisional) product which was protected by Sentinel HASP SL software key of the Sentinel HASP - software protection and licensing security solution.*

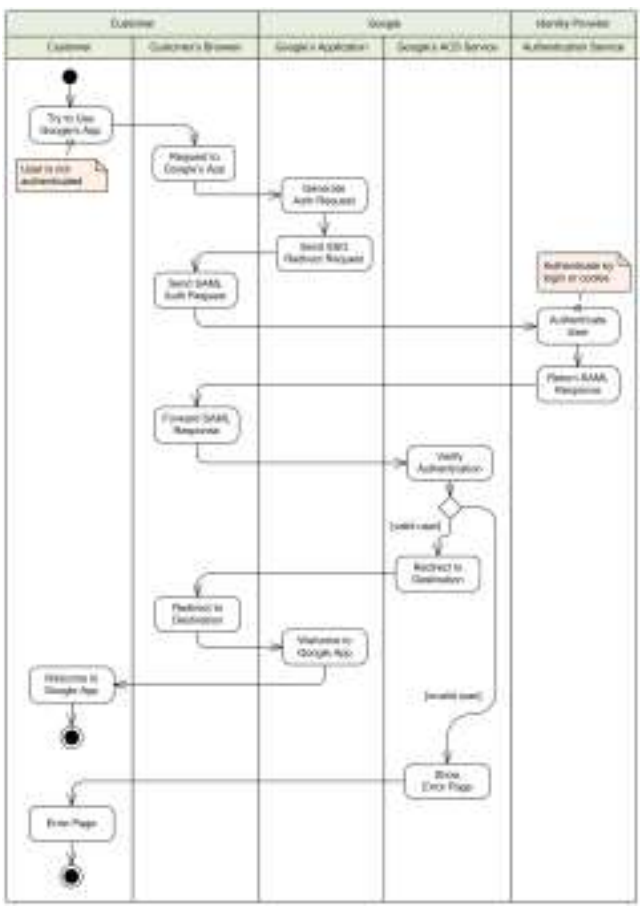
Summary: *Customer has some trial product installed, for example, some game or tool, which has specific trial period and could have some limited features or options. After using the product for some time customer decides to activate product by requesting a permanent, full product license. Order Manager creates a new activation key for the product.*



A Single Sign-On (SSO) for Google Apps

Purpose: *An example of UML activity diagram which describes Single Sign-On (SSO) to Google Apps for customers using some hosted Google application, such as Gmail.*

Summary: *When a user attempts to use some hosted Google application, such as Gmail, Google generates a SAML authentication request and sends redirect request back to the user's browser. Redirect points to the specific identity provider. SAML authentication request contains the encoded URL of the Google application that the user is trying to reach.*

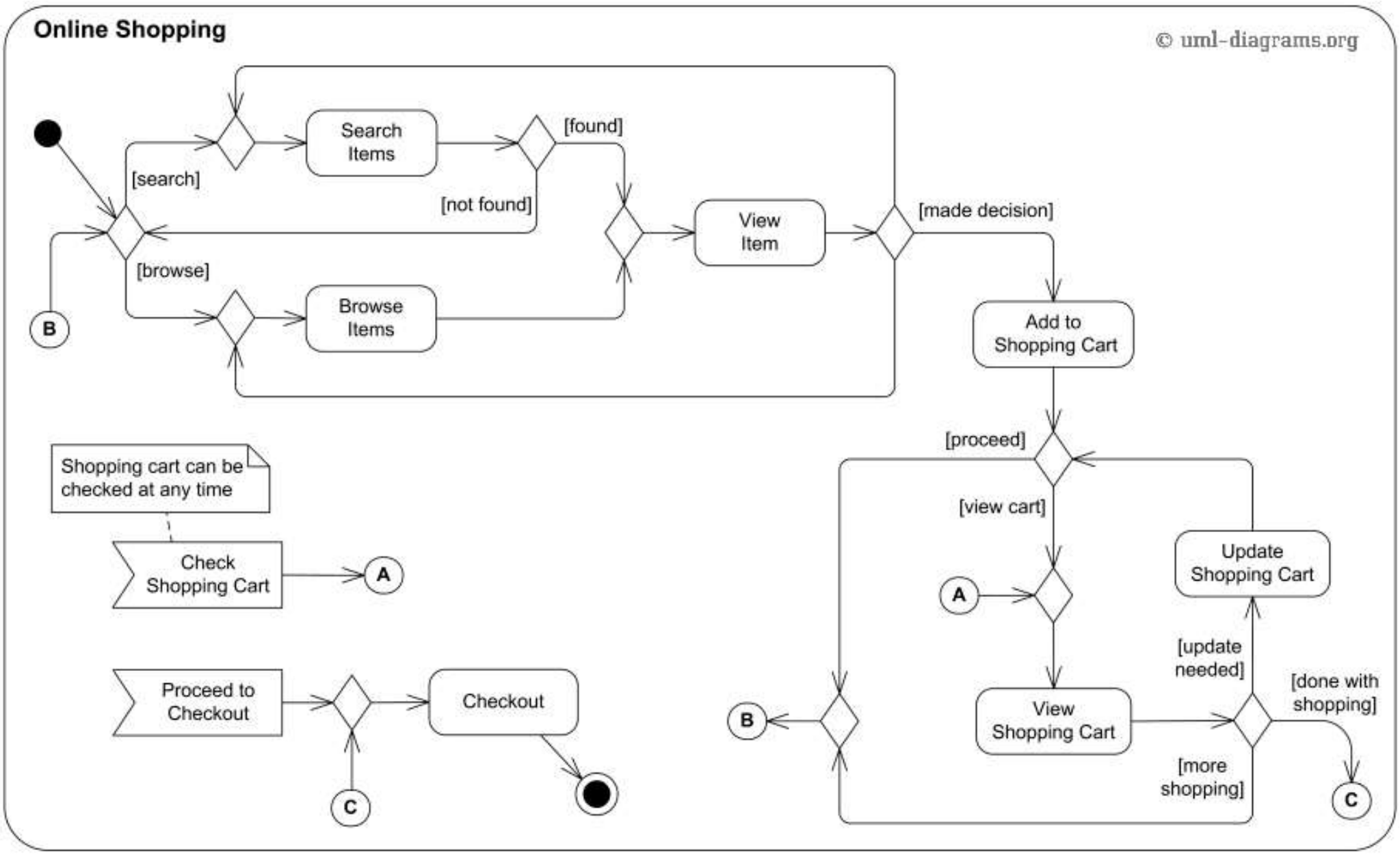


Online Shopping

UML Activity Diagram Example

An example of **activity diagram** for **online shopping**. Online customer can browse or search items, view specific item, add it to shopping cart, view and update shopping cart, checkout. User can view shopping cart at any time. Checkout is assumed to include user registration and login.

This example does not use partitions, most of the actions are assumed to be fulfilled by online customer.



An example of UML activity diagram for online shopping.

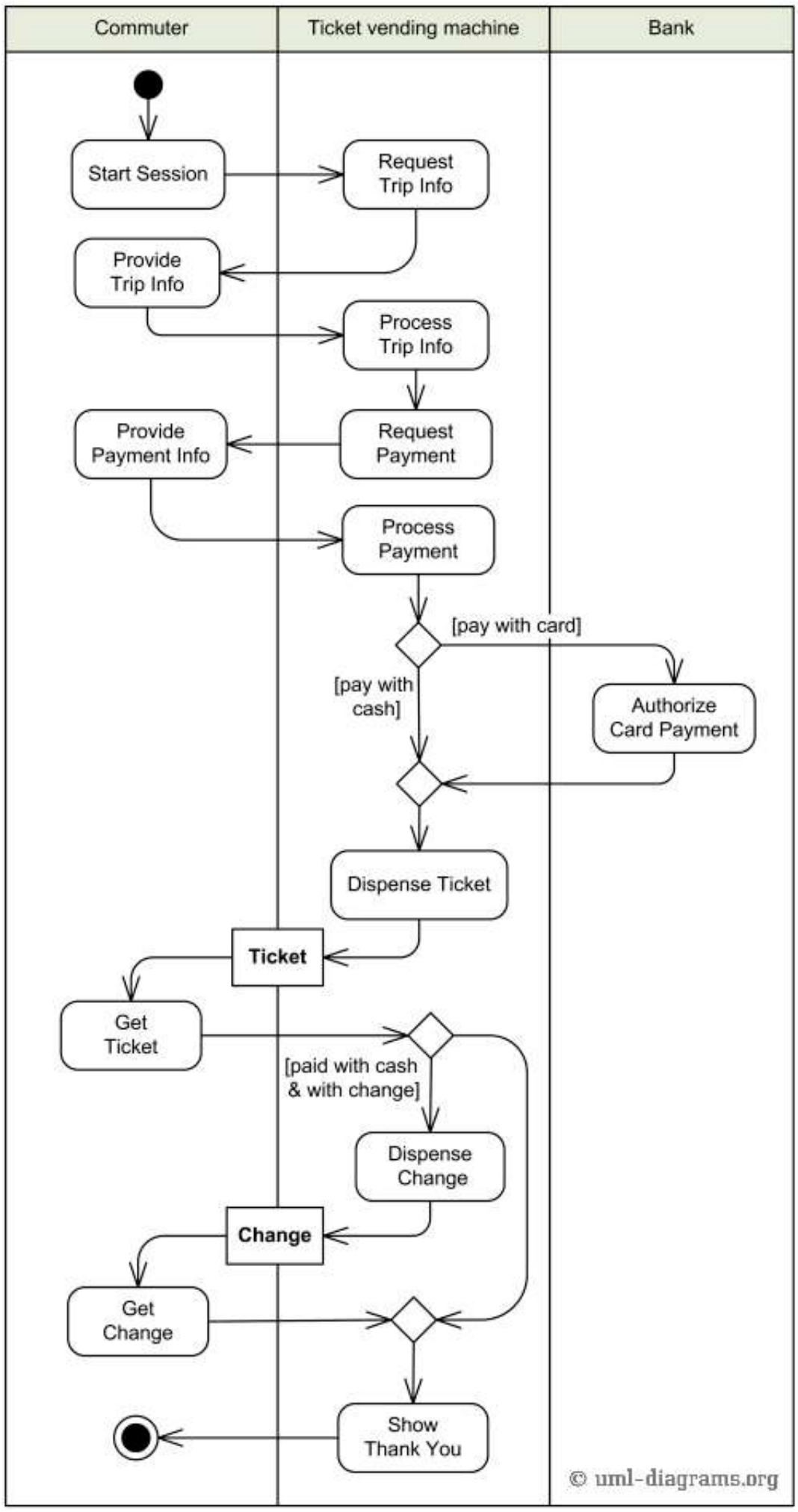
Ticket Vending Machine

UML Activity Diagram Example

This is an example of UML **activity diagram** describing behavior of the **Purchase Ticket use case**.

Activity is started by Commuter **actor** who needs to buy a ticket. Ticket vending machine will request trip information from Commuter. This information will include number and type of tickets, e.g. whether it is a monthly pass, one way or round ticket, route number, destination or zone number, etc.

Based on the provided trip info ticket vending machine will calculate payment due and request payment options. Those options include payment by cash, or by credit or debit card. If payment by card was selected by Commuter, another actor, Bank will participate in the activity by authorizing the payment.



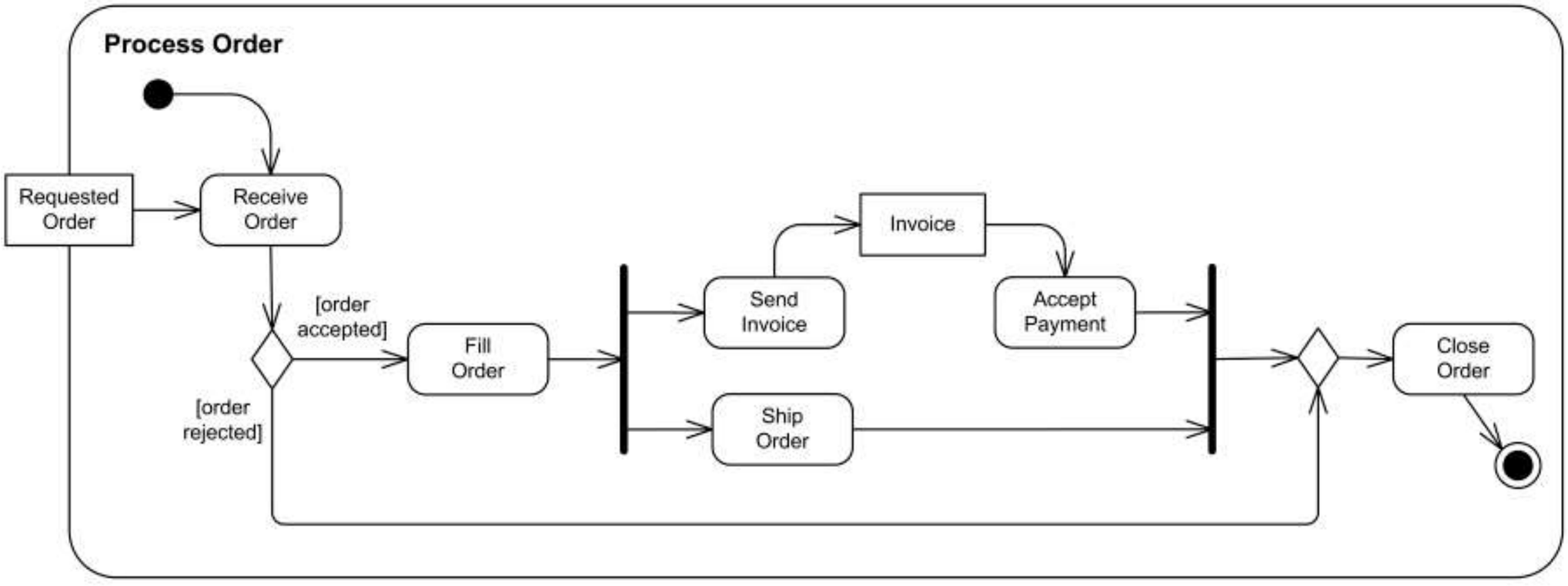
Example of Purchase Ticket use case behavior described using UML activity diagram.

After payment is complete, ticket is dispensed to the Commuter. Cash payment might result in some change due, so the change is dispensed to the Commuter in this case. Ticket vending machine will show some "Thank You" screen at the end of the activity.

Process Shopping Order

UML Activity Diagram Example

An example of business flow **activity** of order processing, based on the Example 12.35 from ***[UML 2.4.1 Specification]***. Requested order is input parameter of the activity. After order is accepted and all required information is filled in, payment is accepted and order is shipped. Note, that this business flow allows order shipment before invoice is sent or payment is confirmed.



An example of business flow activity to process purchase order.

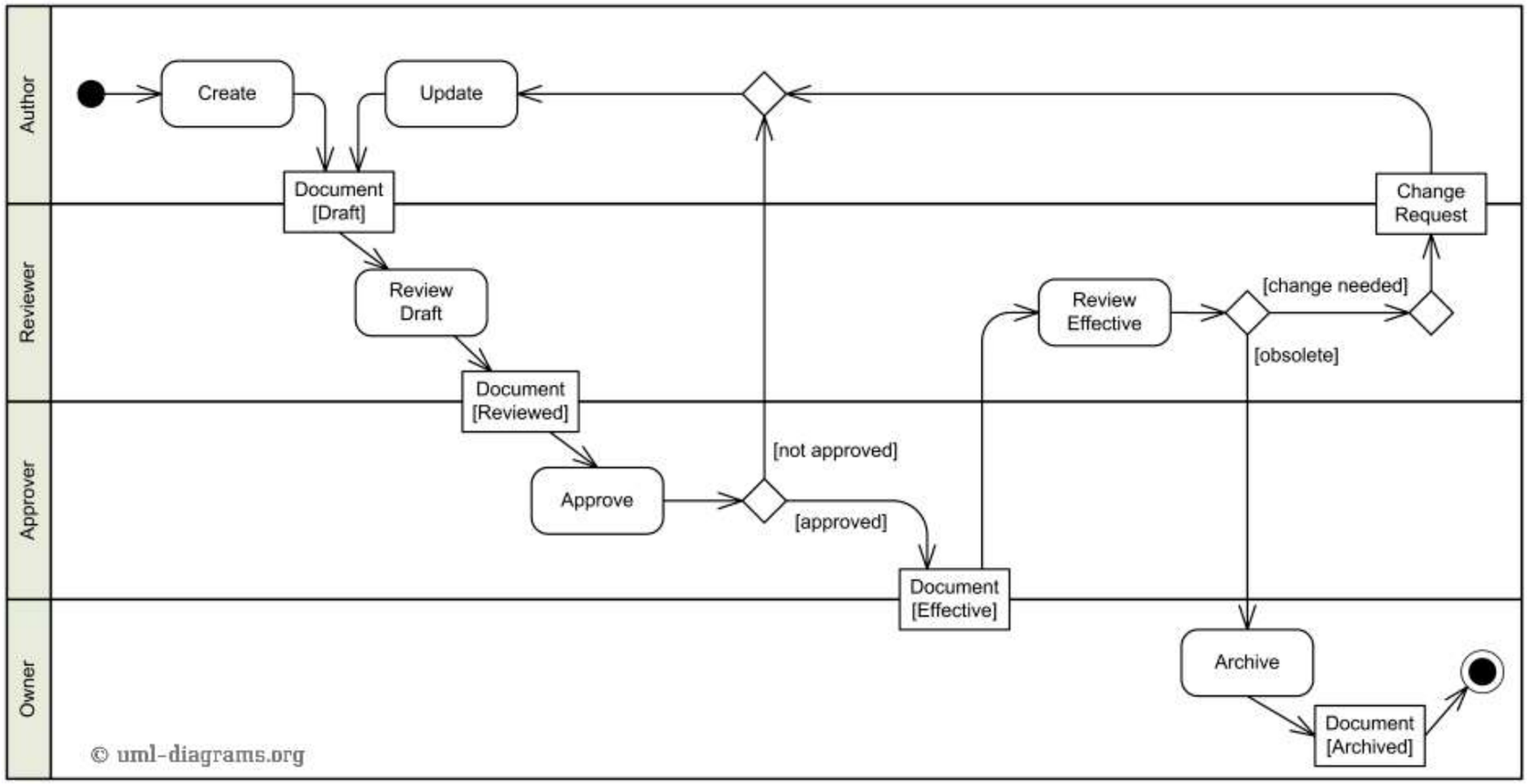
This example does not use partitions, so it is not clear who is responsible for fulfilling each specific action.

Document Management Process

UML Activity Diagram Example

An example of UML activity diagram describing a **Document Management Process**. Some kind of formal and properly communicated document management process is usually required in any major corporation especially under a regulatory compliance.

A **document** goes through different **state** or stages - it is created, reviewed, updated, approved, and at some point archived. Different roles participating in this process are **Author**, **Reviewer**, **Approver**, and **Owner**. These roles are represented on the diagram by **partitions** rendered as horizontal "**swimlanes**".



An example of Document Management Process activity.

This activity diagram shows responsibilities of different roles and flow or sequence of document changes. Alternative type of diagram - **state machine diagram** - could also be used in this case to show how document changes its state over time.

Note, that the **Document** object is not the only object node shown on this activity diagram. There is also another object - **Change Request**, an object which is used to pass changes to the document requested by **Reviewer**. State diagram for the Document will only show the document states and transitions, so activity diagram is useful when different roles and several object nodes are involved.

Electronic Prescriptions

UML Activity Diagram Example

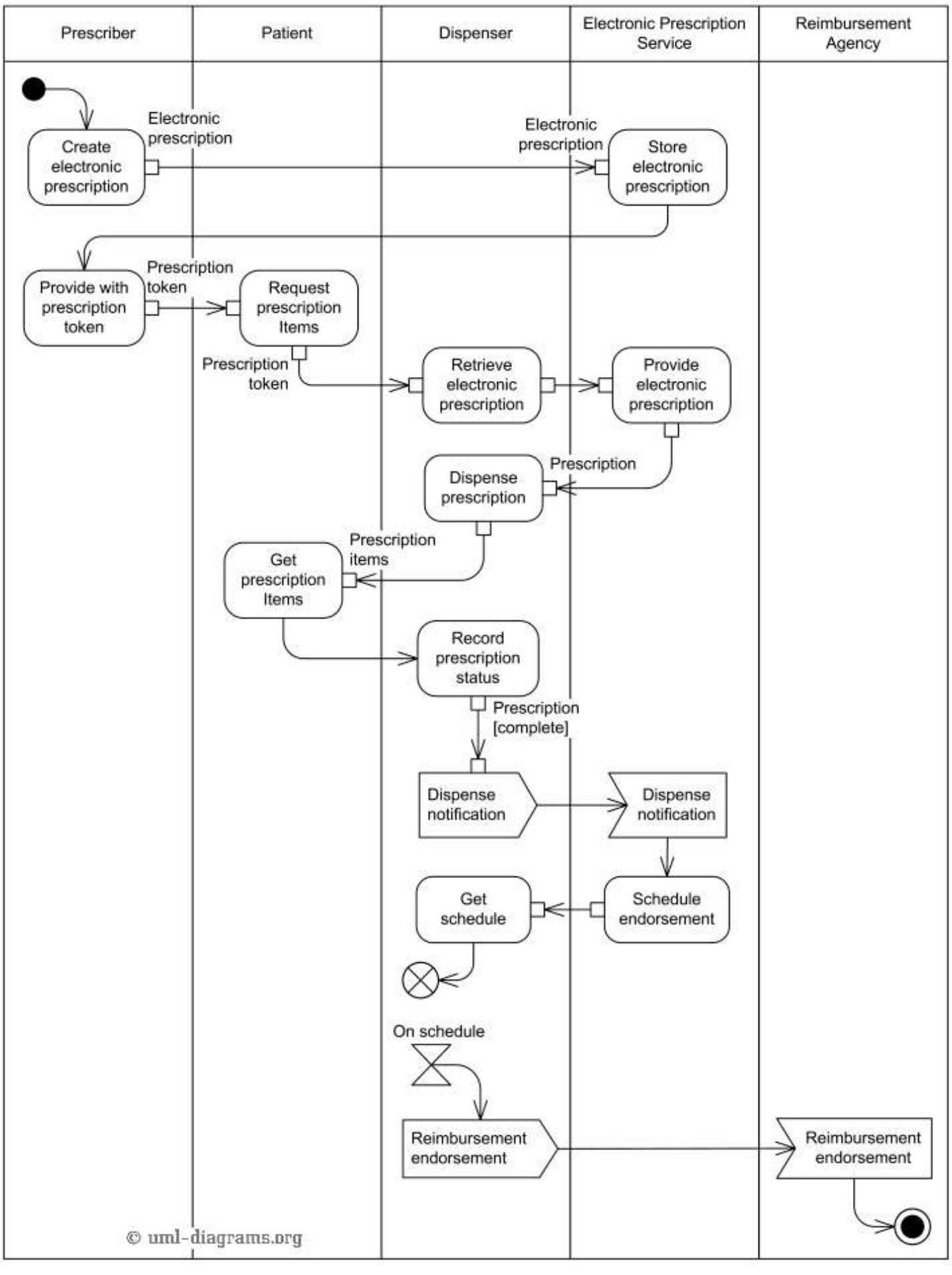
This is an example of **activity diagram** for **electronic prescriptions**. This example is based on documentation for the impressive ***Electronic Prescription Service*** (EPS) Release 2 developed by the ***NHS Connecting for Health*** (NHS CFH) as a part of the National Programme for IT in the National Health Service (NHS) in England. Note, that this page represents informal UML diagram example. Please refer to the NHS CFH to get a range of official communications and guidance materials.

The **Electronic Prescription Service** enables **prescribers** - such as general practitioners (GPs) and practice nurses - to send prescriptions electronically to a **dispenser** (such as a pharmacy) of the patient's choice.

As with all NHS Connecting for Health services, access to the EPS is controlled through the use of the NHS **Smartcard** having user's name, photograph and unique identity number printed on, and with embedded smart chip. The Smartcard gives individual users different levels of access depending on their role.

Prescriber logs onto the clinical system using their Smartcard and passcode, chooses medication or medical appliance for the patient, adds prescribing endorsements where required, and applies electronic signature to authorize the electronic prescription. Electronic prescription is transmitted to the EPS. Prescription token is printed where required. Authorized person hands prescription token to patient where necessary.

Paper copies of electronic prescriptions are called **tokens**. They act as a hard copy of the details contained within the electronic prescription. There are two types of tokens - 'prescription tokens' and 'dispensing tokens'. When the EPS is fully introduced, paper copies of electronic prescriptions will no longer be needed but should still be available by request, whenever necessary.



An example of UML activity diagram for electronic prescriptions.

A **dispenser** (or dispensing contractor) is any organization that dispenses NHS primary care prescriptions to patients, such as a community pharmacy, a dispensing appliance contractor or a dispensing GP practice. With EPS, only prescriptions sent to a patient's nominated dispensing contractor can be signed and sent electronically.

Dispenser retrieves electronic prescriptions from the EPS. It could be done in three ways either by automatic download (e.g. overnight), by manually entering the prescription ID printed on the token, or by scanning a barcode on a prescription token. Dispensing token is printed if required. Prescription items are issued to the patient or patient's representative.

Dispenser should record the status of each of the prescription items as one of the 'dispensed', 'not dispensed', 'owing' or 'partial'. In order to complete the dispensing process the whole prescription has to be completed, meaning that all prescribed items must be marked as either 'dispensed' or 'not dispensed'. Some clinical systems will automatically record the status of dispensed items.

If dispensing process is complete, dispenser should send **dispense notification** to the Electronic Prescription Service. The message informs the EPS which medication has/has not been supplied to the patient. A schedule will be issued for pharmacists to follow as to when to submit the electronic **reimbursement endorsement** message. The electronic reimbursement endorsement message can only be sent once the dispense notification message has been sent for the electronic prescription.

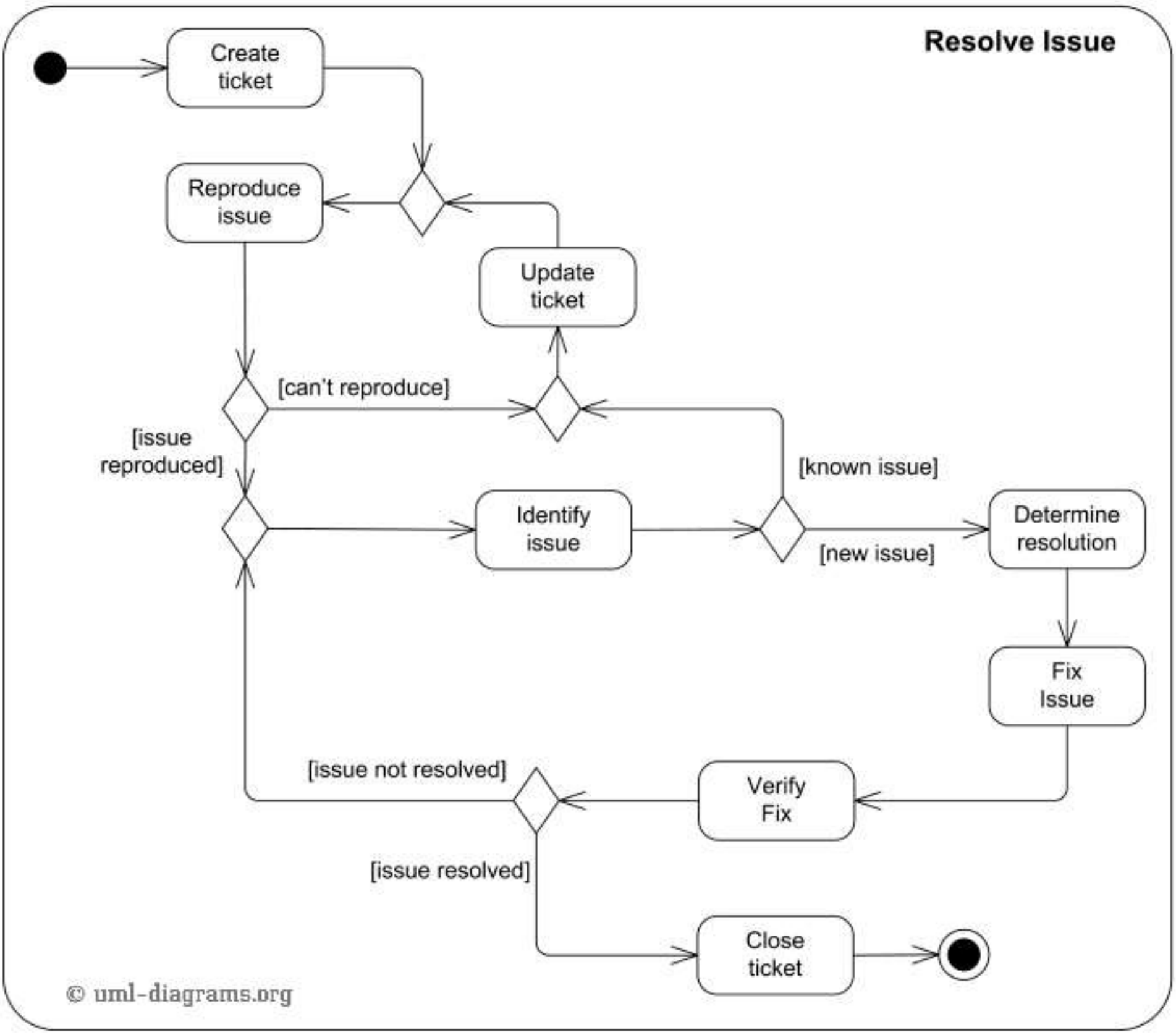
To support the reimbursement claim process, the EPS will allow dispensers to electronically submit **reimbursement endorsement** messages to the **reimbursement agency** for the dispensed electronic prescriptions so that the reimbursement agency can make a payment. The messages are sent according to the reimbursement agency scheduling.

Resolve Software Issue

UML Activity Diagram Example

An example of **UML activity diagram** which shows how to resolve an issue in a software design. After ticket is created by some authority and the issue is reproduced, issue is identified, resolution is determined, issue is fixed and verified, and ticket is closed, if issue was resolved.

This example does not use partitions, so it is not very clear who is responsible for fulfilling each specific action.



An example of UML activity diagram to resolve an issue in software design.

Sentinel HASP SL - Activation of Trial Product

UML Activity Diagram Example

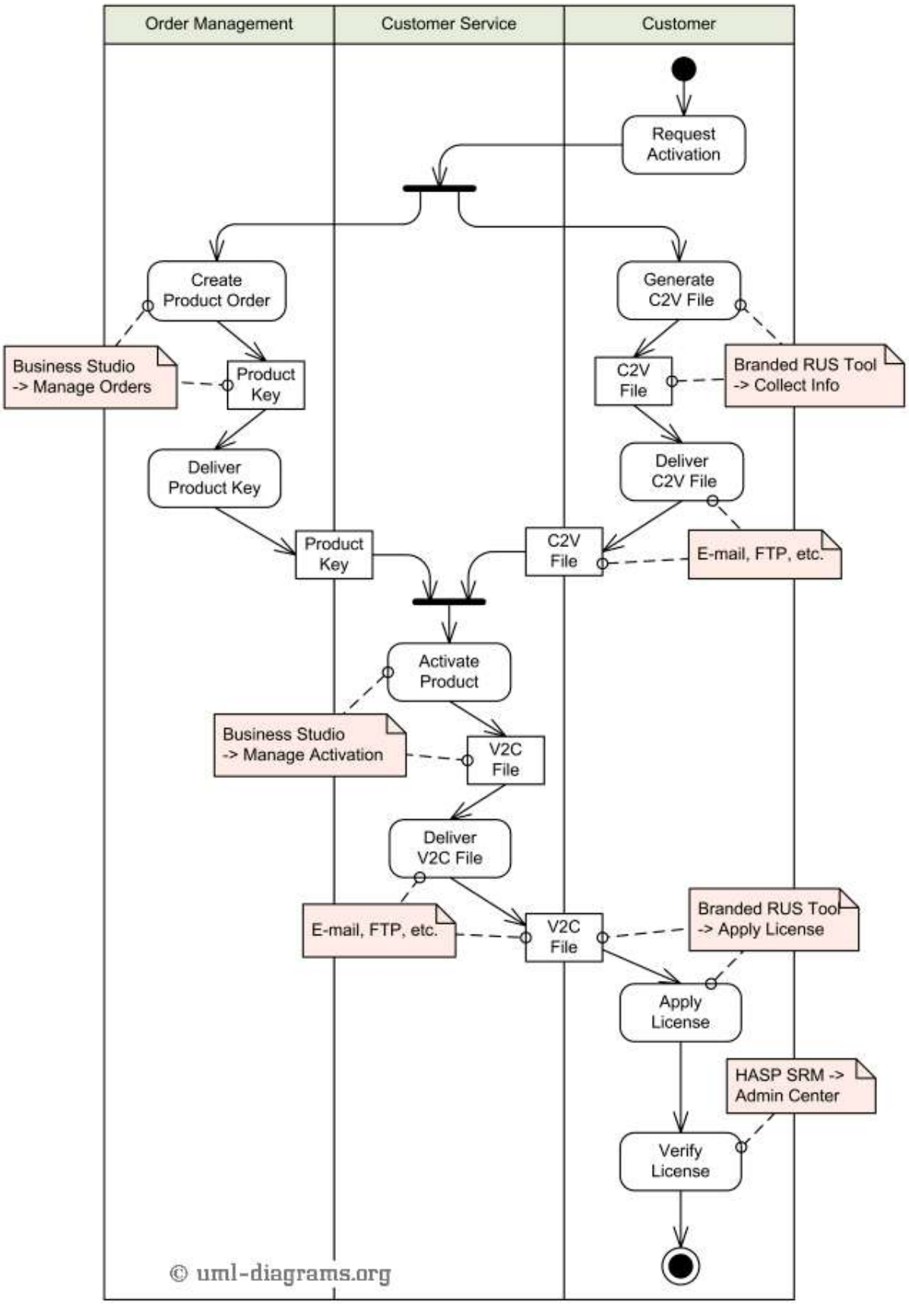
An example of **activity diagram** describing manual activation of trial (provisional) product which was protected by **Sentinel HASP SL** software key of the Sentinel HASP - software protection and licensing security solution.

Sentinel HASP protects against losses from software piracy and intellectual property theft. For example, it offers industry-leading support for licensing in virtual environments, and is the first software licensing and reverse engineering protection tool solution on the market to support J2EE applications.

Sentinel HASP software includes **Business Studio** - a powerful, role-based software licensing and management tool. The Business Studio is used by product, marketing and development staff to prepare software product for market and includes all of the tools necessary to license and lock application to a Sentinel HASP HL hardware or HASP SL software product key, to manage and track licenses, to create product keys that are later used for the product activation process, etc.

Three **activity partitions** are shown on the diagram as vertical **swimlanes** and represent **actors** participating in the activity - **Order Management**, **Customer Service**, and **Customer**.

Customer has some trial product installed, for example, some game or tool, which has specific trial period and could have some limited features or options. After using the product for some time customer decides to activate product by requesting a permanent, full product license.



An example of activity for manual activation of trial product protected by HASP SL.

Order Manager will have to create a new activation key for the product while the customer could create and deliver C2V file ("computer fingerprint"). Once both the new product key and the C2V file become available to the customer service, it could activate product, generate V2C file and deliver it back to the customer.

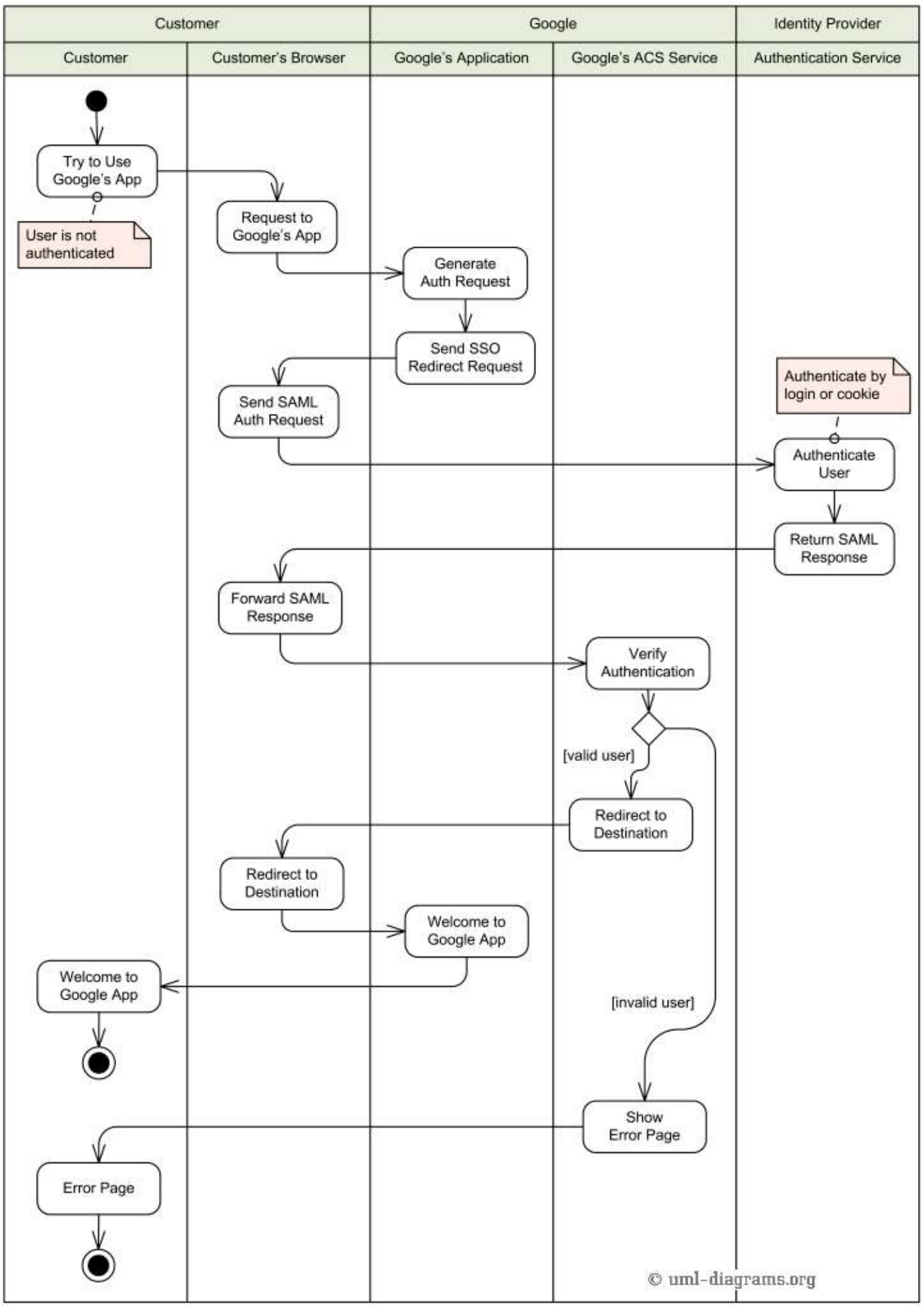
The customer applies license received and thus activates installed trial product to become a full product. The product could be protected from copying to other computers or virtual machines by the HASP SL license and protection key.

Single Sign-On for Google Apps

UML Activity Diagram Example

An example of **UML activity diagram** describing **Single Sign-On** (SSO) to Google Apps.

To interact with partner companies Google uses single sign-on based on OASIS **SAML 2.0** protocol. Google acts as service provider with services such as Gmail or Start Pages. Partner companies act as identity providers and control user names, passwords, and other information used to identify, authenticate and authorize users for web applications that Google hosts. Each partner provides Google with the URL of its SSO service as well as the public key that Google will use to verify SAML responses.



An example of UML activity diagram for Single Sign-On to Google Apps.

When a user attempts to use some hosted Google application, such as Gmail, Google generates a SAML authentication request and sends redirect request back to the user's browser. Redirect points to the specific identity provider. SAML authentication request contains the encoded URL of the Google application that the user is trying to reach.

The partner identity provider authenticates the user by either asking for valid login credentials or by checking for its own valid authentication cookies. The partner generates a SAML response and digitally signs it. The response is forwarded to **Google's Assertion Consumer Service** (ACS).

Google's ACS verifies the SAML response using the partner's public key. If the response is valid and user identity was confirmed by identity provider, ACS redirects the user to the destination URL. Otherwise user will see error message.