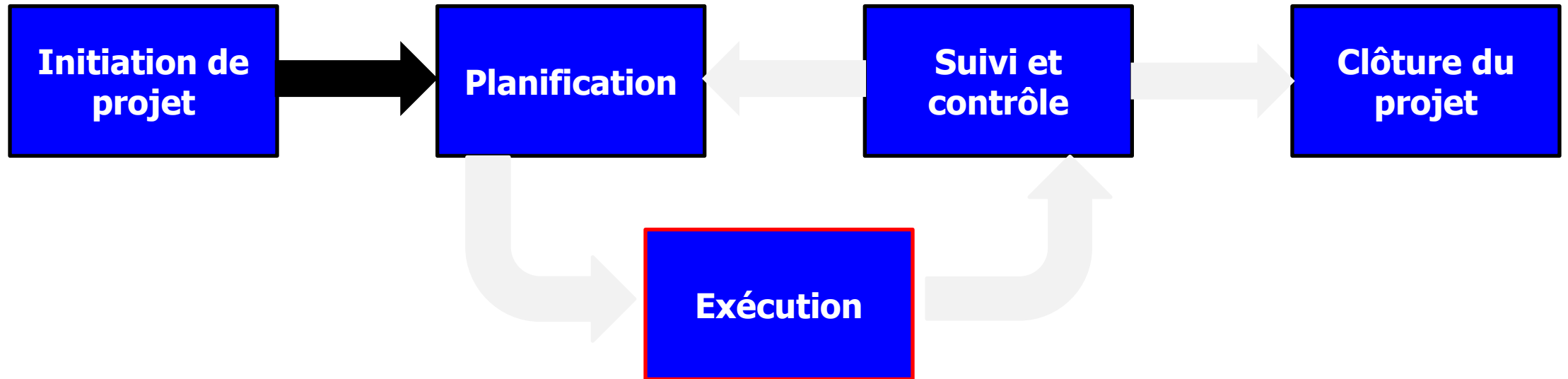


Chapitre 6

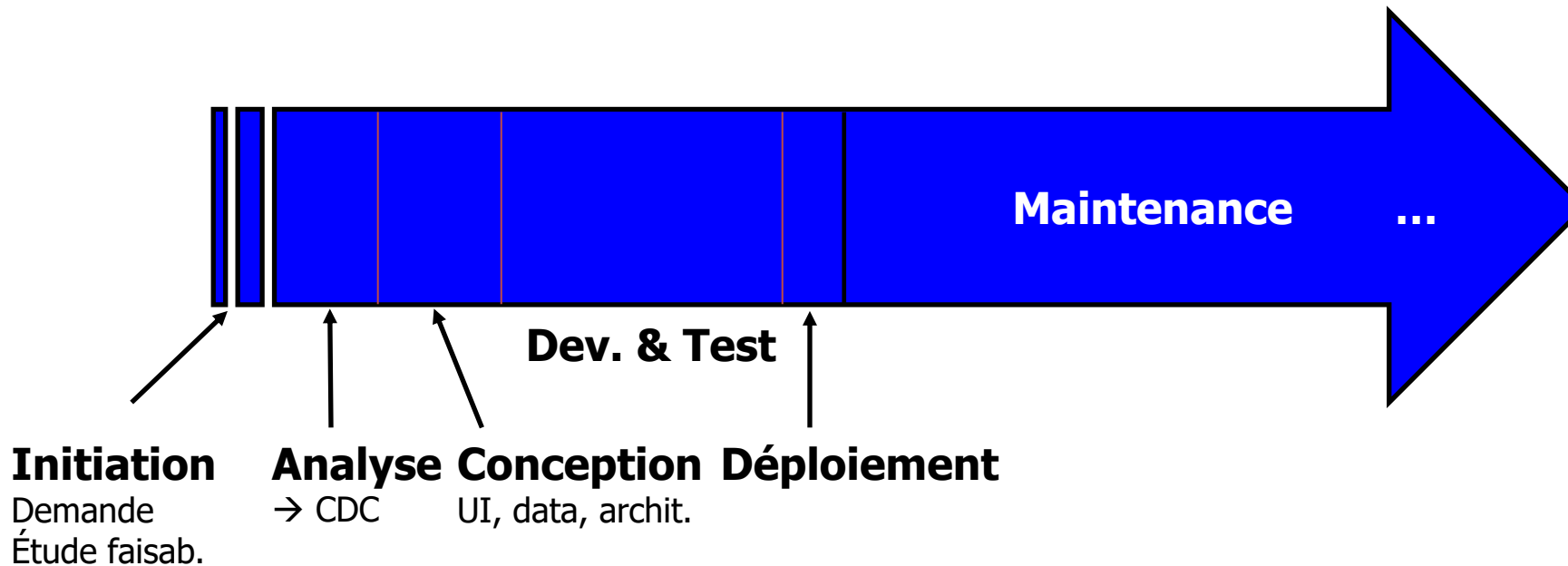
UML : Modélisation Dynamique

Vue d'ensemble



RAPPEL

Cycle de vie d'un logiciel ou système:



Objectifs du chapitre

Pouvoir expliquer l'utilité d'un diagramme d'Etats / Transition

Connaitre les éléments d'UML pour ce type de diagramme

Modéliser les états d'un objet en utilisant le diagramme UML Etats/Transition

Introduction

Décrit le comportement des objets d'une classe au moyen d'un automate d'états associés à la classe

Le comportement est modélisé par un graphe :

Nœuds = états possibles des objets

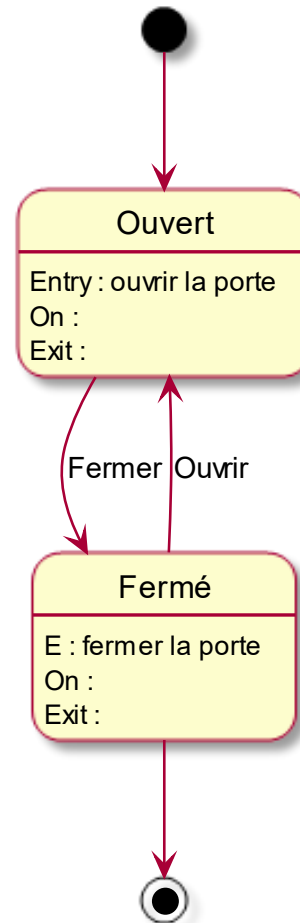
Arcs = transitions d'état à état

Une transition :

exécution d'une action

réaction de l'objet suite à un événement

Exemple



Principe

Description du cycle de vie d'un objet d'une classe

États possibles pour les objets d'une classe

Événements provoquant la transition d'un état à un autre

Actions subies ou provoquées qui accompagnent un changement d'état

Activités qui surviennent tant que l'objet est dans un état donné.

Objectif global : étudier les états d'un Système

Comprendre le système en s'intéressant aux classes qui présentent des traitements complexes

On se limite aux classes qui sont cruciales pour le champ de l'étude

On se limite aux états qui sont d'un intérêt pour le champ de l'étude

Fournir une représentation dynamique du comportement des objets d'une classe

Aider à déterminer les événements qui occasionnent les transitions

Aider à déterminer les opérations qui vont permettre ces transitions

Notion d'Etat

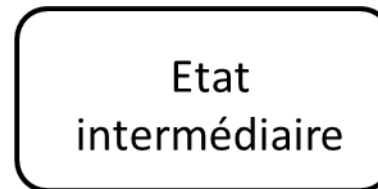
État : étape dans le cycle de vie d'un objet

Chaque objet possède à un instant donné un état particulier

Chaque état est identifié par un nom.

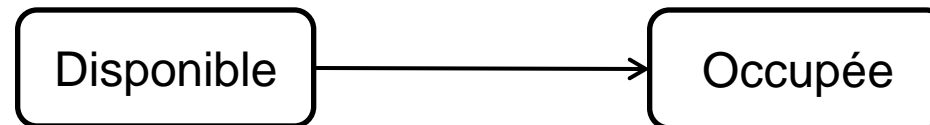
Un état est **stable** et **durable**

Chaque diagramme d'états-transitions comprend un état initial



Notion de Transition

Transition : Connexion **unidirectionnelle** qui relie deux états



Exemple: Objet « place de parking »

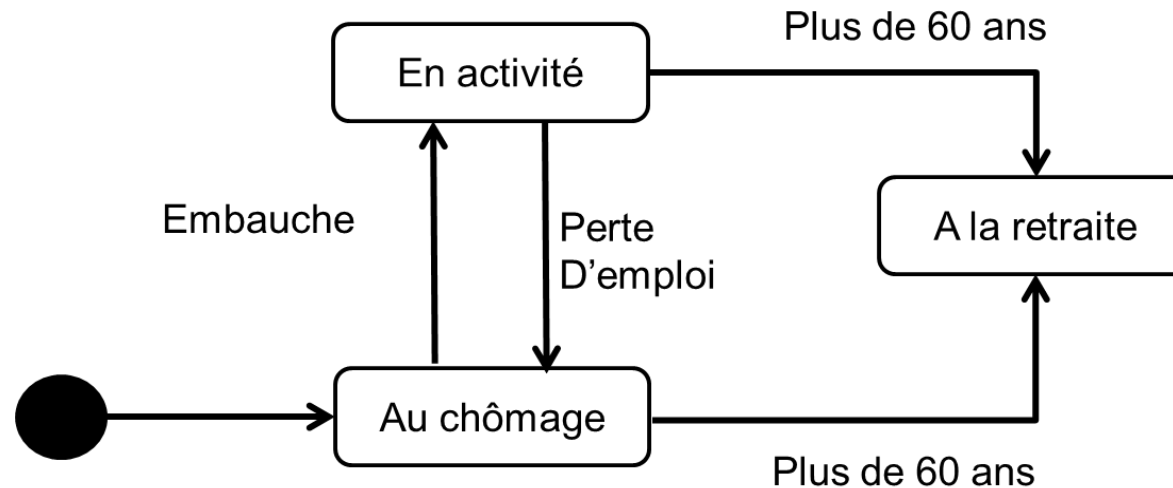
Notion d'évènement

correspond à l'occurrence d'une situation donnée

Information instantanée qui doit être traitée quand elle apparaît



Exemple:



Résumé

Etat d'un objet

Situation d'un objet que l'on désire connaître et gérer.

Transition

Passage d'un objet d'un état à un autre. Elle est déclenchée par un événement

Événement

Stimulus qui provoque une ou plusieurs transitions. À chaque stimulus peut correspondre une action responsable des modifications de l'objet (les valeurs des attributs)

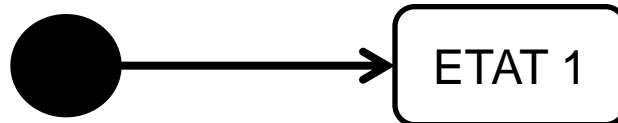
Etats spéciaux

2 états sont prédéfinis

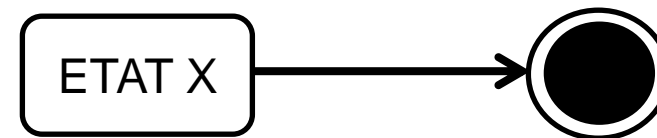
L'état de démarrage : obligatoire, unique

L'état de fin : optionnel, peut-être multiple

Création de l'objet

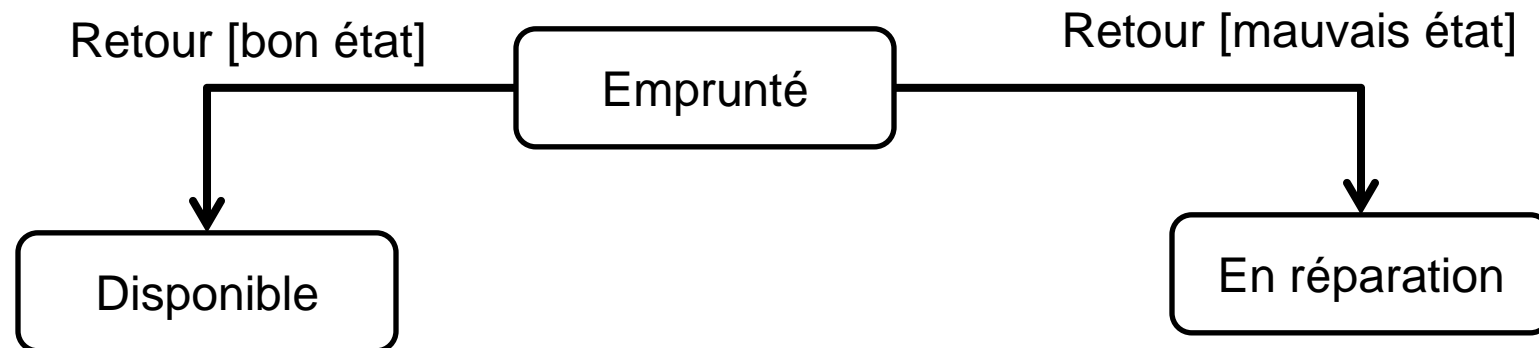


Fin de vie de l'objet



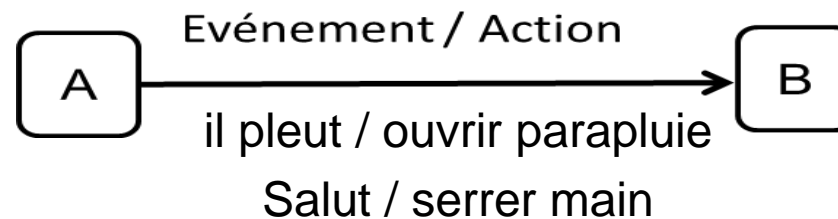
Notion de garde

Une garde est une **condition booléenne** qui permet ou non le déclenchement d'une transition lors de l'occurrence d'un événement



Notions d'opération et d'action

- Action et activités : le lien entre les **opérations** définies dans la spécification d'une **classe** et les événements apparaissent dans le diagramme d'états-transitions
- Chaque transition peut avoir une action à exécuter lorsqu'elle est déclenchée. L'action est considérée comme instantanée et atomique
- Action signifie l'exécution d'une des opérations déclarées dans la classe de l'objet destinataire de l'événement
- L'action a accès aux paramètres de l'événement ainsi qu'aux attributs de l'objet sur lequel elle s'applique

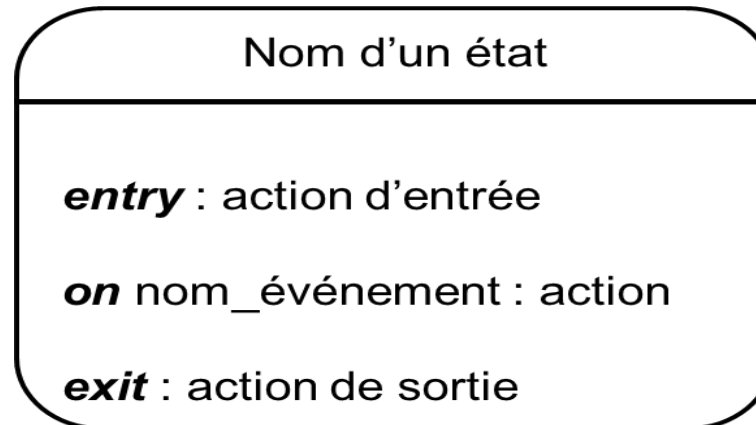


Actions dans les états

Les états peuvent également contenir des actions, exécutées soit

à l'entrée (entry) ou à la sortie (exit) de l'état

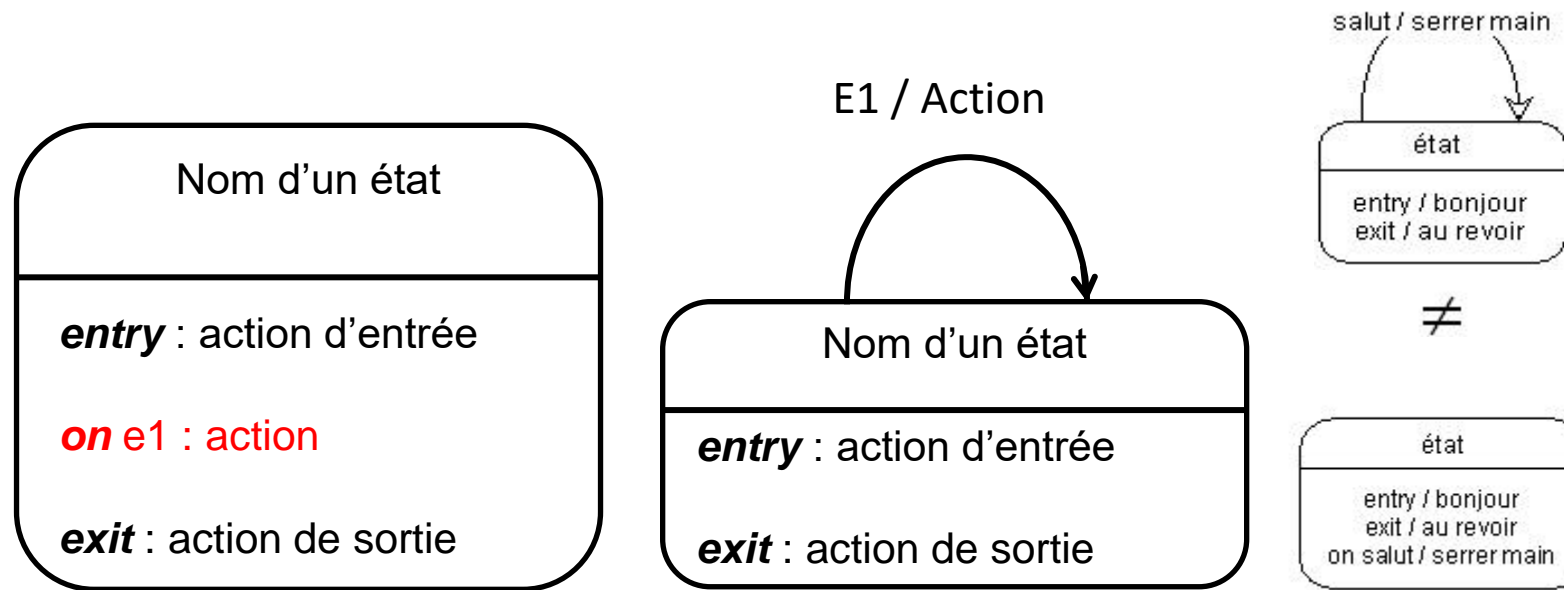
lorsqu'une occurrence d'événement interne (on) survient (exécutée lors de l'occurrence d'un événement qui ne conduit pas à un autre état)



Evénements internes

Evènement interne = évènement associé à un état donné

- Un événement interne n'entraîne pas l'exécution des actions de sortie et d'entrée, contrairement au déclenchement d'une transition réflexive.



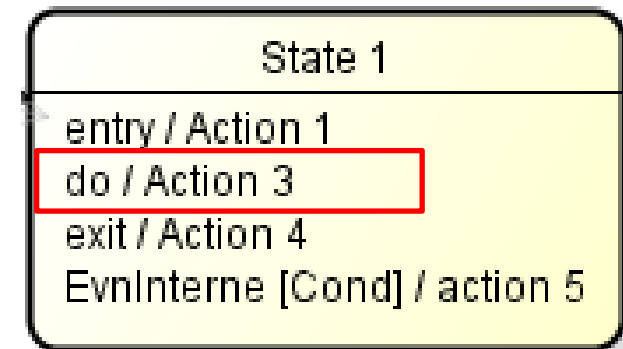
Activités

Contrairement à une action, une **activité** est une opération qui dure un certain temps. Elle est associée à un état, et

commence une fois entré dans l'état

s'exécute jusqu'à la fin si elle n'est pas interrompue par une transition sortante (donc tant que l'état ne change pas)

peut être interrompue, car elle ne modifie pas l'état de l'objet



Résumé

6 manières **d'associer une opération** à une transition

l'action **associée** à la transition d'entrée (op1)

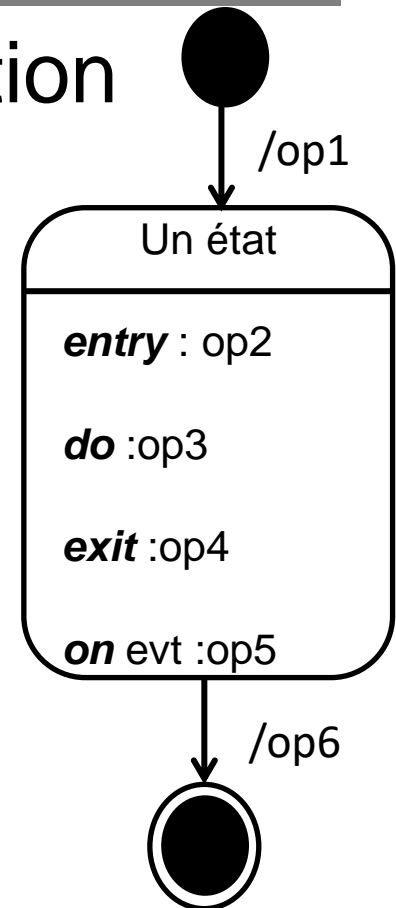
l'action **d'entrée** de l'état (op2)

l'**activité** dans l'état (op3)

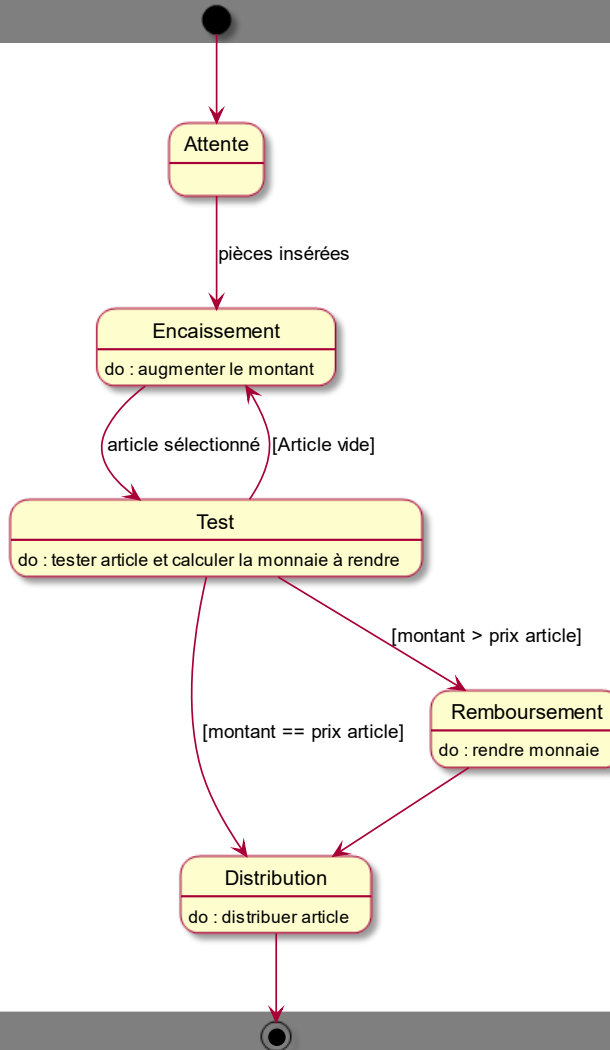
l'action de **sortie** de l'état (op4)

l'action **associée** aux **événements internes** (op5)

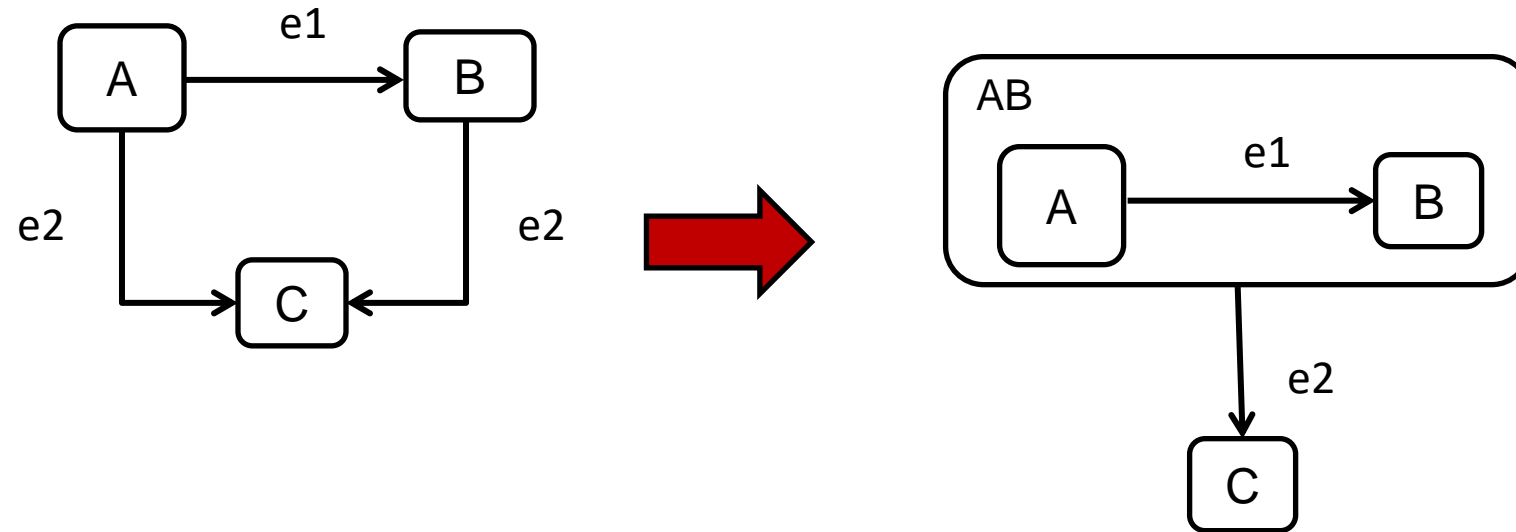
l'action **associée** à la **transition** de la sortie de l'état (op6)



Exemple : distributeur de boissons



Généralisation d'états



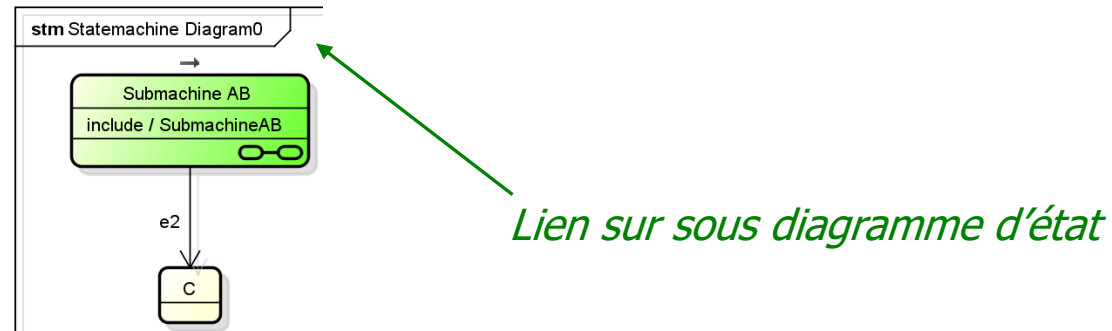
Les états A et B ont un comportement équivalents → Généralisation en AB

Etats imbriqués (composites)

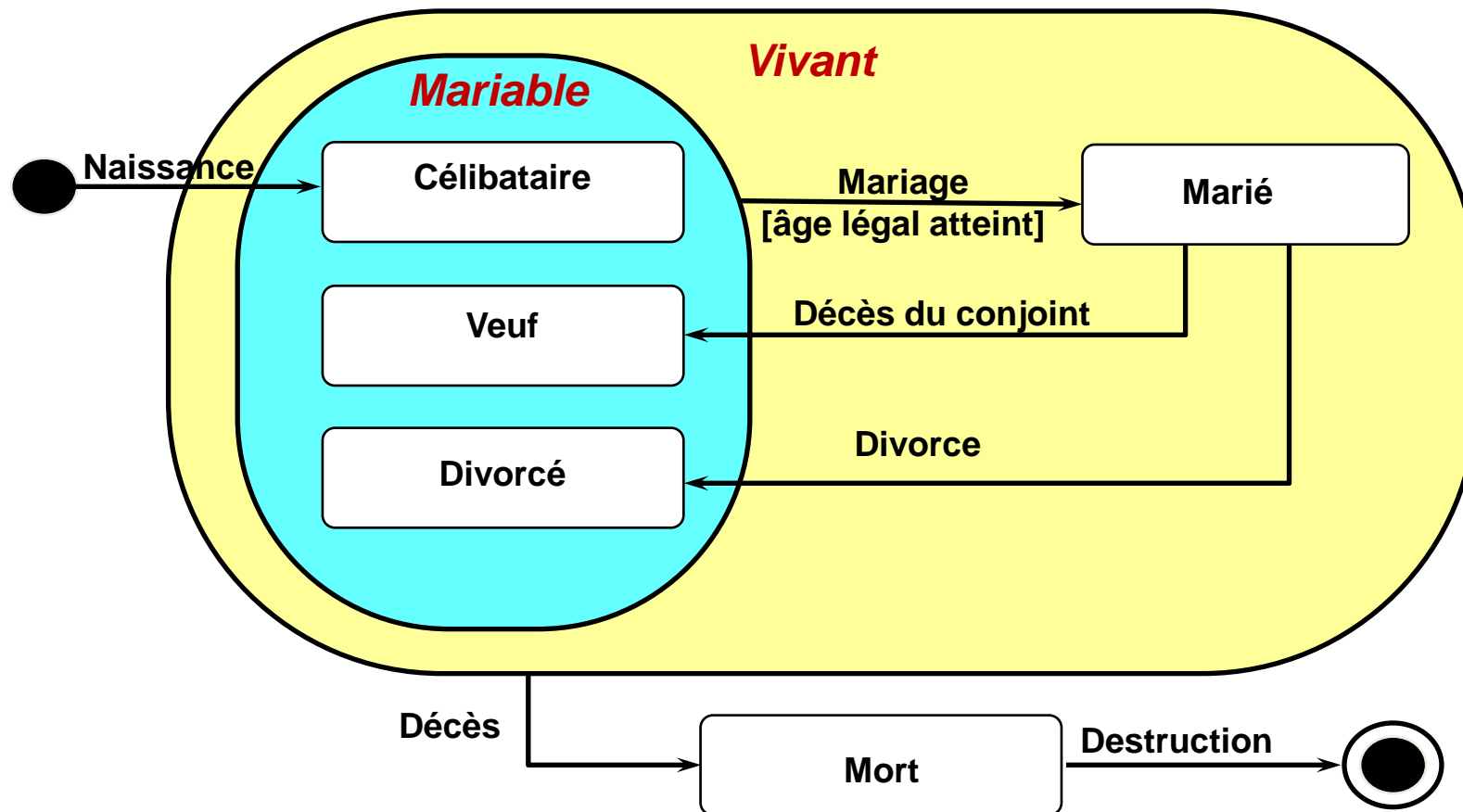
Si le diagramme d'état transition devient trop complexe, on peut utiliser des états imbriqués pour le simplifier

Un super-état ou état composite est un état qui englobe d'autres états appelés sous-états

Le nombre d'imbrications n'est pas limité, mais ne pas en abuser (lisibilité!)



Exemple d'états imbriqués



Exercice

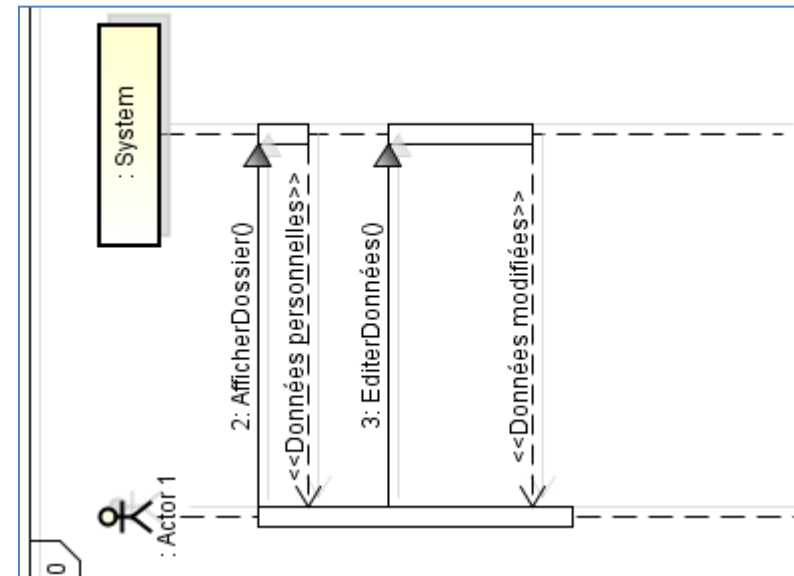
Faire le diagramme d'états d'une session Register / Login d'un nouvel utilisateur.

Timing Diagram

A partir d'un diagramme de séquence système (DSS) tourné de 90°, on peut créer un diagramme de temps.

Acteur + Ligne de vie → Timeline

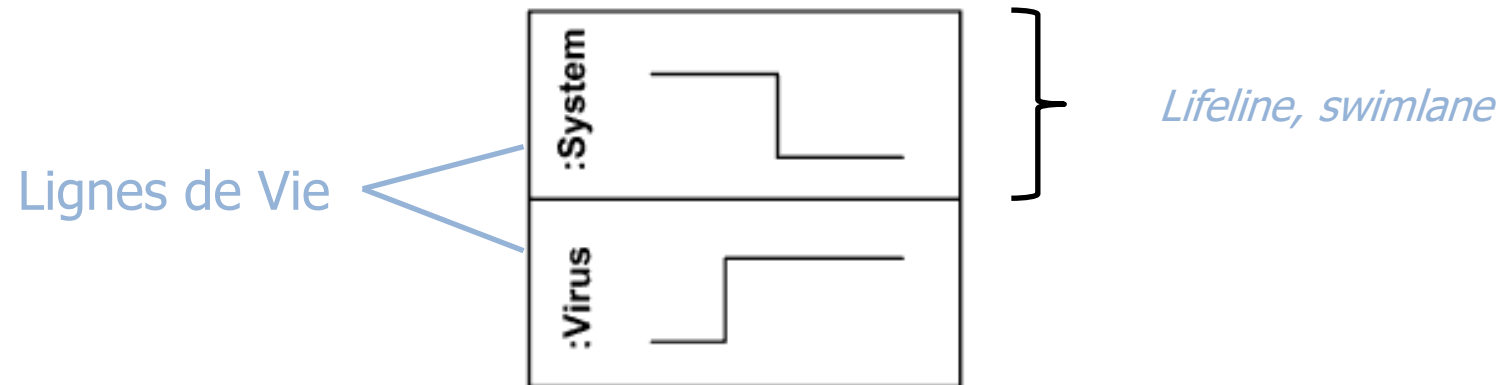
Acteur + Ligne de vie → Timeline



- Montre les **changements d'états** (continus ou discrets) d'un ou plusieurs acteurs
- Surtout destiné aux systèmes «**temps réel**»

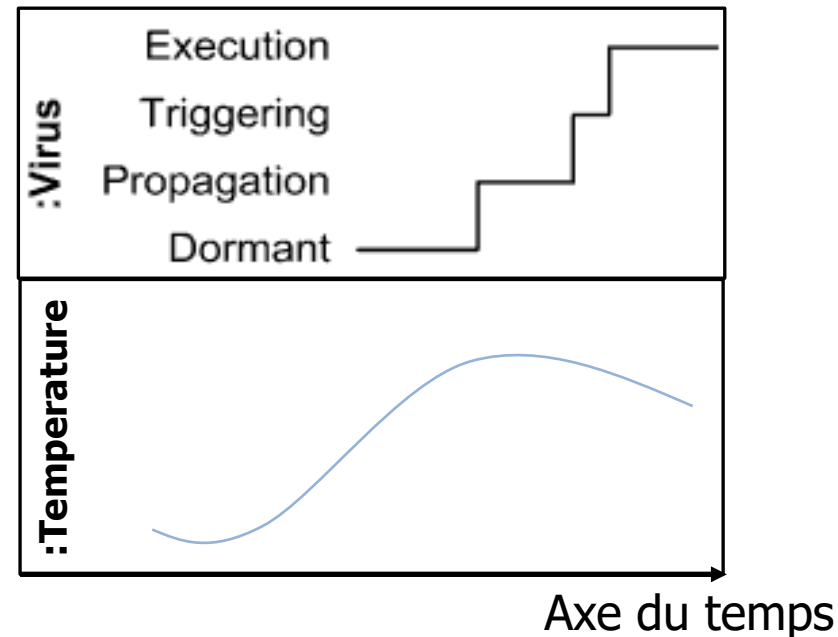
Ligne de vie

"Software modelers use a UML timing diagrams to precisely document a schedule of interactions or state changes in a system"



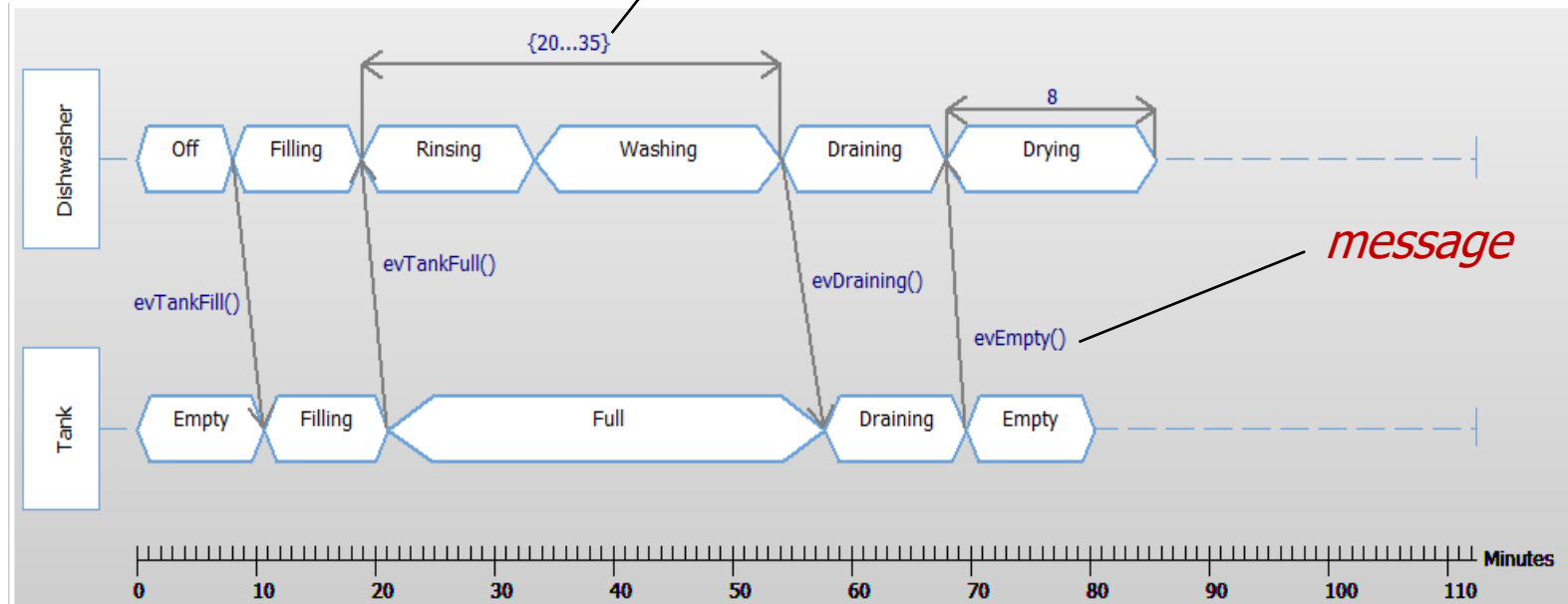
<http://www.uml-diagrams.org/timing-diagrams.html>

State / condition timeline



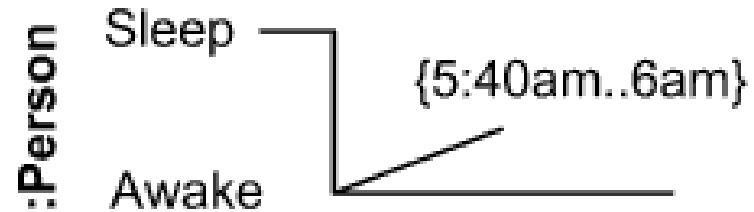
Un diagramme de temps peut illustrer les états, **discrets** ou **continus**, d'un composant, d'un objet, d'un attribut.

Contrainte de durée



Une **contrainte de durée**, indiquée entre accolades, par exemple $\{20 \dots 35\}$, $\{<10'\}$, $\{15' \dots\}$, est une condition qui invalide la suite si non respectée

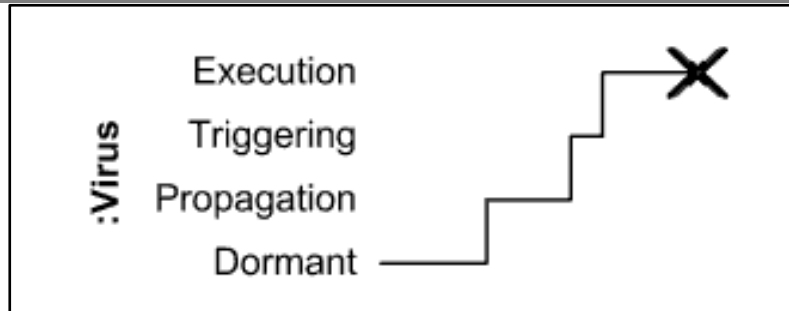
Contrainte de temps



*La personne doit se lever entre 5h40 et 6h du matin
sous peine ... de quitter le scénario nominal, rater sa vie*

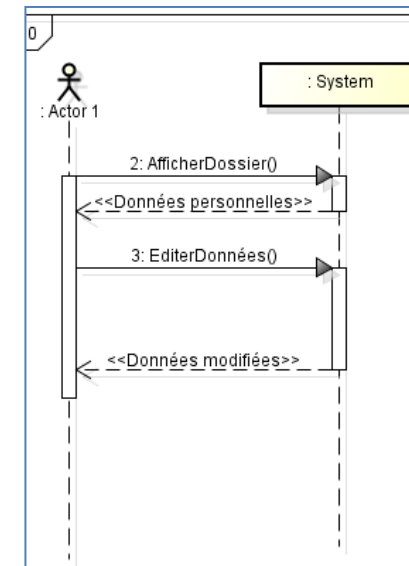
Trait entre intervalle $\{05h40..06h00\}$ ou borne $\{...6h00\}$ et *timeline*

End of timeline

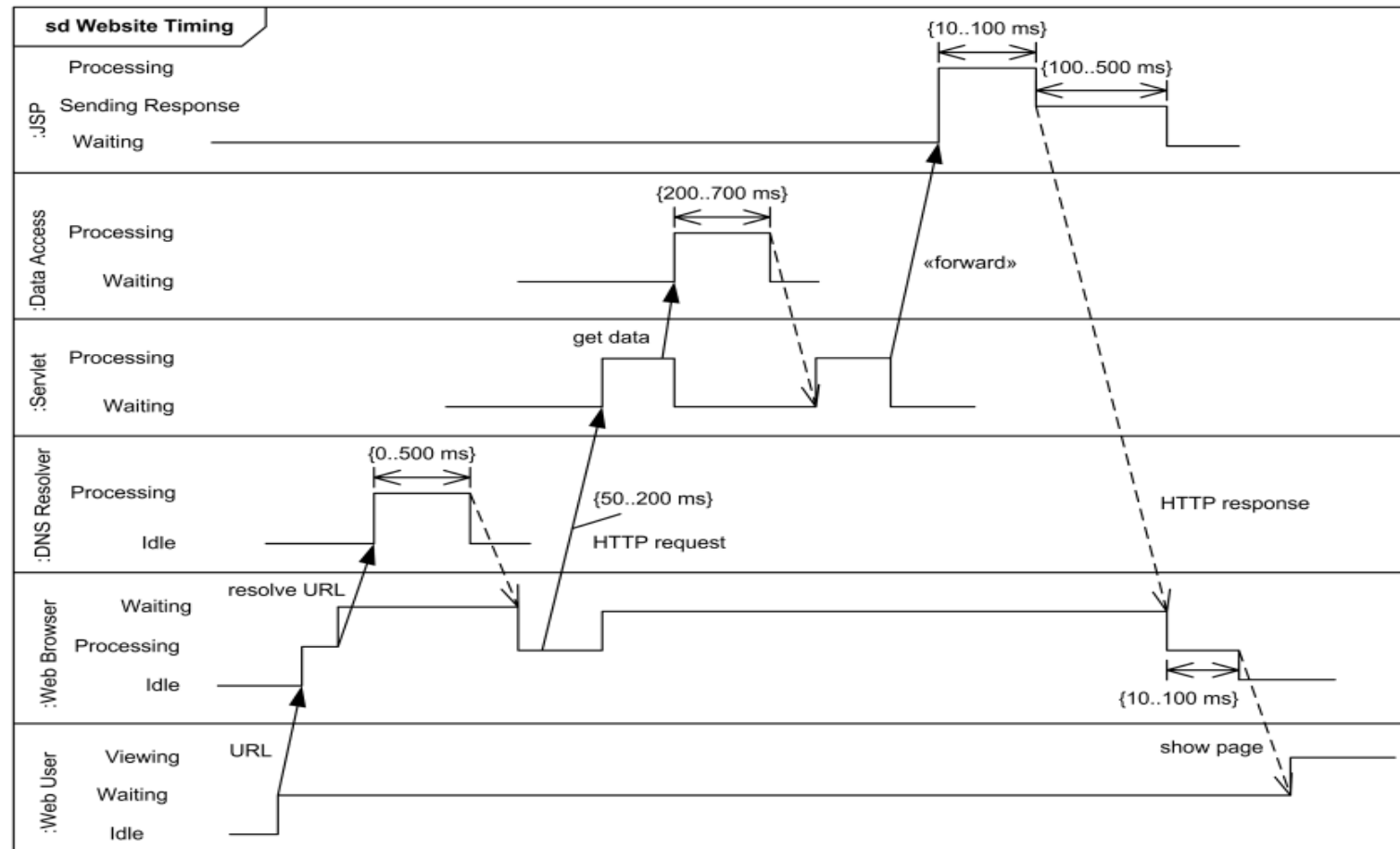


X marque la fin de vie de l'acteur Virus.

Correspond à la fin de la période d'activité du DSS.



Exemple



Exercice

Faire le **diagramme de temps** pour un événement ayant lieu à une heure précise (concert, match de hockey, etc)

- Achat du billet
- Recevoir le billet
- Aller à l'événement
- Assister à l'événement
- Etc.

Diagramme d'activités : rôle











*The term "**activity diagram**" simply refers to a standardized system of notation for a flowchart. An activity diagram is nothing more than a **flowchart**.*

Montre l'enchaînement des **actions** au sein d'une activité à l'aide de **flots de contrôle** et d'objets.

Diagramme d'activités : Notation

Activity Diagram Notation

Now that you've seen some examples, let's break down an activity diagram into its individual elements.

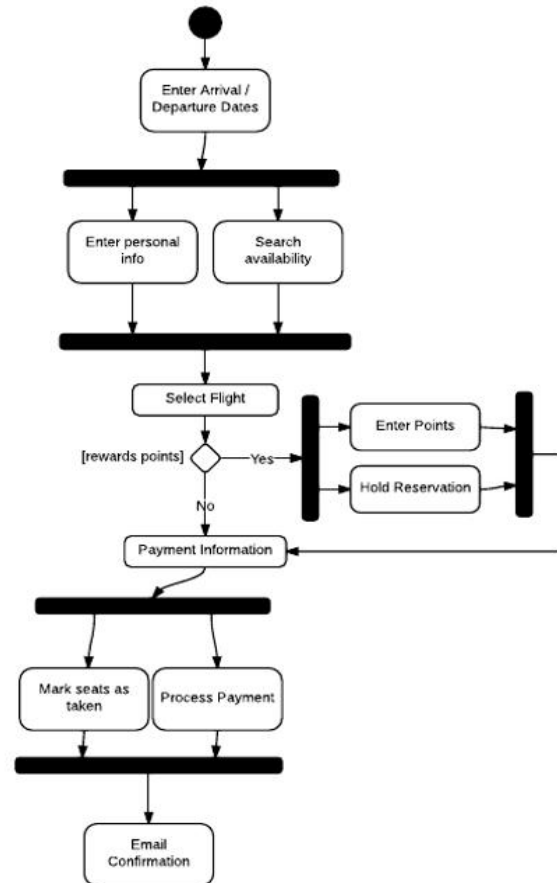
-  A black circle is the standard notation for an initial state before an activity takes place. It can either stand alone or you can use a note to further elucidate the starting point.
-  The black circle that looks like a selected radio button is the UML symbol for the end state of an activity. As shown in two examples above, notes can also be used to explain an end state.
-  The activity symbols are the basic building blocks of an activity diagram and usually have a short description of the activity they represent.
-  Arrows represent the direction flow of the flow chart. The arrow points in the direction of progressing activities.
-  A join combines two concurrent activities back into a flow where only one activity is happening at a time.
-  A fork splits one activity flow into two concurrent activities.
-  Condition text is placed next to a decision marker to let you know under what condition an activity flow should split off in that direction.
-  A marker shaped like a diamond is the standard symbol for a decision. There are always at least two paths coming out of a decision and the condition text lets you know which options are mutually exclusive.
-  The final flow marker shows the ending point for a process in a flow. The difference between a final flow node and the end state node is that the latter represents the end of all flows in an activity.
-  The shape used for notes.

Région
interruptible

interruption

Diagramme d'activités : exemple

Réservation vol



Connexion site sécurisé

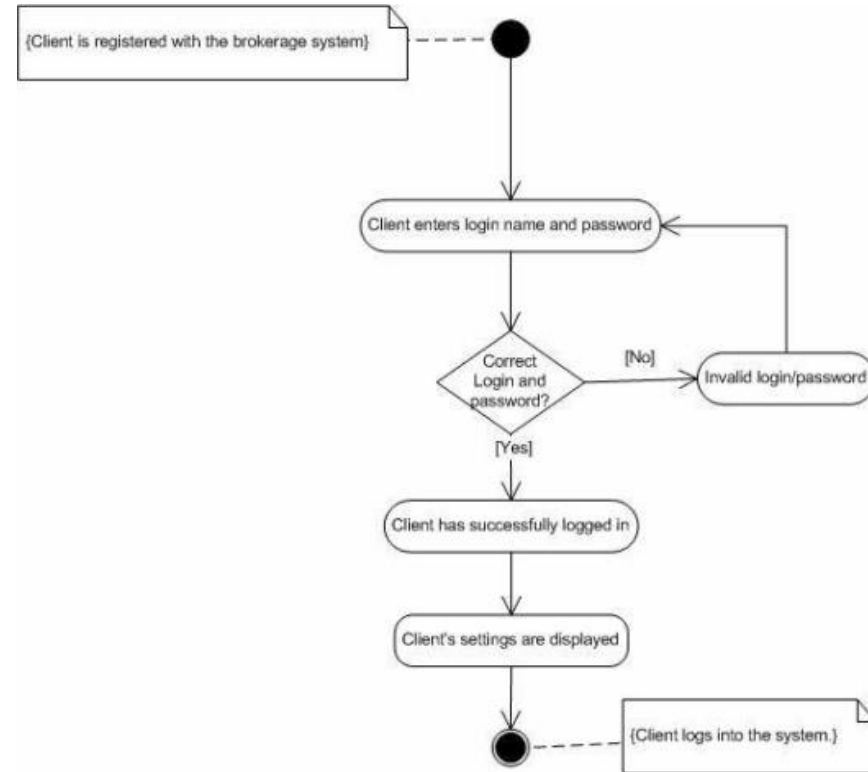
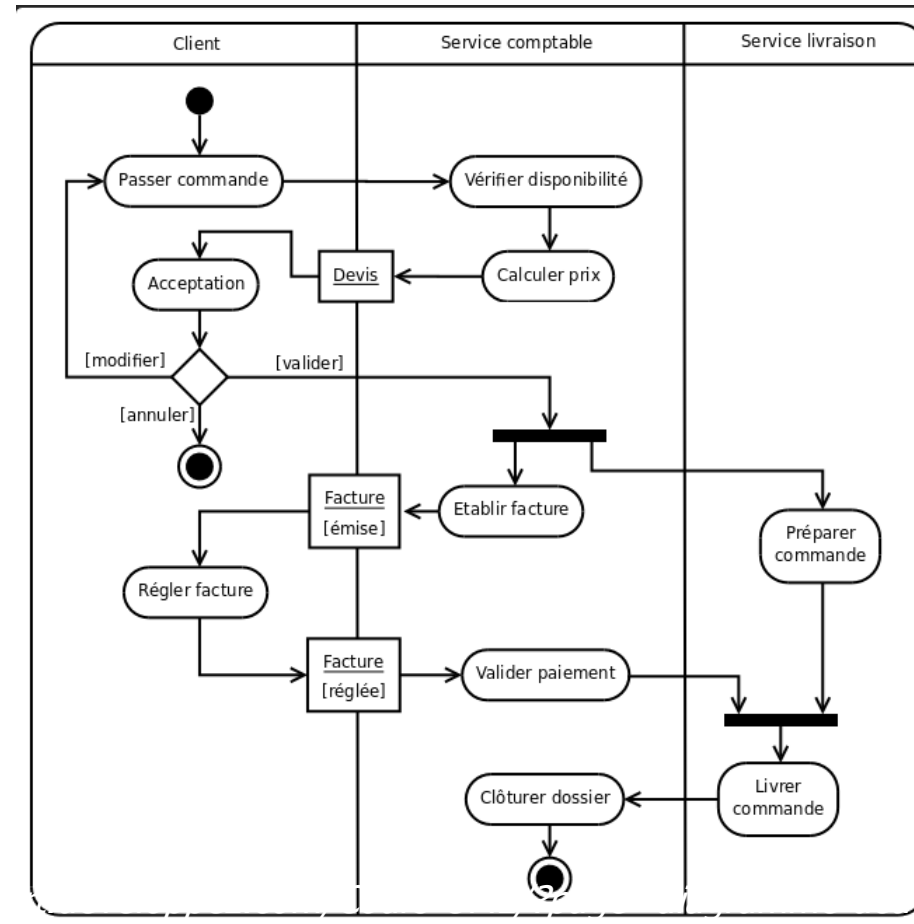


Diagramme d'activités : partitions



Exercice

Faire le diagramme d'activités pour l'enregistrement d'un utilisateur à un événement particulier.

- Login / register
- Browse sessions / events
- Etc.