

DIAGRAMME D'ETATS- TRANSITIONS ET D'ACTIVITES

DIAGRAMME D'ETATS-TRANSITIONS

DIAGRAMME D'ETATS-TRANSITIONS

DEFINITION:

- 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 sous l'effet d'une occurrence **d'événement**

Diagramme d'états-transitions

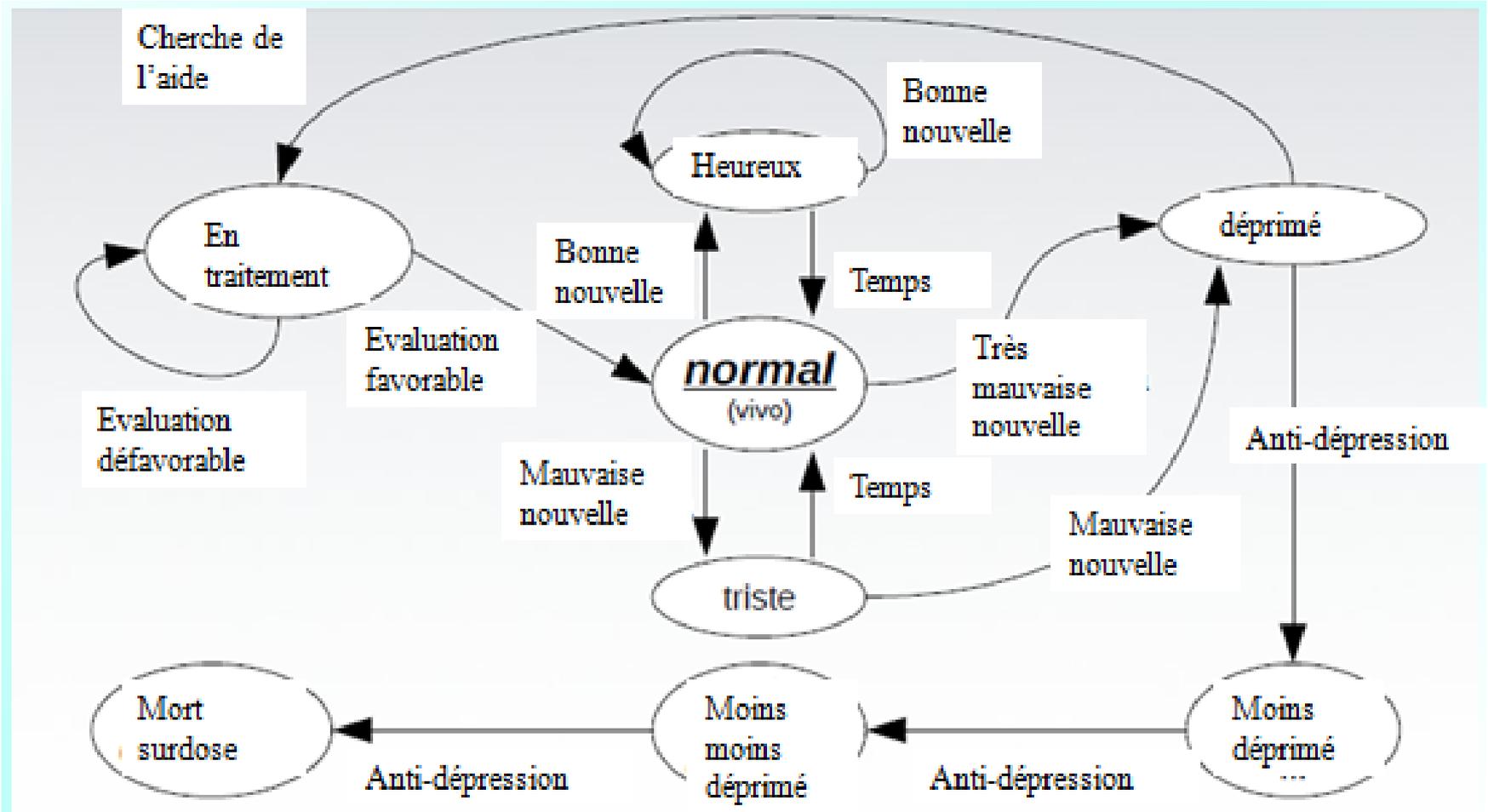


DIAGRAMME D'ETATS-TRANSITIONS

DEFINITION:

- Description du **cycle de vie** d'un objet d'une classe :
 - Les **états** qui peuvent être pris par les **objets** d'une classe
 - Les **événements** qui provoquent la **transition d'un état à un autre**
 - Les **actions** subies/provoquées qui accompagnent un **changement d'état**
 - Les **activités** qui surviennent tant que l'objet est dans un **état donné**.

DIAGRAMME D'ETATS-TRANSITIONS

OBJECTIFS:

- Globalement : étudier les états d'un Système
- Plus particulièrement : 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

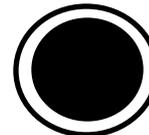
DIAGRAMME D'ETATS-TRANSITIONS

NOTION D'ETAT:

- Un 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



Etat initial

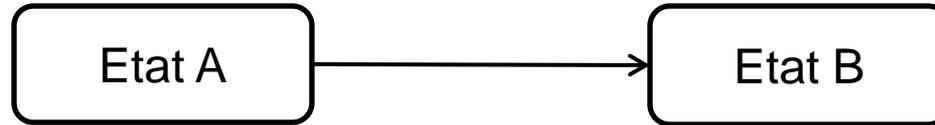


Etat final

DIAGRAMME D'ETATS-TRANSITIONS

NOTION DE TRANSITION:

- Les états sont reliés par des connexions unidirectionnelles appelées **transitions**



- Ex : place de parking

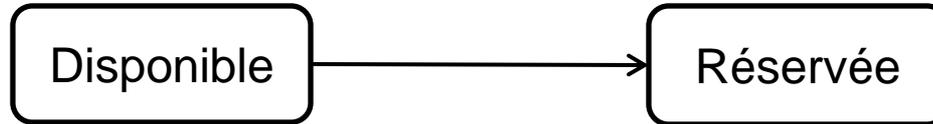


DIAGRAMME D'ETATS-TRANSITIONS

NOTION D'EVENEMENT:

- Un événement correspond à l'occurrence d'une situation donnée dans le domaine étudié
- Un événement est une information instantanée qui doit être traitée à l'instant où il se produit

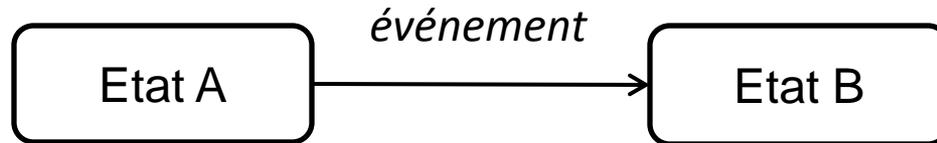


DIAGRAMME D'ETATS-TRANSITIONS

NOTION D'ÉVÉNEMENT:

- Syntaxe d'un événement :
 - Nom de l'événement (Nom de paramètre : Type,.....)
- La description complète d'un événement est donnée par :
 - Nom de l'événement
 - Liste des paramètres
 - Objet expéditeur
 - Objet destinataire
 - Description textuelle

DIAGRAMME D'ETATS-TRANSITIONS

NOTION D'EVENEMENT:

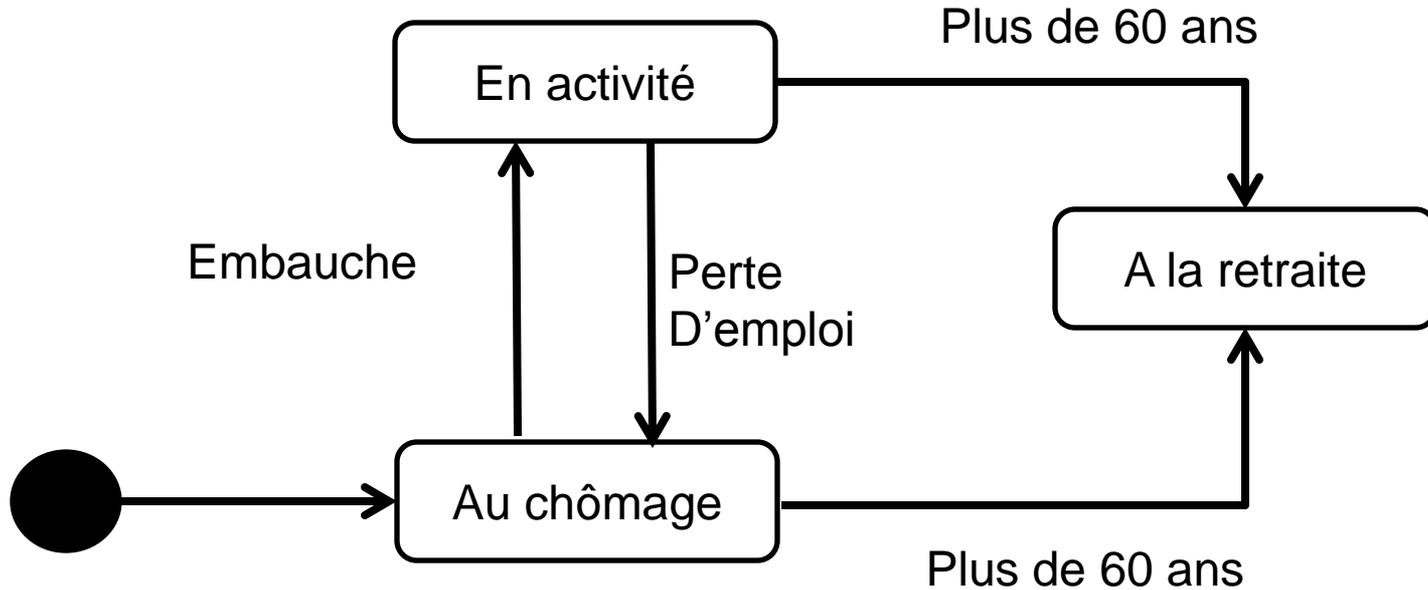


DIAGRAMME D'ETATS-TRANSITIONS

EN RESUME :

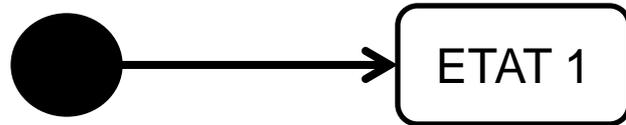
- Etat d'un objet :
 - Situation d'un objet que l'on désire connaître et gérer.
- Transition :
 - Passage d'un objet d'un état à un autre. Elle est déclenchée par un événement
- Evénement :
 - Stimulus qui provoque une (ou plusieurs) transition(s). A chaque stimulus peut correspondre une **action** responsable des modifications de l'objet (les valeurs des attributs)

DIAGRAMME D'ETATS-TRANSITIONS

ETATS SPECIAUX:

- 2 états prédéfinis :
 - état de démarrage : obligatoire, unique
 - état de fin : optionnel, peut-être multiple

Création de l'objet



Fin de vie de l'objet



DIAGRAMME D'ETATS-TRANSITIONS

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.

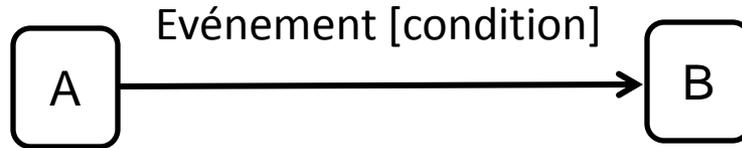


DIAGRAMME D'ETATS-TRANSITIONS

COMMUNICATION ENTRE OBJETS PAR EVENEMENTS :

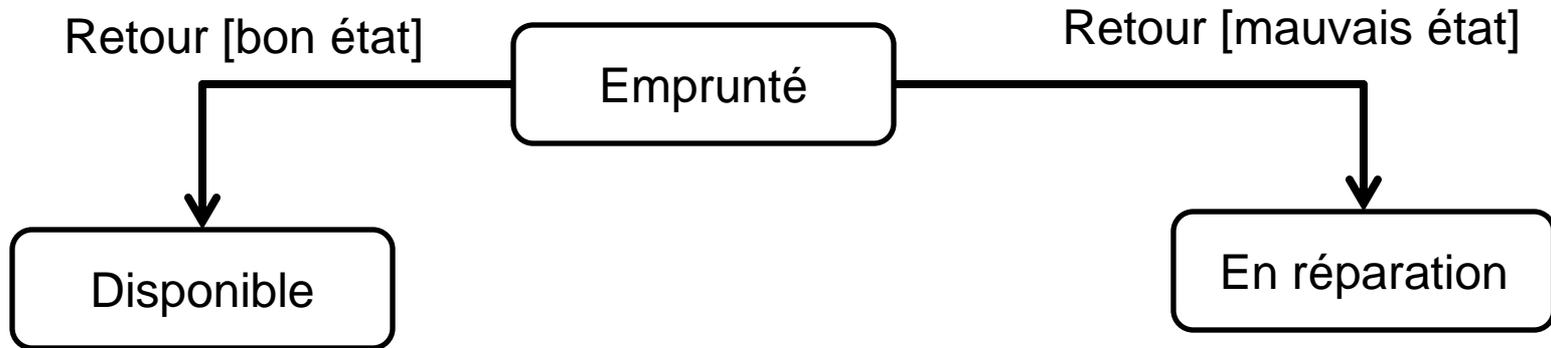


DIAGRAMME D'ETATS-TRANSITIONS

NOTIONS D'OPERATION ET D'ACTION :

- Action et activités : Le lien entre les **opérations** définies dans la spécification d'une **classe** et les **événements** apparaissant dans la diagramme d'états-transitions
- Chaque transition peut avoir une action à exécuter lorsqu'elle est déclenchée
- L'action est considérée comme instantanée et atomique
- Une action correspond à l'exécution d'une des opérations déclarées dans la classe de l'objet destinataire de l'événement.

- il pleut / ouvrir parapluie

- Salut/serrer main

- L'action a accès aux paramètres de l'événement ainsi qu'aux attributs de l'objet sur lequel elle s'applique

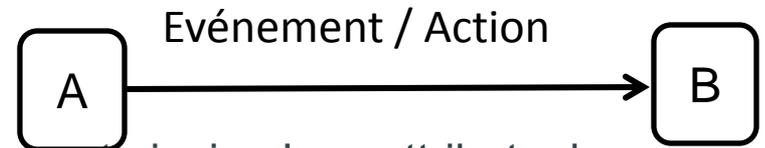


DIAGRAMME D'ETATS-TRANSITIONS

ACTIONS DANS UN ETAT :

- Les états peuvent également contenir des actions :

Elles sont exécutées :

- 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)

DIAGRAMME D'ETATS-TRANSITIONS

ACTIONS DANS UN ETAT :

- Représentation :

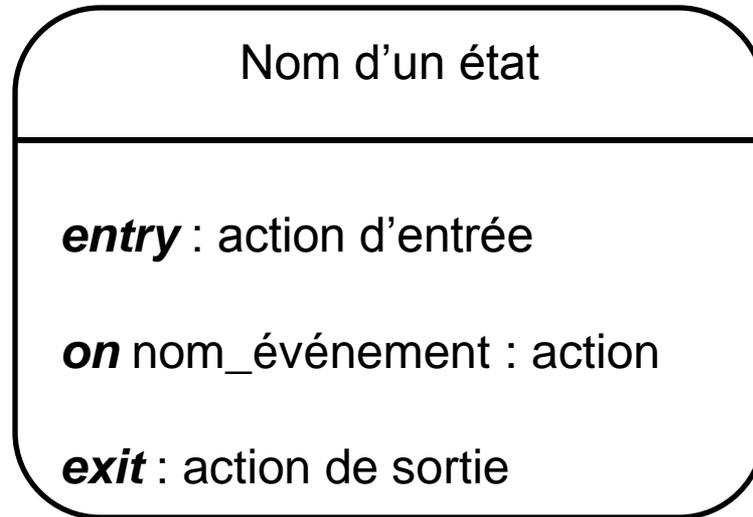
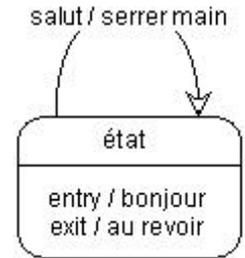
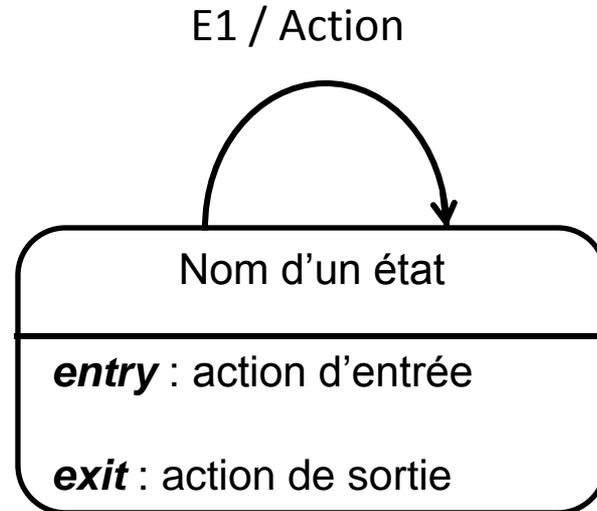
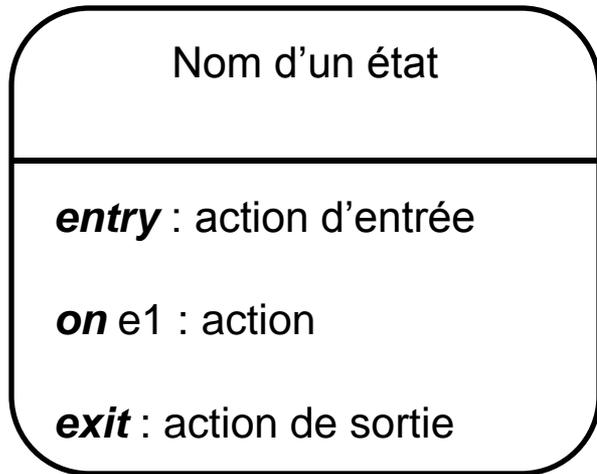


DIAGRAMME D'ETATS-TRANSITIONS

OPERATIONS, ACTIONS ET ACTIVITES :

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



≠



DIAGRAMME D'ETATS-TRANSITIONS

OPERATIONS, ACTIONS ET ACTIVITES :

- Contrairement à une action, une activité est une opération qui dure un certain temps
- Les activités sont associées aux états
 - commencent quand on est entré dans l'état
 - s'exécutent jusqu'à la fin si elles ne sont pas interrompues par une transition sortante (donc tant que l'état ne change pas)
 - peuvent être interrompues car elles ne modifient pas l'état de l'objet
- Les activités sont notées dans la partie inférieure de l'état

DIAGRAMME D'ETATS-TRANSITIONS

OPERATIONS, ACTIONS ET ACTIVITES :

- Lorsqu'une activité se termine, les transitions automatiques (sans événement), mais éventuellement protégées par des gardes, sont déclenchées

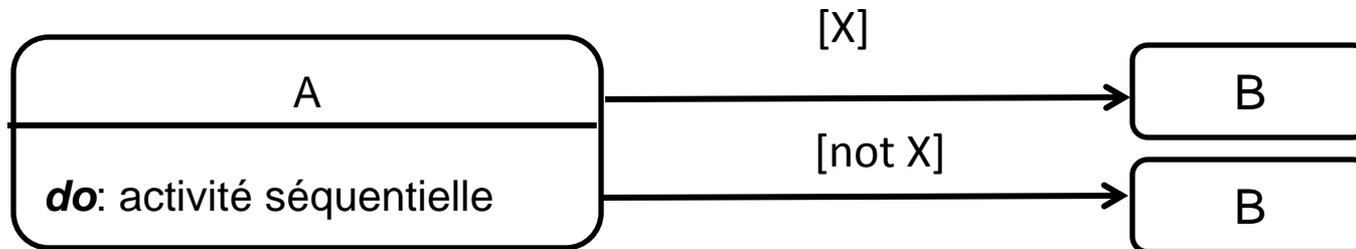
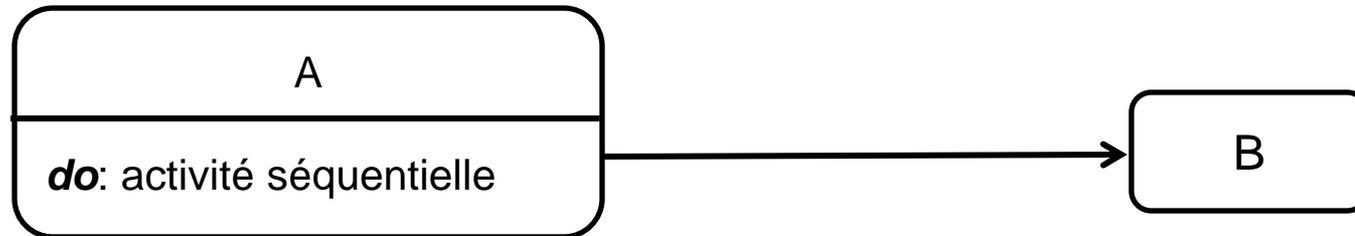
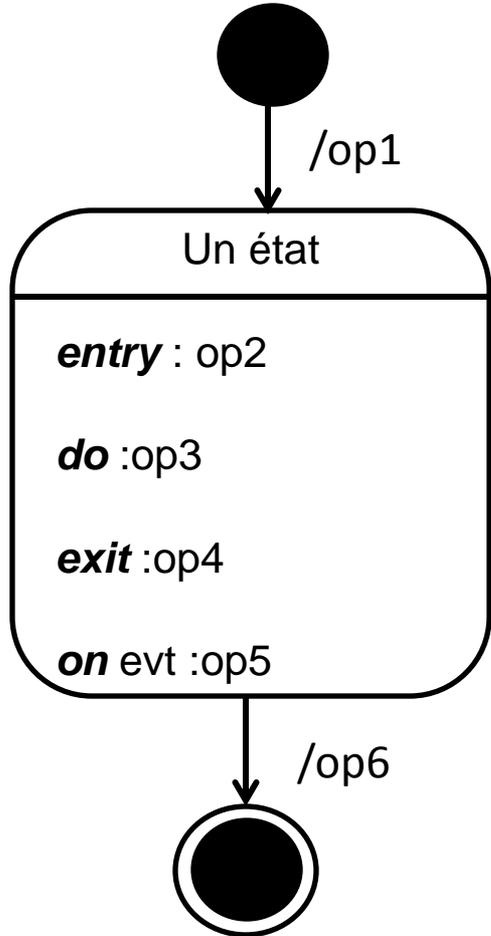


DIAGRAMME D'ETATS-TRANSITIONS



OPERATIONS, ACTIONS ET ACTIVITES :

- 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)

DIAGRAMME D'ETATS-TRANSITIONS

Exemple DAB:

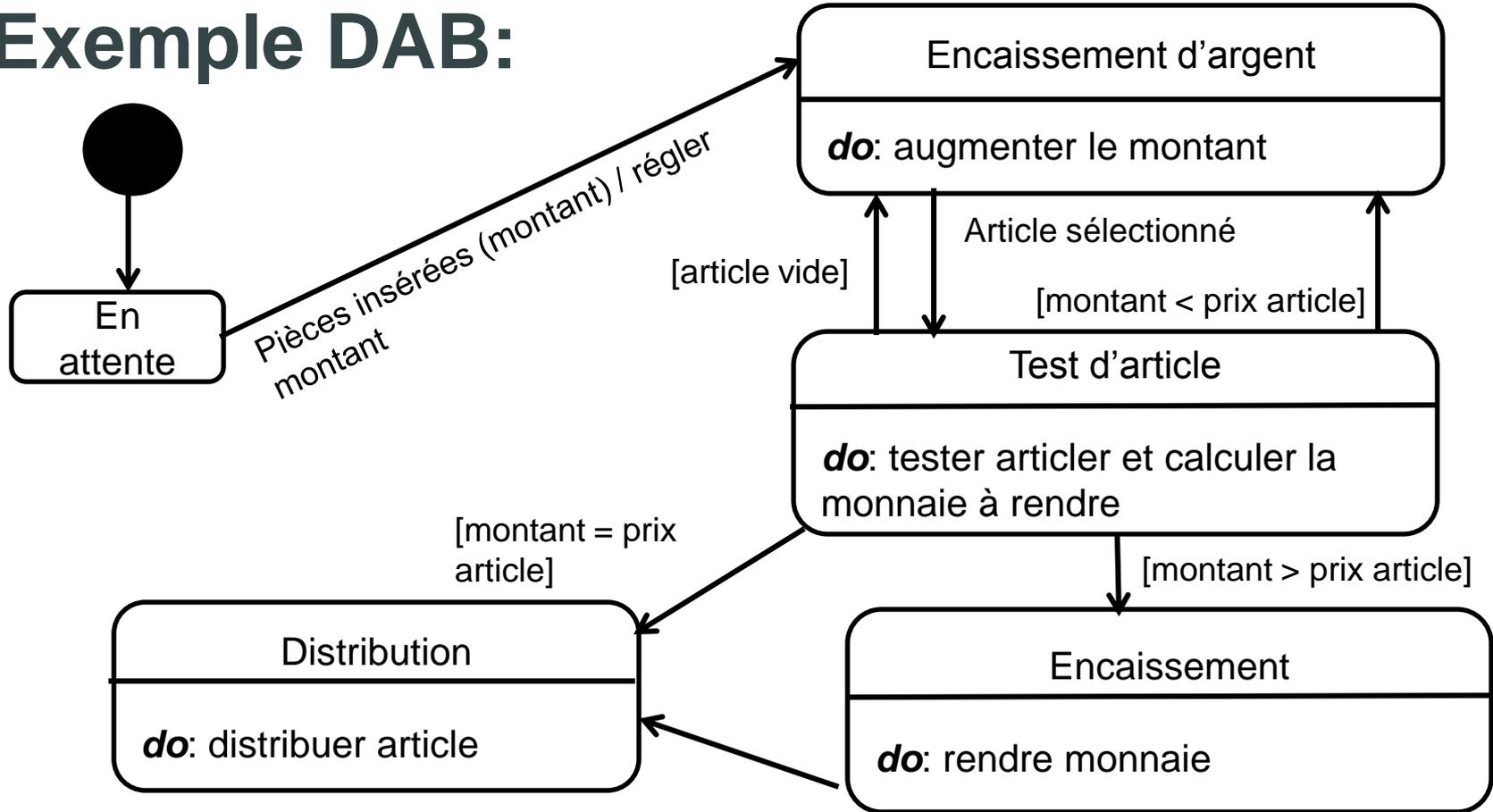


DIAGRAMME D'ETATS-TRANSITIONS

Généralisation d'états:

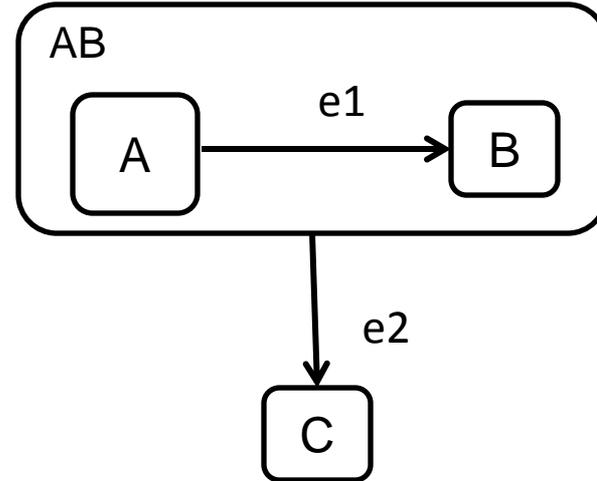
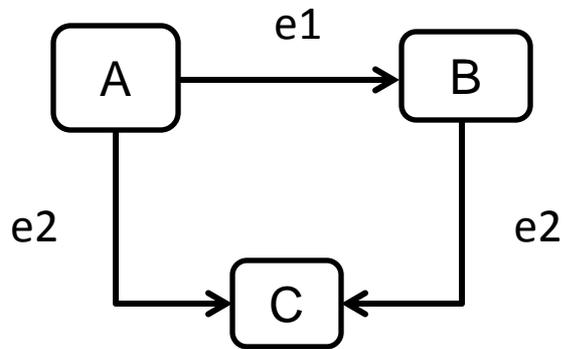


DIAGRAMME D'ETATS-TRANSITIONS

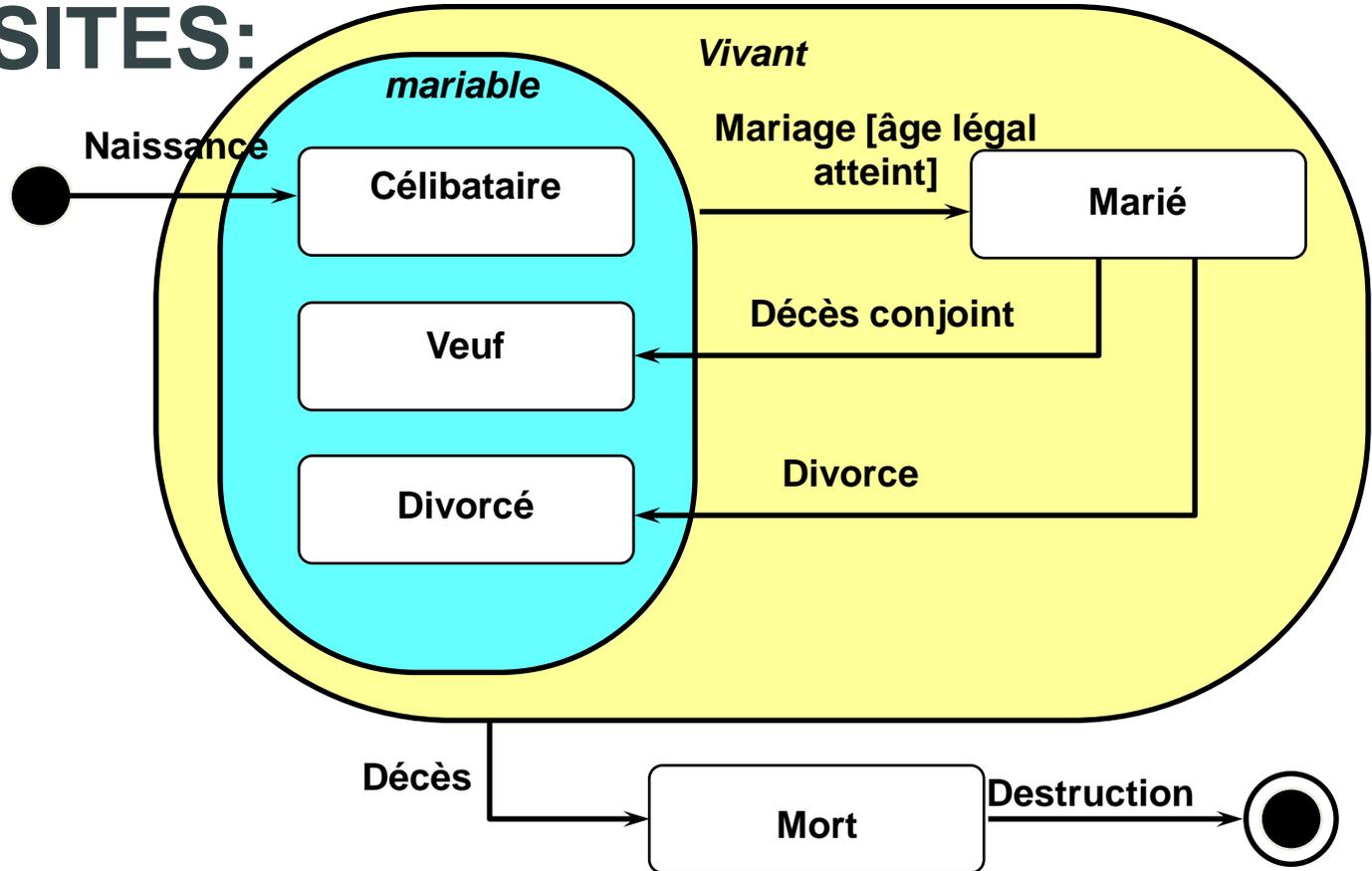
ETATS IMBRIQUES - COMPOSITES:

- 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'imbrication n'est pas limité (ne pas abusé sinon problème de lisibilité)

DIAGRAMME D'ETATS-TRANSITIONS

ETATS IMBRIQUES

COMPOSITES:



TD Diagramme d'Etats-transitions

- On considère une boîte de vitesses automatique de voiture. La boîte au démarrage est au point mort. La marche arrière ainsi que la position *parking* peuvent être enclenchées à partir du point mort. La première marche avant peut également être enclenchée à partir du point mort. En revanche, les autres marches avant, la seconde et la troisième, sont enclenchées en séquence: $1 \rightarrow 2 \rightarrow 3$ pour une accélération, et $3 \rightarrow 2 \rightarrow 1$ pour une décélération. Seules la marche arrière, la position *parking* et la première marche avant peuvent être ramenées directement au point mort.

TD Diagramme d'Etats-transitions

- On veut représenter par un diagramme d'états, la vie d'un exemplaire de livre dans une bibliothèque. Quand un exemplaire est créé, il entre dans le « hors-prêt » (consultation sur place seulement) s'il est le seul exemplaire de l'ouvrage ou devient disponible pour le prêt sinon. Quand il y a plusieurs exemplaires le bibliothécaire peut décider de changer l'exemplaire en « hors prêt ».

Quand un exemplaire prêté ne revient pas après les relances il est considéré comme perdu et retiré de la base un an après. S'il est retrouvé, il redevient disponible pour le prêt.

En cas de détérioration, l'exemplaire est sorti des rayons et marqué « à enlever ». Il peut être renouvelé ou retiré de la base. Il n'y a pas de notion de réservation d'un ouvrage non disponible.

DIAGRAMME D'ACTIVITES

DIAGRAMME D'ACTIVITES

DEFINITION:

- Variante des diagrammes d'états-transitions
- Le diagramme d'activité permet de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.
- Le passage d'une activité vers une autre est matérialisé par une transition. Les transitions sont déclenchées par la fin d'une activité et provoquent le début immédiat d'une autre (elles sont automatiques).
- En théorie, tous les mécanismes dynamiques pourraient être décrits par un diagramme d'activités, mais seuls les mécanismes complexes ou intéressants méritent d'être représentés.

DIAGRAMME D'ACTIVITES

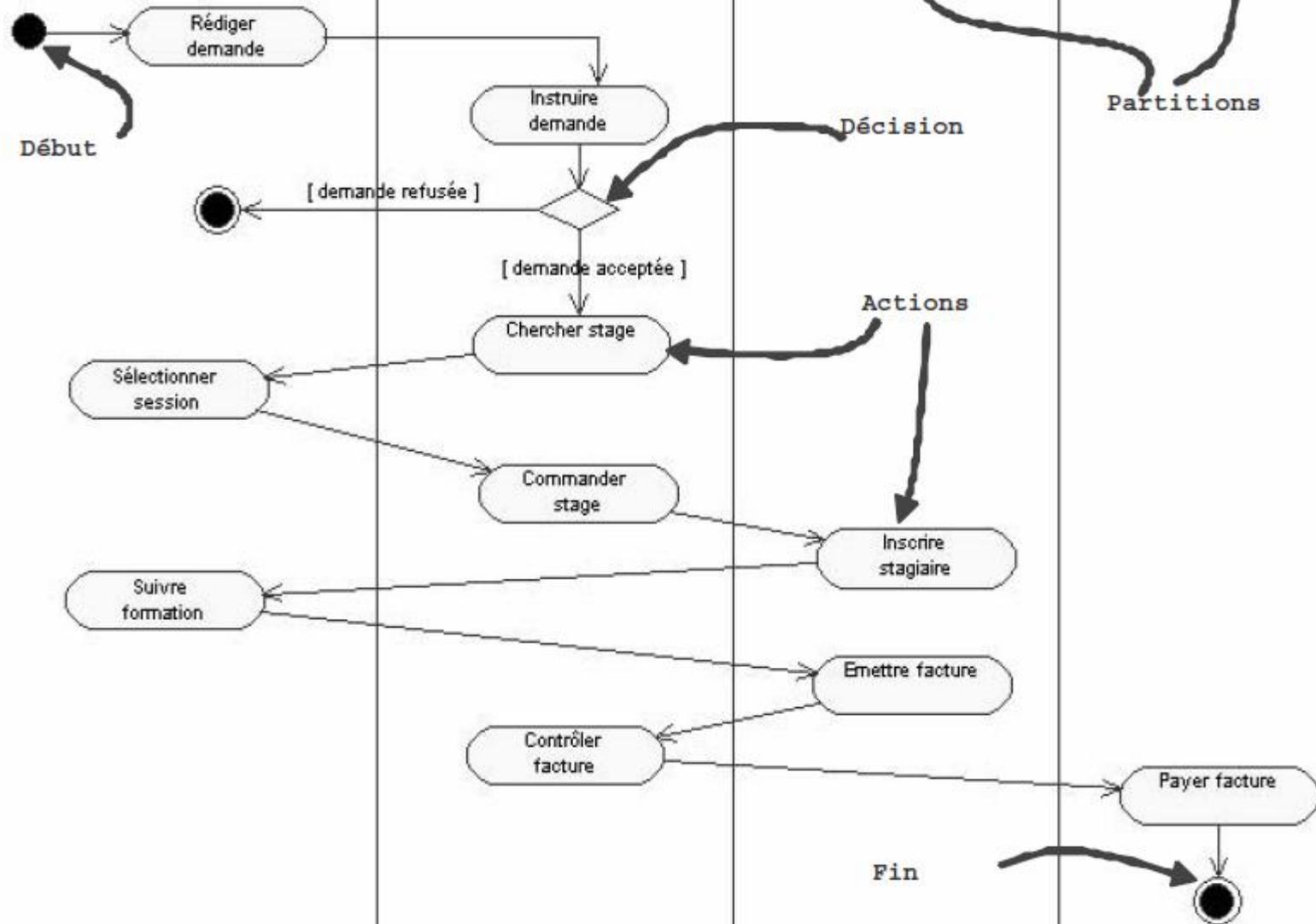
- Pour montrer le flux des opérations qui s'enchaînent dans un processus interne du système.
- Ce diagramme est le comment de la description d'un système que nous avons évité dans les diagrammes de cas d'utilisation.
- Ce n'est pas un diagramme d'états-transitions
 - Le diagramme d'états-transitions met l'accent sur l'évolution d'un objet au cours du temps.
 - Le diagramme d'activités se focalisent sur le flux d'activités concourant à la réalisation d'un processus

: Employé

: Responsable formation

: Organisme de formation

: Comptable



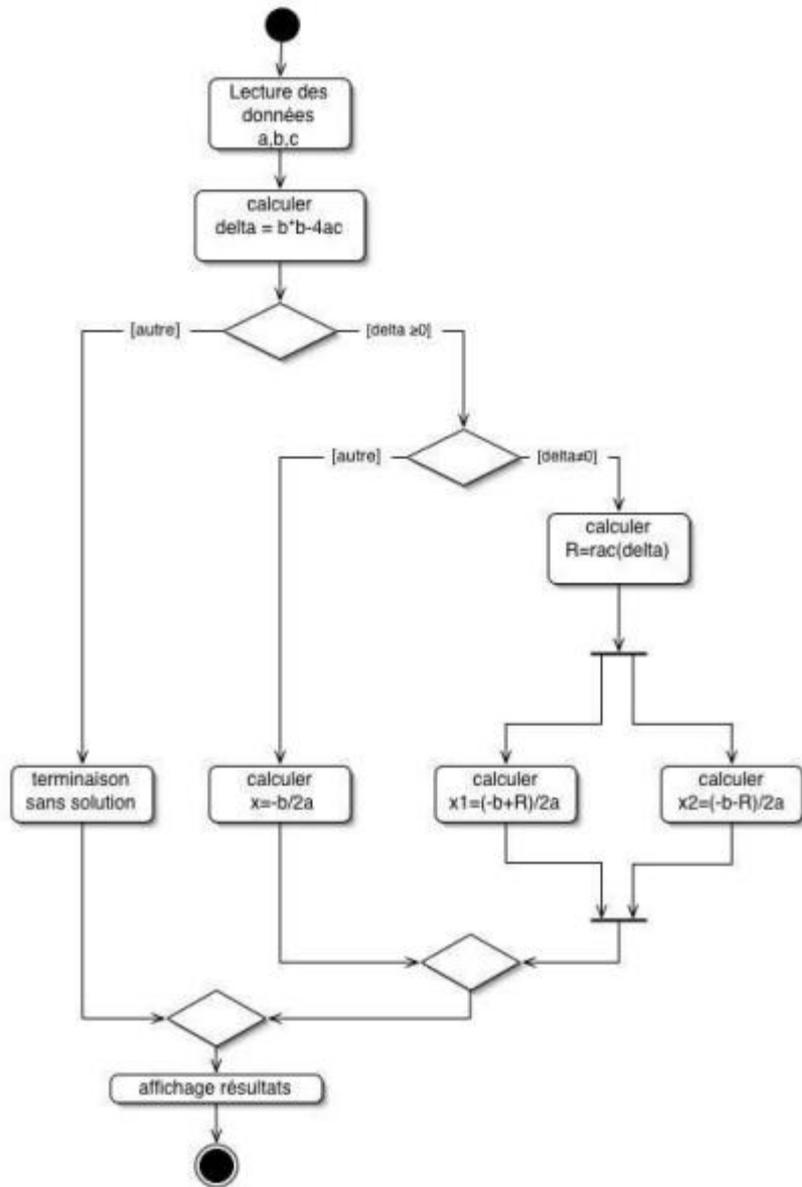


DIAGRAMME D'ACTIVITES

- Flux d'objet :
 - Un flux d'objet représente la participation d'un objet dans un flux de contrôle.
 - Les flux d'objets sont représentés par des relations de dépendance entre objets et états d'action ou d'activités.
 - Les flux d'objets sont représentés par des flèches :
- Le diagramme ainsi obtenu est très intéressant, puisqu'il fait le pont entre les trois axes de modélisation : fonctionnel (actions), dynamique (flots) et statique (entités et partitions) !

: Employé

: Responsable formation

: Organisme de formation

: Comptable

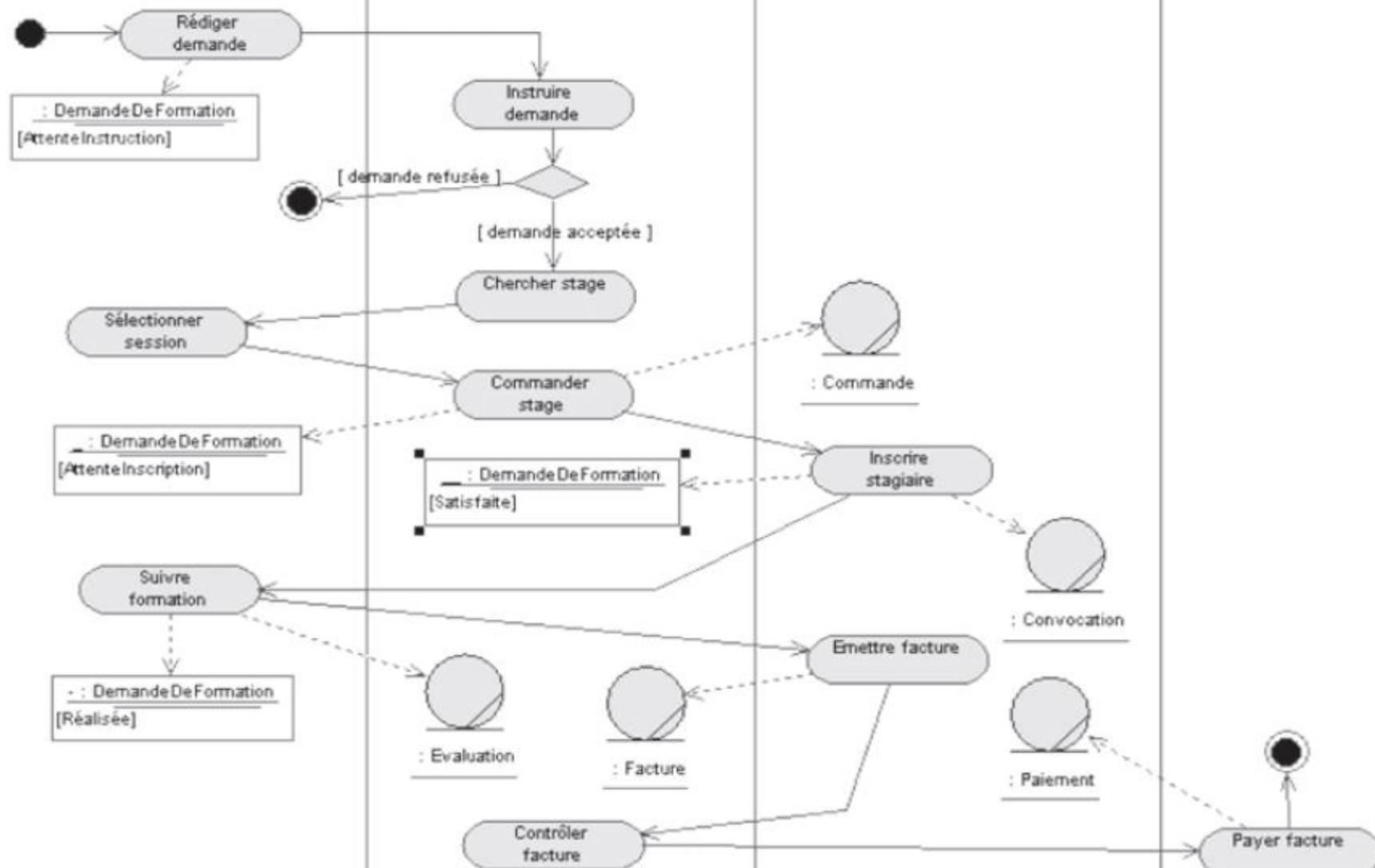


DIAGRAMME D'ACTIVITES

GARDES:

- Les gardes sont les bales des transitions dont elles valident le déclenchement

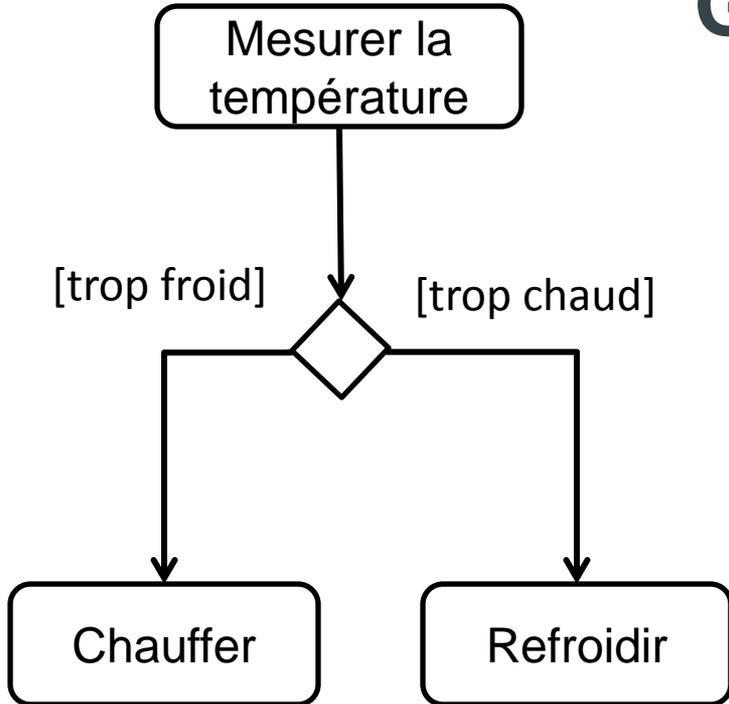
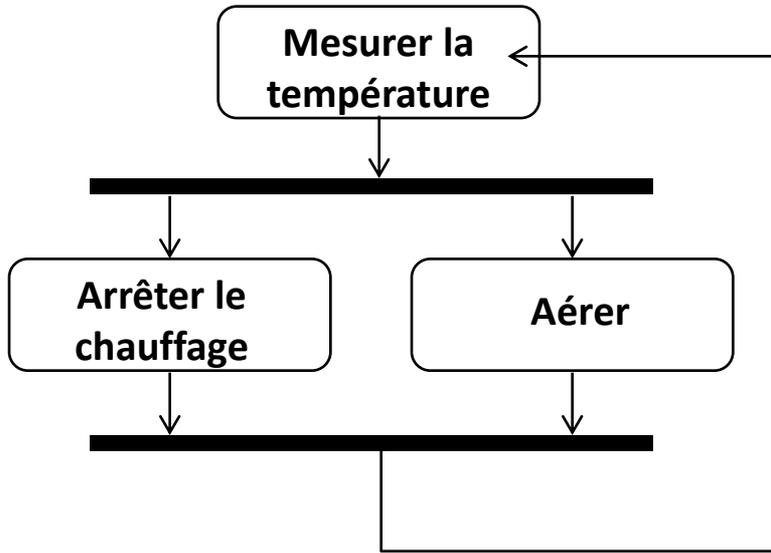


DIAGRAMME D'ACTIVITES

PARALLELISME ET SYNCHRONISATION :

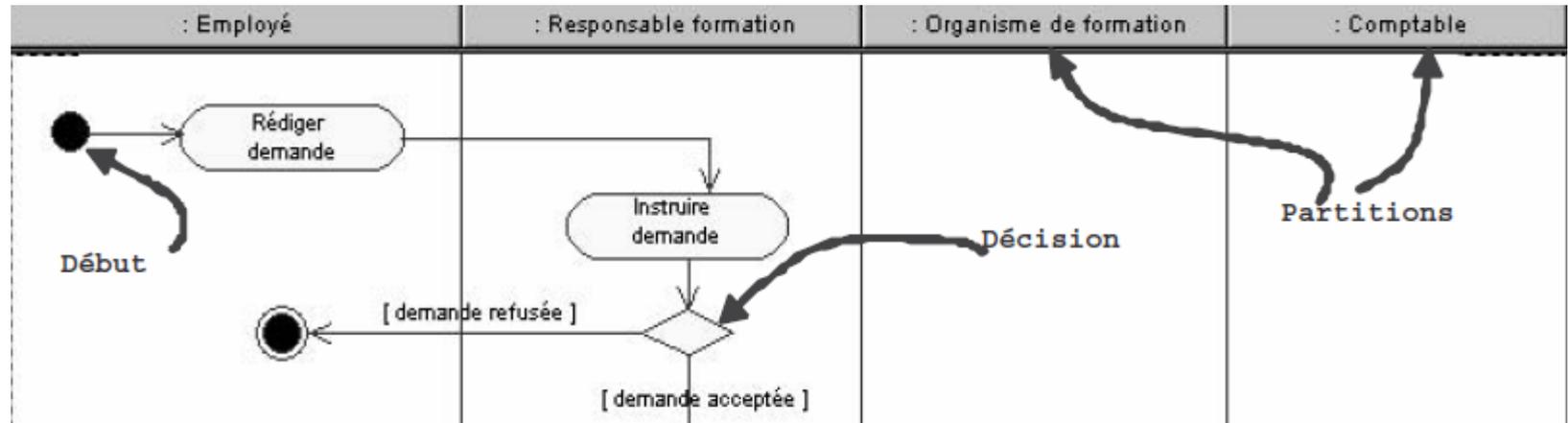


- On représente une synchronisation et le parallélisme par une barre verticale ou horizontale:  
- Parallélisme utilisé pour représenter des déroulements parallèles.
- Synchronisation utilisée pour représenter la fin des traitements parallèles.

DIAGRAMME D'ACTIVITES

DECOUPAGE (COULOIR D'ACTIVITE – SWIM LANE):

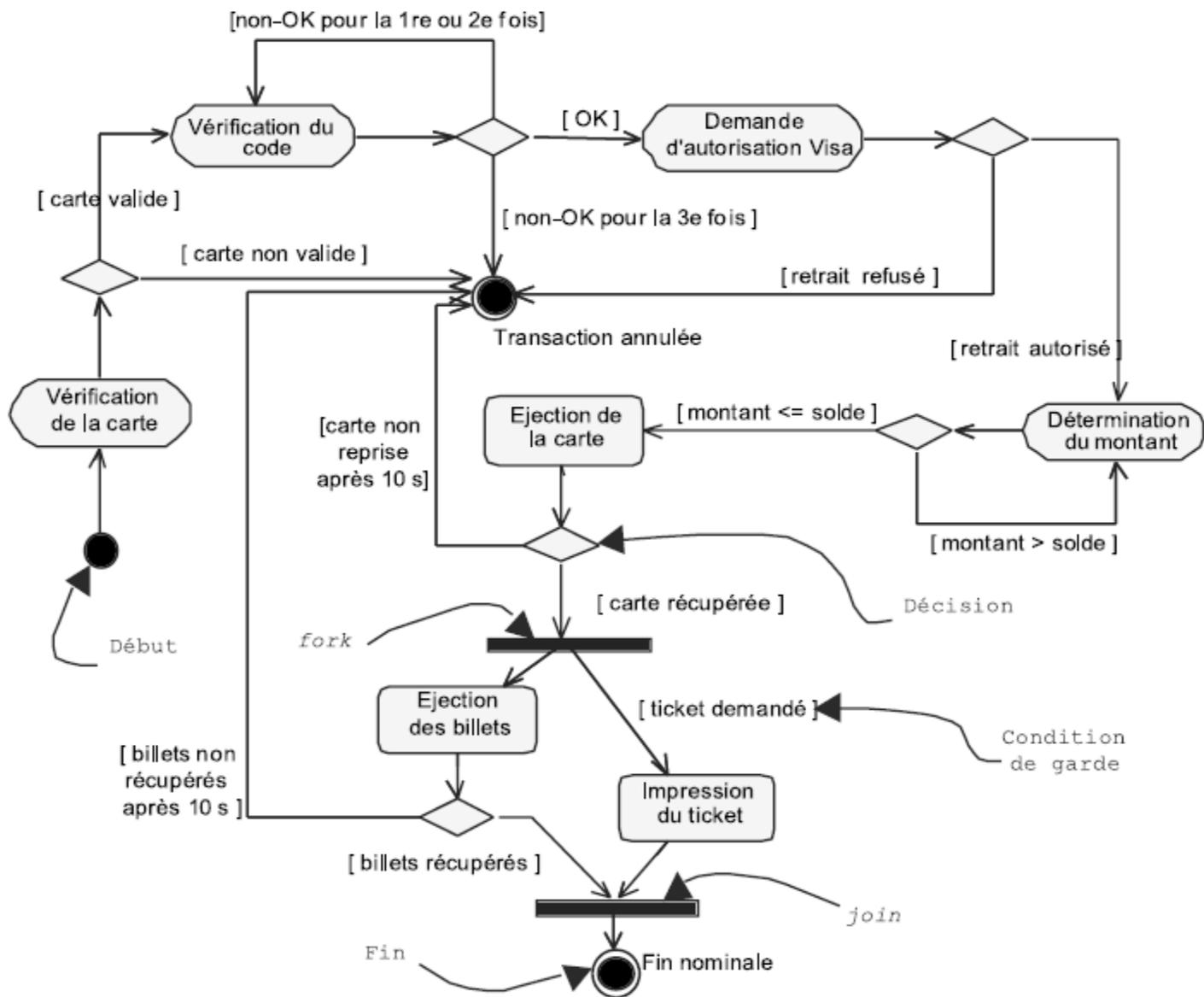
- Pour montrer les différentes responsabilités au sein d'un mécanisme ou d'une organisation, on schématise des couloirs d'activités.
- Chaque activité est allouée à un couloir correspondant à la ressource concernée : partenaire, travailleur



TD Diagramme d'Activités

- DISTRIBUTEUR DE BILLETS

Décrire le fonctionnement d'un distributeur de billets. Le client introduit sa carte dont la validité est immédiatement vérifiée. Il est ensuite invité à saisir le code de la carte. Après trois tentatives infructueuses, la carte est avalée. Sinon le client peut indiquer le montant qu'il désire retirer, le solde de son compte bancaire est alors consulté pour s'assurer que le retrait est possible. En cas de solde insuffisant, le client en est informé et peut alors saisir un montant inférieur. Si le solde du compte est suffisant, le distributeur restitue la carte et délivre alors les billets accompagnés d'un reçu.



A retenir

- Contrairement au diagramme de séquence précédent qui ne décrit que le scénario nominal, le diagramme d'activité doit représenter l'ensemble des actions réalisées par le système, avec tous les branchements conditionnels et toutes les boucles possibles.
- C'est un graphe orienté d'actions et de transitions. Les transitions sont franchies lors de la fin des actions ; des étapes peuvent être réalisées en parallèle ou en séquence.

Architecture « physique »

Architecture *physique*...

- L '*architecture physique* du système s'intéresse à la description détaillée de celui-ci sur les plans du *hardware* et du *software*.
- L'architecture physique répond aux questions suivantes:
 - Dans quels *programmes* ou *processus* les *classes* et *objets* du modèle logique *résident-ils physiquement*?
 - Sur quel(s) *processeur*(s) ces *programmes* et *processus* s'*exécutent-ils*?
 - Quels *types d'ordinateurs* composent le système et comment sont-ils reliés *physiquement*?
 - Quelles sont les *dépendances* entre les différents *fichiers* ou *packages* de classes (lors d'un changement de fichier, quels autres fichiers doivent être recompilés)?

Diagrammes de l'architecture physique

- En **UML**, l'architecture physique est principalement décrite par **deux diagrammes**:
 - le **diagramme des composants** (« **component diagram** »): contient les **composantes logicielles** du projet (unités de code et fichiers concrets-binaires et sources).
 - le **diagramme de déploiement** (« **deployment diagram** »): décrit l'architecture physique du système en couvrant les **dispositifs physiques** (**ordinateurs, processeurs**, etc.) et le **logiciel** qui leur est respectivement associé.

Le *hardware*

- La partie *hardware* d'un système est divisée en *trois* éléments:
 - Les *processeurs*: ce sont les ordinateurs qui exécutent les programmes du système.
 - Les «*devices*»: ce sont les périphériques de support tels les imprimantes, routeurs, lecteurs de disquettes/CD, etc. Ils sont généralement connectés à un processeur qui les contrôle. Il y a souvent peu de différence entre un processeur et un device.
 - Les *connexions*: ce sont les liens physiques entre les processeurs (câbles-fibre optique et/ou protocoles p.e. TCP/IP)

Le *software*...

- UML définit 5 stéréotypes aux composants :
 - « document »: un document quelconque
 - « exécutable »: un programme qui peut s'exécuter
 - « fichier »: un document contenant un code source ou des données
 - « bibliothèque »: une bibliothèque statique ou dynamique
 - « table »: une table de base de données relationnelle

Le diagramme de ***composants***
(« ***component diagram*** »)

Le diagramme de composants

- Il décrit les **composants logiciels** et leur **interdépendance**.
- Il représente par conséquent la **structure** du **code**.
- Les **composantes** sont les **implantations physiques** des **concepts** et **fonctionnalités** définis dans le **modèle logique** du système (i.e., les **classes**, les **objets**, les **liens** et les **collaborations**).
- Les **composantes** sont typiquement les **fichiers d'implantation** des éléments logiques du système.

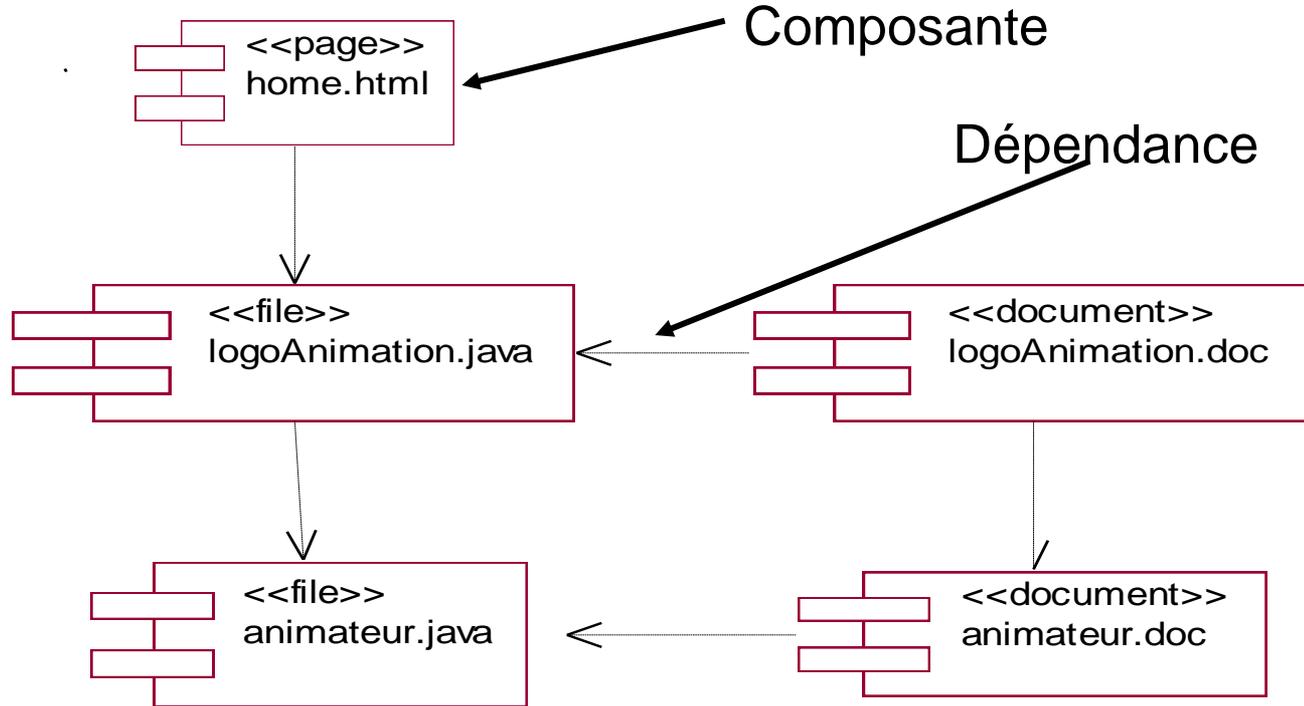
Les composantes...

- En **UML**, un **composant** est représenté par un **grand rectangle** avec **deux petits rectangles** placés à la gauche du grand rectangle.
- Le **nom** du composant est inscrit en dessous ou à l'intérieur du grand rectangle.
- Les **composants** sont des **types** et seules les composantes **exécutables** peuvent être **instanciées**.

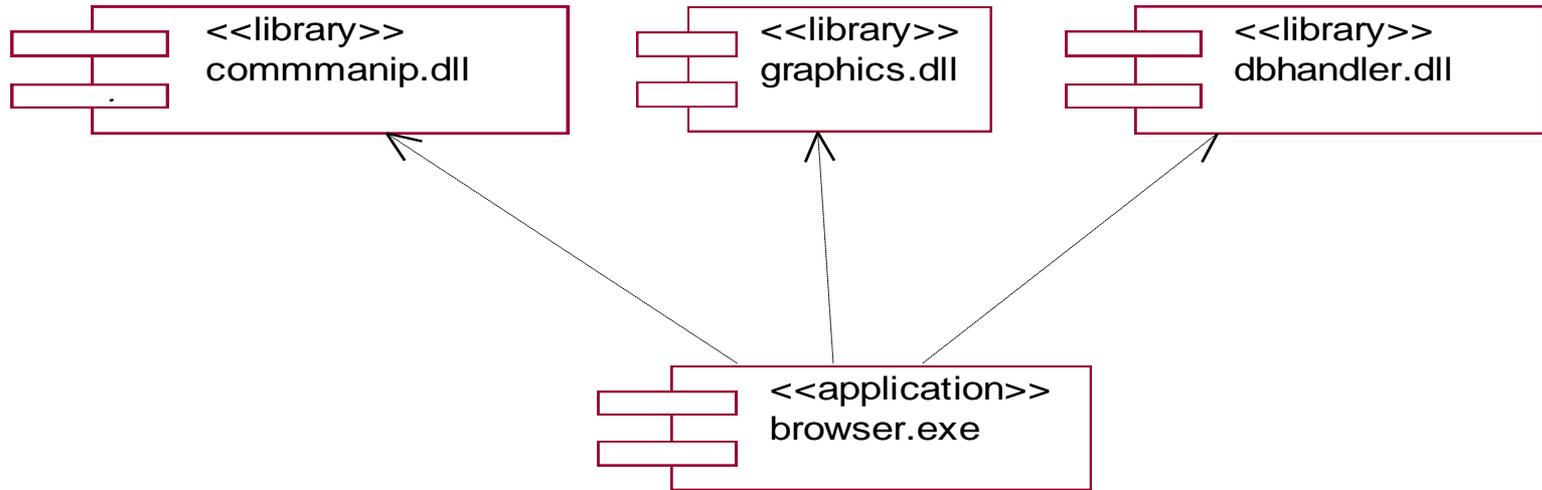
Les composantes...

- Les **dépendances** entre composants sont illustrées par une **ligne pointillée** avec une **flèche simple**.
- Une **dépendance** signifie qu'un composant a besoin d'une autre pour que sa **définition** soit **complète**.
 - Dans un langage **compilé**, si un module A dépend d'un module B, cela signifie que module A doit être recompilé si des changements sont apportés à B.
 - Si les composantes sont **exécutables**, les dépendances montrent les bibliothèques dynamiques qui sont nécessaires lors de l'exécution d'un programme.

Exemple de diagramme de **composantes de code source**



Exemple de diagramme de **composantes run-time**



Le diagramme de **déploiement**
(« **deployment diagram** »)



Le diagramme de déploiement

- Les diagrammes de déploiement montrent la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels.
- Il est la **description finale** de la **topologie du système** car il unit les facettes **hardware** et **software** du système.
- Dans ce type de diagramme, il devrait être possible d'observer un **nœud** de la topologie et de voir les **composantes** qui s'y exécutent, et quelles **structures logiques** sont implantées dans ces composantes.

Les diagrammes de déploiement

Les nœuds (suite):

- Pour montrer qu'un composant réside dans un nœud, deux possibilités:
 - Le symbole du composant est emboîté dans le symbole du nœud.
 - Une dépendance avec le mot clé `<<support>>` orienté du composant vers le nœud.

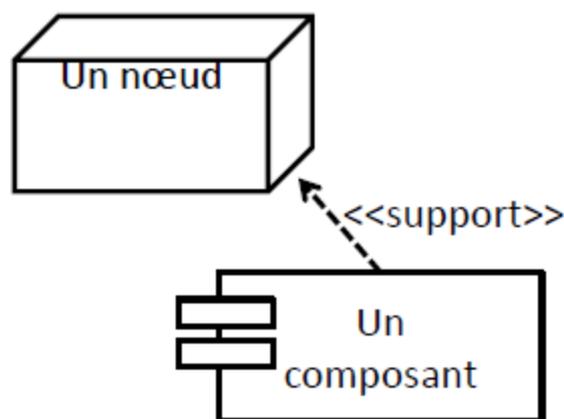
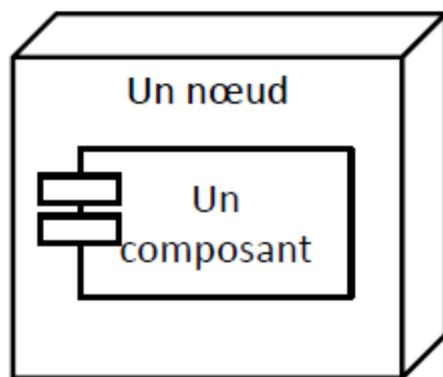


Diagramme de déploiement

- Il comprend:
 - des *nœuds*
 - des *connexions*
 - des *composantes*

Diagramme de déploiement...

- **Nœuds**: unités **physiques** (matérielles) dotées d'une **puissance de calcul** donnée (processeurs, imprimantes, lecteurs de cartes, périphériques de communication, etc).
- Un **nœud** est représenté par un **cube en trois dimensions** avec son **nom** à l'intérieur.
- Les **périphériques** sont aussi représentés par des **nœuds** avec un **stéréotype** ou au moins un nom qui indique clairement que le nœud n'en est pas un de calcul.

Diagramme de déploiement...exemple simple

