

Conception Orientée Objet

3IIR

Partie II

Pr. Khalid SRAIDI

2022/2023

Plan

- 1) DIAGRAMME DE CLASSES; Contraintes.
- 2) DIAGRAMME D'ETATS-TRANSITIONS.
- 3) DIAGRAMME D'ACTIVITES.
- 4) DIAGRAMME DE PAQUETAGES.
- 5) DIAGRAMME DE Composants.
- 6) DIAGRAMME DE DEPLOIEMENT.

Contraintes d'association

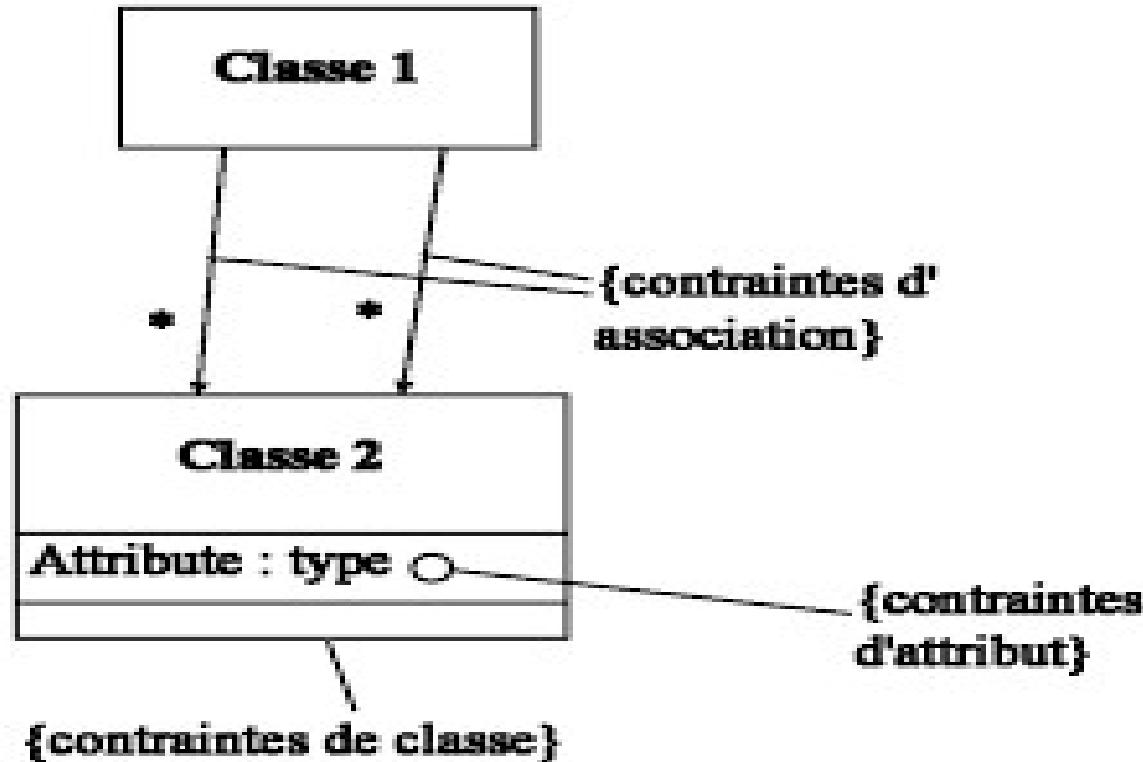
Une contrainte constitue une condition ou une restriction sémantique exprimée sous forme d'instruction dans un langage textuel qui peut être naturel ou formel.

En effet, UML permet d'associer une contrainte à un, ou plusieurs, élément(s) de modèle de différentes façons :

- En plaçant directement la contrainte à côté d'une propriété ou d'une opération dans une classe.
- En ajoutant une note associée à l'élément à contraindre ;
- En plaçant la contrainte à proximité de l'élément à contraindre, comme une extrémité d'association par exemple.
- En plaçant la contrainte sur **une flèche** en pointillés joignant les deux éléments de modèle à contraindre ensemble, la direction de la flèche constituant une information pertinente au sein de la contrainte ;
- En plaçant la contrainte sur **un trait** en pointillés joignant les deux éléments de modèle à contraindre ensemble dans le cas où la contrainte est **bijective**.
- En utilisant une note reliée, par des traits en pointillés, à chacun des éléments de modèle, subissant la contrainte commune, quand cette contrainte s'applique sur plus de deux éléments de modèle.

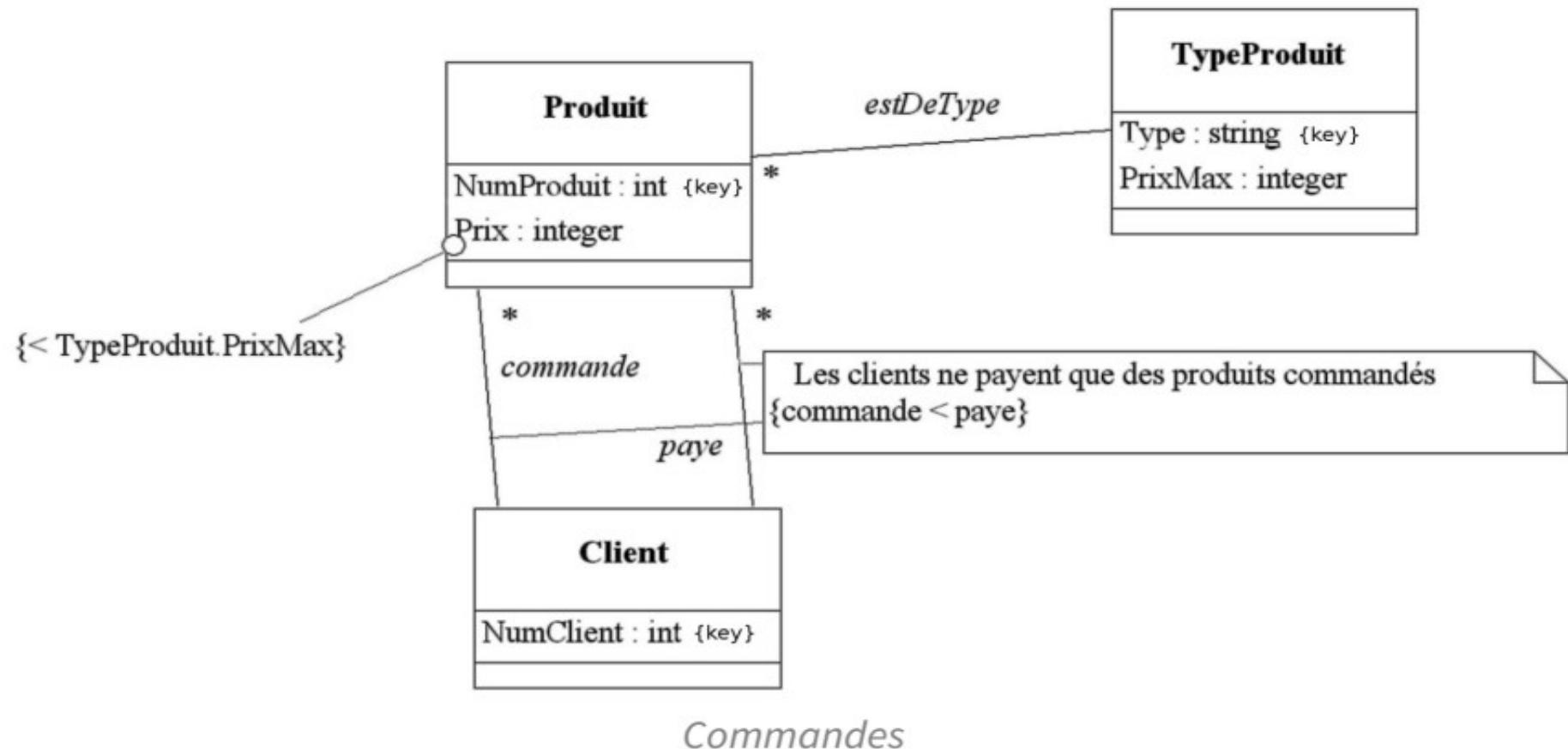
Rappel : toutes les contraintes en UML s'expriment **entre accolades {}**.

Expression des contraintes



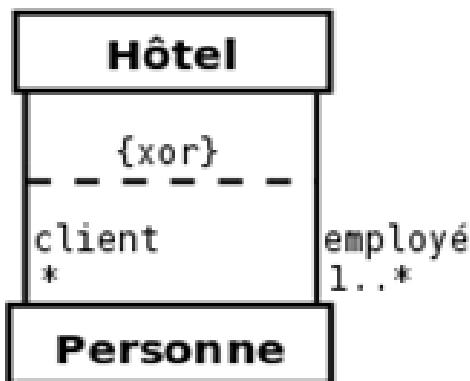
Expression des contraintes

- Exemple:

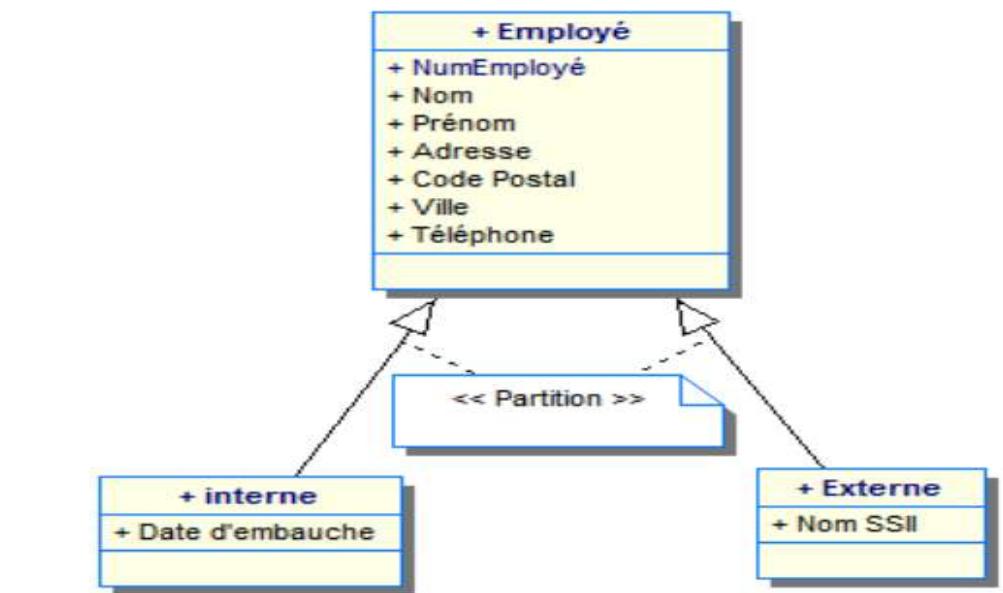


Contraintes d'association

Partition: {XT}, {+}, {xor} ou {ou-exclusif}, {P} : indique qu'une instance ne peut appartenir en même temps à la réalisation de deux associations, Exactement une des deux associations doit être instanciée.



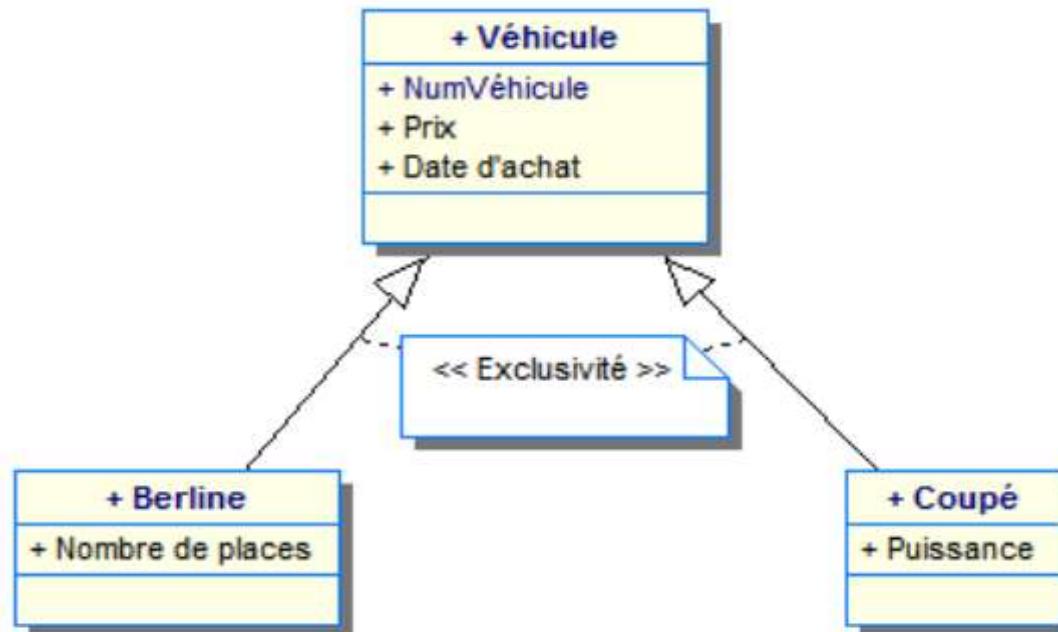
Une personne est soit un employé ou client d'un hôtel et non pas les deux à la fois, il n'existe pas d'autre type d'association.



Un employé est soit un employé interne, soit un employé externe, mais pas les deux, il n'existe pas d'autre type d'employé.

Contraintes d'association

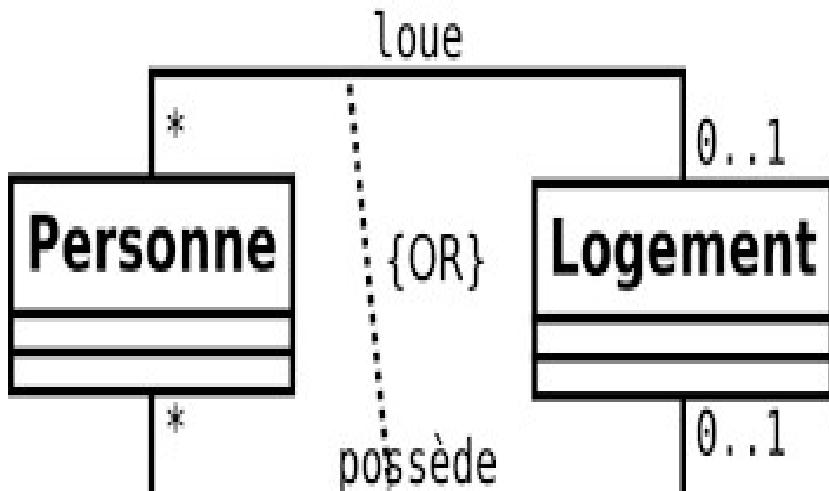
Exclusion {X} Les deux associations ne peuvent être instanciées en même temps.



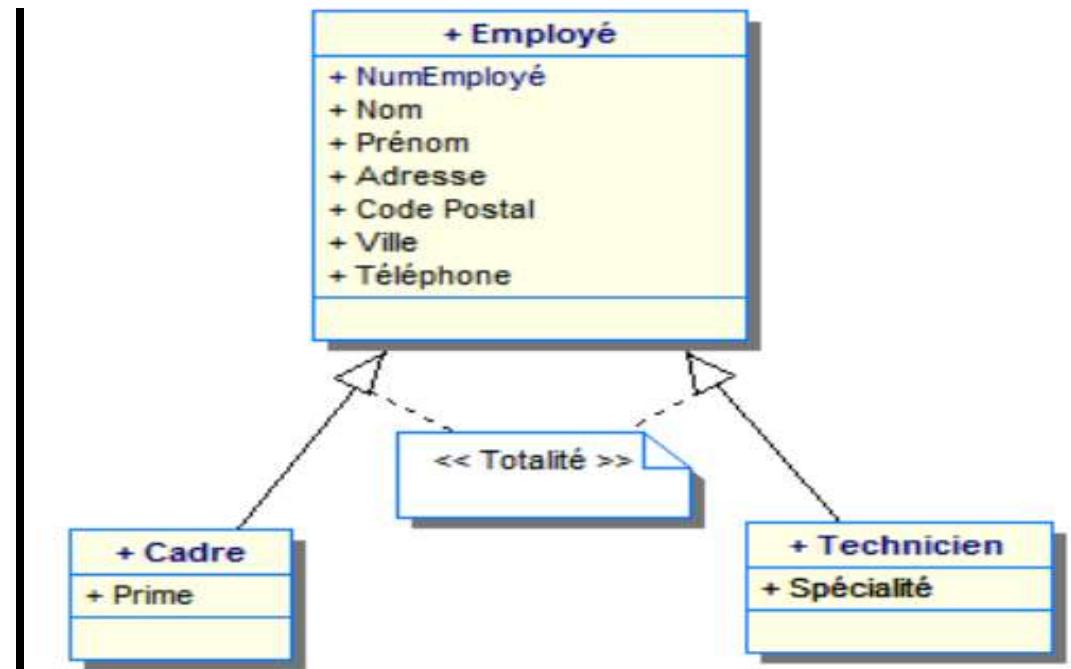
Un véhicule est soit une berline, soit un coupé mais pas les deux et il peut exister d'autres véhicules.

Contraintes d'association

Totalité {T}, également notée {OR} : Au moins une des deux associations doit être instanciée.



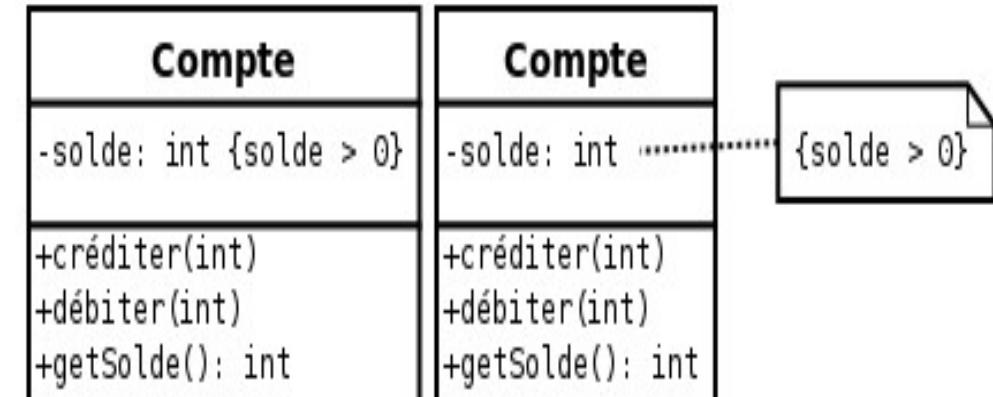
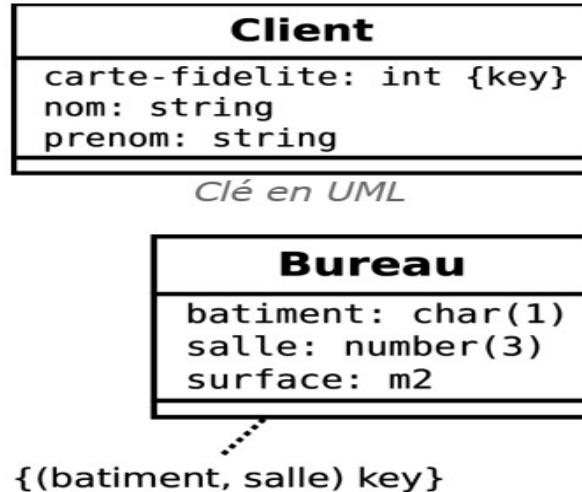
Une personne peut à la fois louer et posséder un logement, mais au moins l'un ou l'autre.



Un employé peut être à la fois cadre et technicien. Il est au moins l'un ou l'autre.

Contraintes sur les attributs

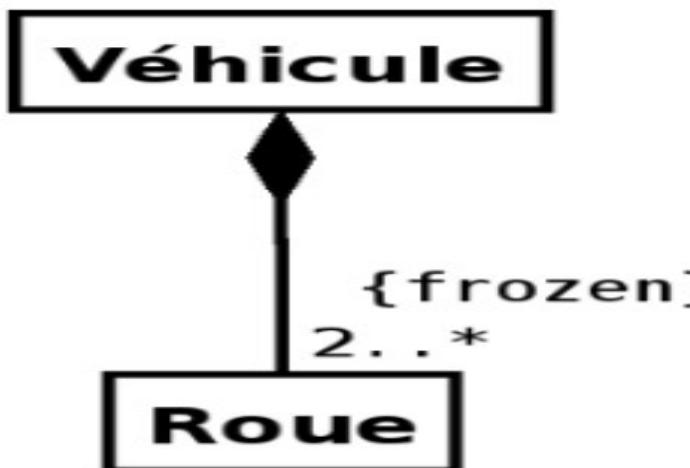
- **{unique}**: La valeur de l'attribut est unique pour la classe.
- **{frozen}**: Une fois instancié l'attribut ne peut plus changer.
- **{key}**: Bien que non standardisé en UML, en base de données, il est courant d'utiliser {key}.



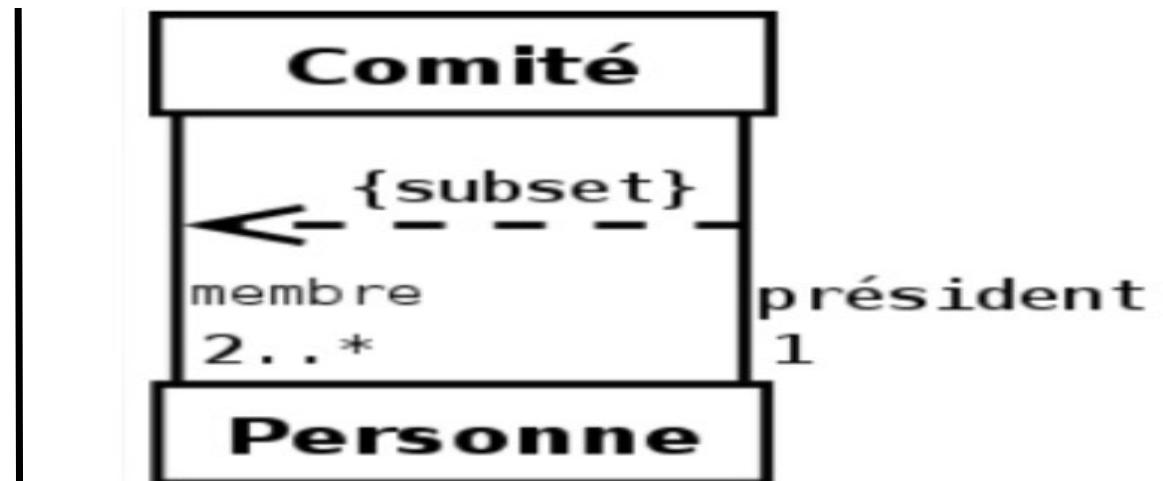
la contrainte porte sur un attribut qui doit être positif.

Contraintes d'association

- **Inclusion:** {subset} /{IN}/{I}: indiquant qu'une occurrence fait partie obligatoirement partie des occurrences d'une autre association ;
- {frozen}/ {immutable} : Un lien ne peut plus être modifié ni détruit après sa création ;



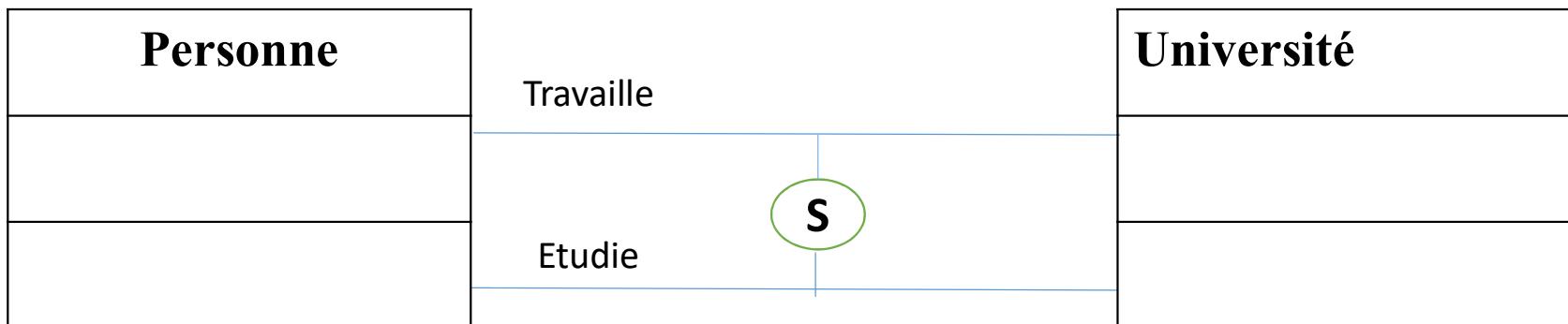
Une fois un véhicule est fabriqué, le nombre des roues est fixé, on ne peut pas ni le changer ni les détruire



un Comité possède au moins 2 personnes (membres), et un président qui fait nécessairement partie du Comité.

Contraintes d'association

- Simultanéité {S}, également notée {=} ou {AND} : Si une association est instanciée, l'autre doit l'être aussi. La simultanéité est équivalente à une double inclusion.



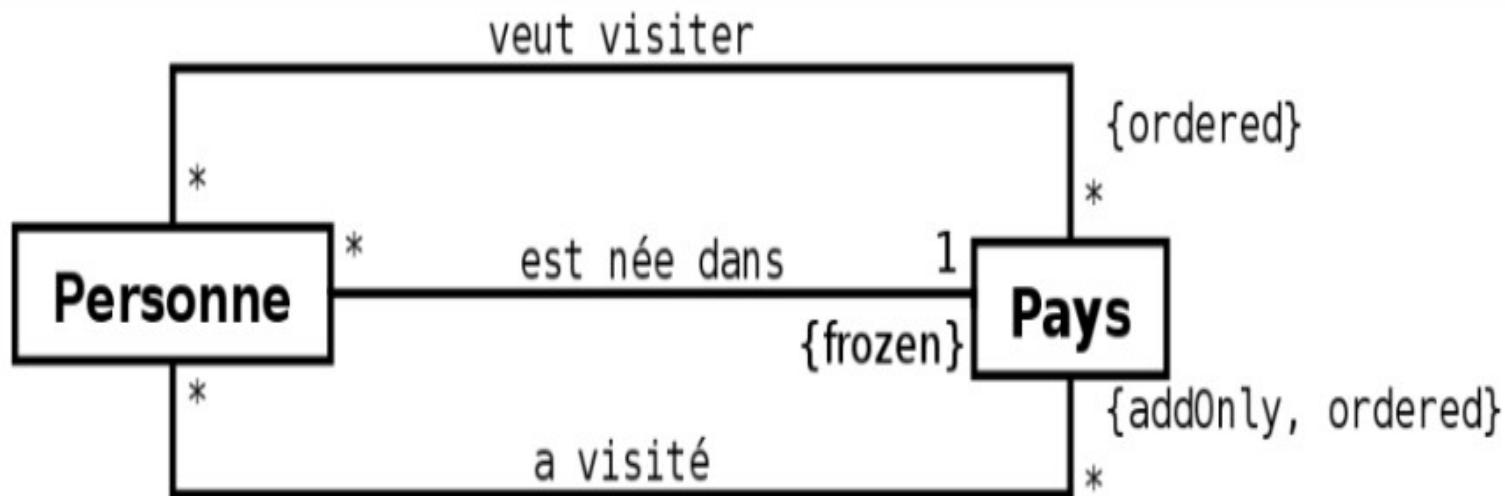
Une personne qui travaille dans une université est obligatoirement un étudiant et vice versa

Contraintes d'association

- {ordered} ou {ordonné} : Les objets doivent être ordonnés.

Rq : on ne sait pas *comment* cela sera ordonné (c'est un choix de conception) ;

- {addOnly} : On ne peut qu'ajouter un objet, pas le détruire ;



Une Personne veut visiter ou pas des pays (ordre traduisant sa préférence) ; il est né dans un pays (qui ne changera jamais) ; il a visité un certain nombre de pays (peut-être aucun), on ne peut qu'en ajouter, et ces pays visités sont ordonnés (selon sans doute leur date de visite).

Exercices: Dessiner deigramme de classes de chaque situation ci-dessous

- 1°)** Le nom d'un médicament est unique.
- 2°)** Une personne a un nom, prénom, lieu et date de naissance (date).
- 3°)** Dans une entreprise de transport, les chauffeurs (caractérisés par un N° de permis, un nom et un prénom) se classent en deux catégories : les chauffeurs « Transport International Routier (TIR) et les chauffeurs « transport national » (TN). Sachant qu'un chauffeur « TIR » est aussi habilité à assurer des transports nationaux.
- 4°)** Un client peut être soit un prospect soit un client en portefeuille s'il a déjà passé au moins une commande
- 5°)** Une personne peut être un membre de jury d'une soutenance ou un président, ce dernier est obligatoirement fait partie des membres de jury.
- 6°)** Les types de produit (CodeTypeProduit, LibTypeProduit) sont commercialisés sur un ou plusieurs secteurs (CodeSecteur, NomRégion). Les représentants (NumRep, NomRep) sont responsables d'un ou plusieurs types de produit sur un ou plusieurs secteurs. Un représentant ne peut être responsable que si le produit est commercialisé sur son secteur.
- 7°)** Même question que 4, sauf qu'on considère cette fois qu'un produit commercialisé sur un secteur a obligatoirement un responsable.

DIAGRAMME DE PACKETAGES

1°) Définitions:

- **Un diagramme de paquetages** est un diagramme structurel (statique) d'UML qui représente des **paquetages** (ou espaces de noms) composant un système, ainsi que les relations qui lient ces différents paquetages.
- Un diagramme de paquetage couramment utilisé pour organiser les éléments du système de haut niveau.
- Un diagramme de paquetage peut simplifier les diagrammes de classes ou de cas d'utilisation complexes et les organiser en packages.
- Les classes d'un même package doivent rendre des services de même nature aux utilisateurs.
- Les dépendances entre packages doivent refléter des relations internes au système.

DIAGRAMME DE PACKETAGES

- Un diagramme de paquetages est un assemblage des éléments appelé paquetages.
- Un paquetage est un regroupement de différents éléments d'un système (regroupement de classes, diagrammes, fonctions, interfaces...). Cela permet de clarifier le modèle en l'organisant.
- Il est représenté par un dossier avec son nom à l'intérieur:



DIAGRAMME DE PACKAGES

Exemple:

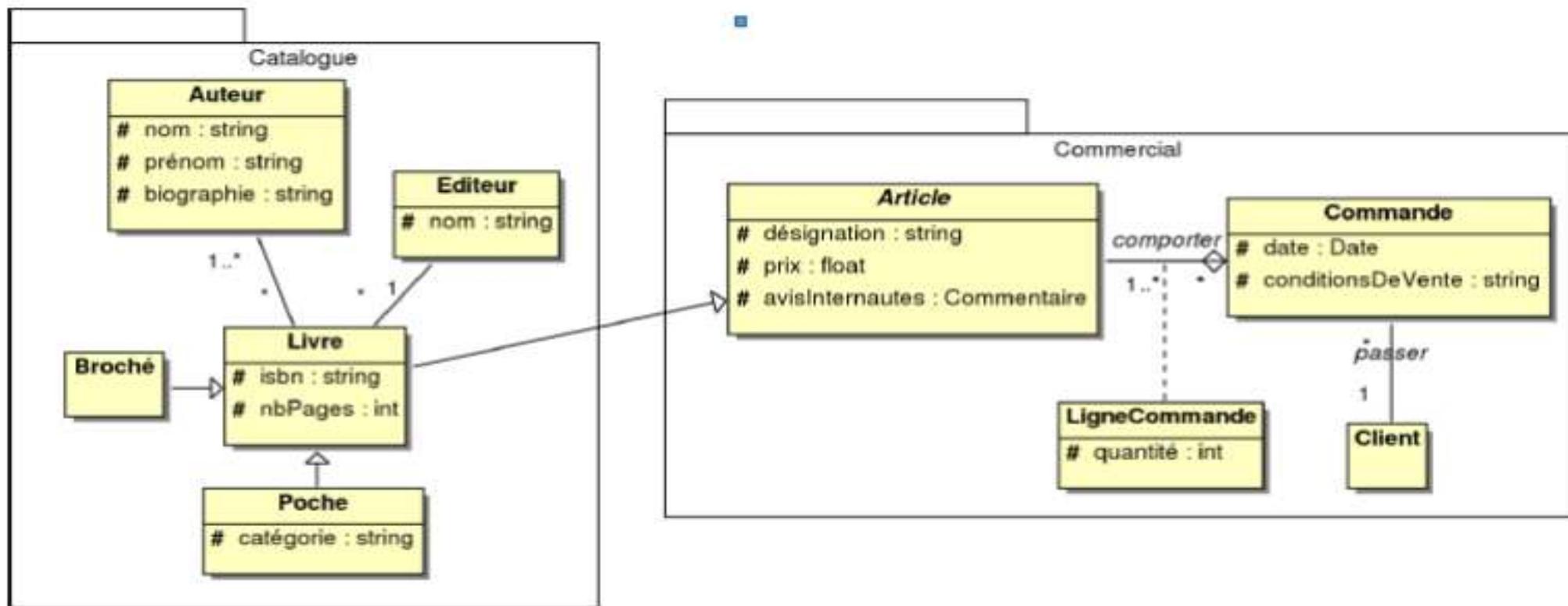


DIAGRAMME DE PACKAGES

Exemple:

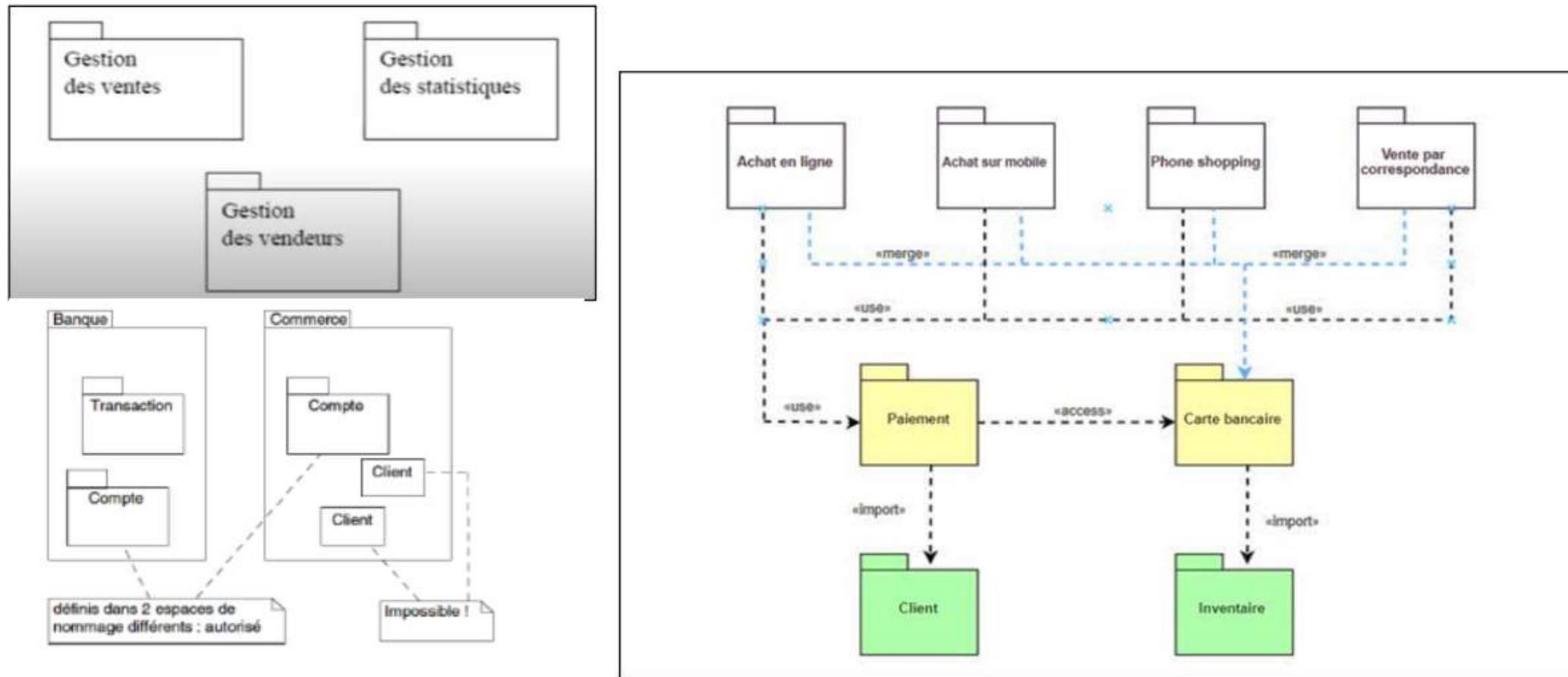
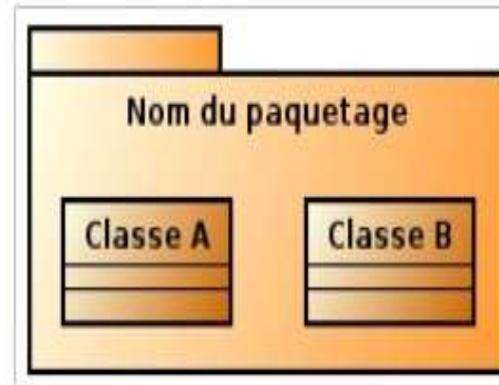


DIAGRAMME DE PACKAGES

- Il est possible de représenter les éléments du système appartenant au paquetage :

à l'intérieur de celui-ci :



ou à l'extérieur :

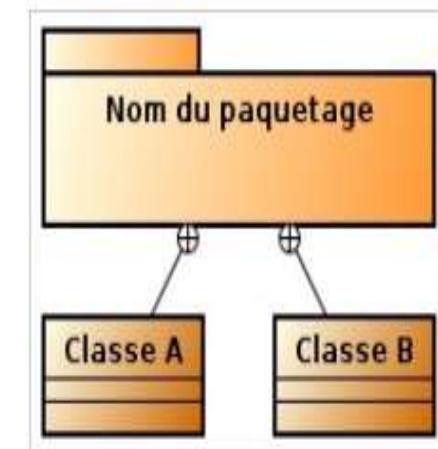
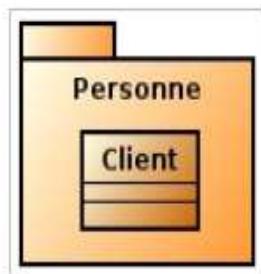
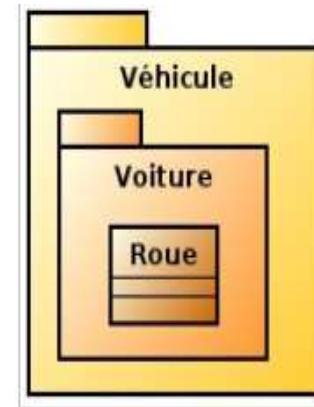


DIAGRAMME DE PACKAGES

- Pour faire appel à un élément d'un paquetage, nous indiquons le nom du paquetage (espace de nommage) suivi de deux fois deux points (::) puis du nom de l'élément:



→ *Personne::Client*

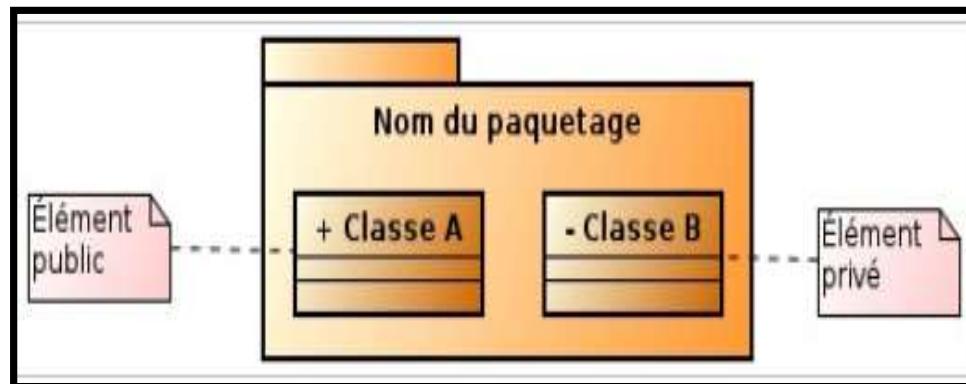


→ *Véhicule::Voiture::Roue*

Dépendances entre paquetages

1) Visibilité : Chaque éléments d'un paquetage est soit :

- **privé**, c'est-à-dire **encapsulé** dans le paquetage et **invisible** à l'extérieur de celui-ci. Un élément privé est désigné par un signe – devant lui.
- **public**, c'est-à-dire **visible** et **accessible** de l'extérieur du paquetage. Un élément public est désigné par un signe + devant lui.



N.B: *Par défaut, les éléments d'un paquetage sont publics.*

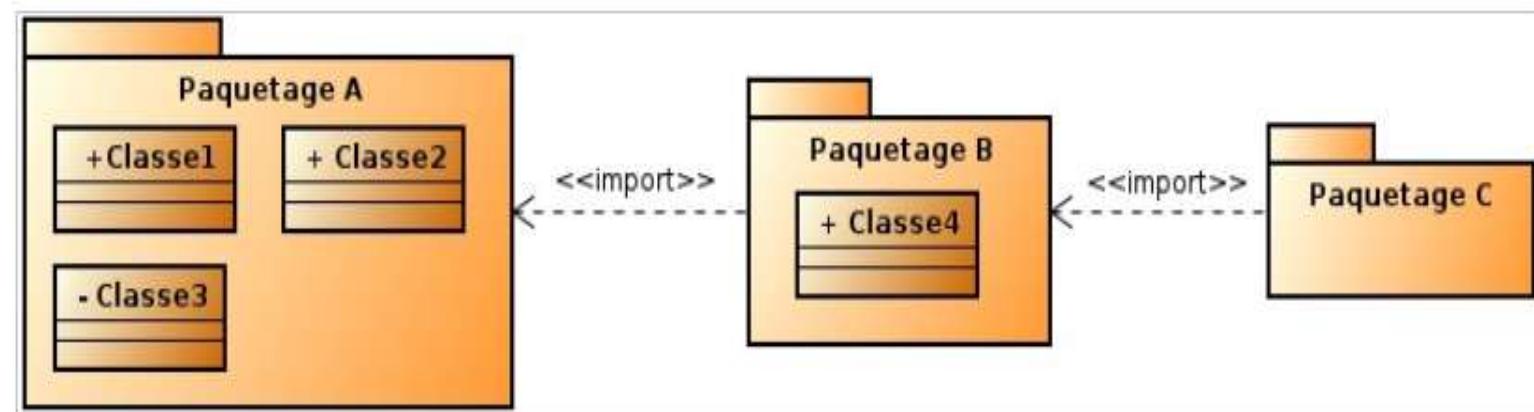
Dépendances entre paquetages

2) *import* : Elle correspond à l'importation par un paquetage **B** de tous les éléments publics d'un paquetage **A**.

Ces éléments :

- auront la visibilité « **public** » dans le paquetage **B** (et seraient donc aussi transmis à un paquetage **C** qui ferait une importation du paquetage **B**).
-Seront accessibles au paquetage **B** sans avoir à utiliser explicitement le nom du paquetage **A**

Représentation:

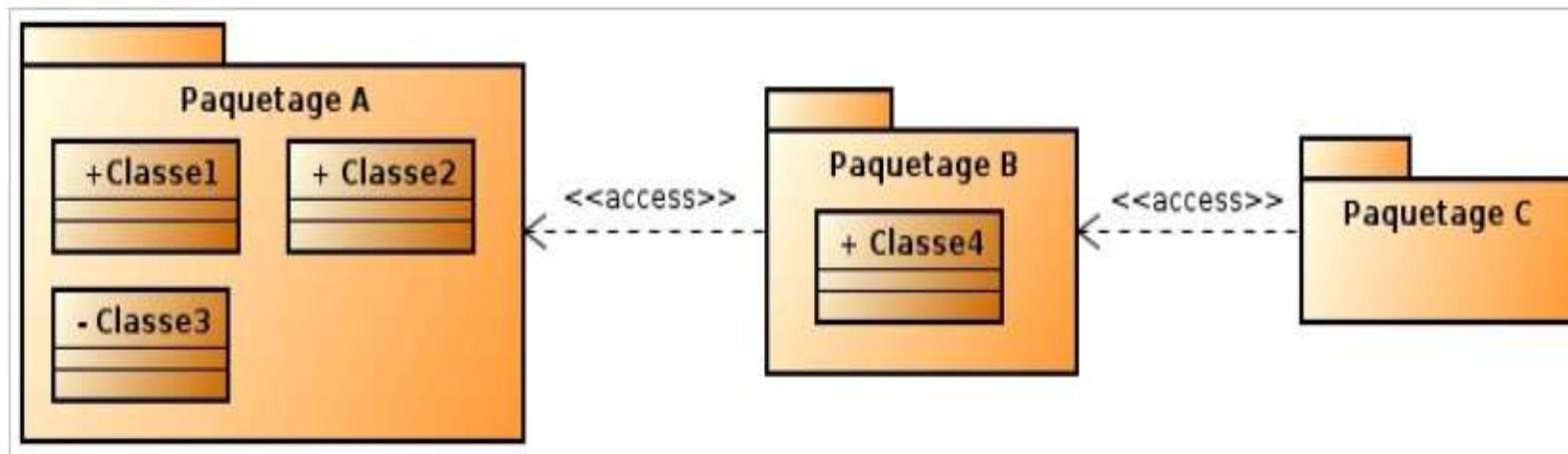


Le paquetage B importe ***Classe1*** et ***Classe2*** (pas ***Classe3*** qui a une visibilité de type ***privée***). ***Classe1*** et ***Classe2*** ont une visibilité de type ***public*** dans paquetage B. Le paquetage C importe ***Classe1***, ***Classe2*** et ***Classe4***.

Dépendances entre paquetages

3) **access** : Elle correspond à l'accès par un paquetage B de tous les éléments publics d'un paquetage A.

Ces éléments : auront la visibilité **privé** dans le paquetage B, ils ne peuvent donc pas être transmis à un paquetage C qui ferait une importation ou un accès au paquetage B.

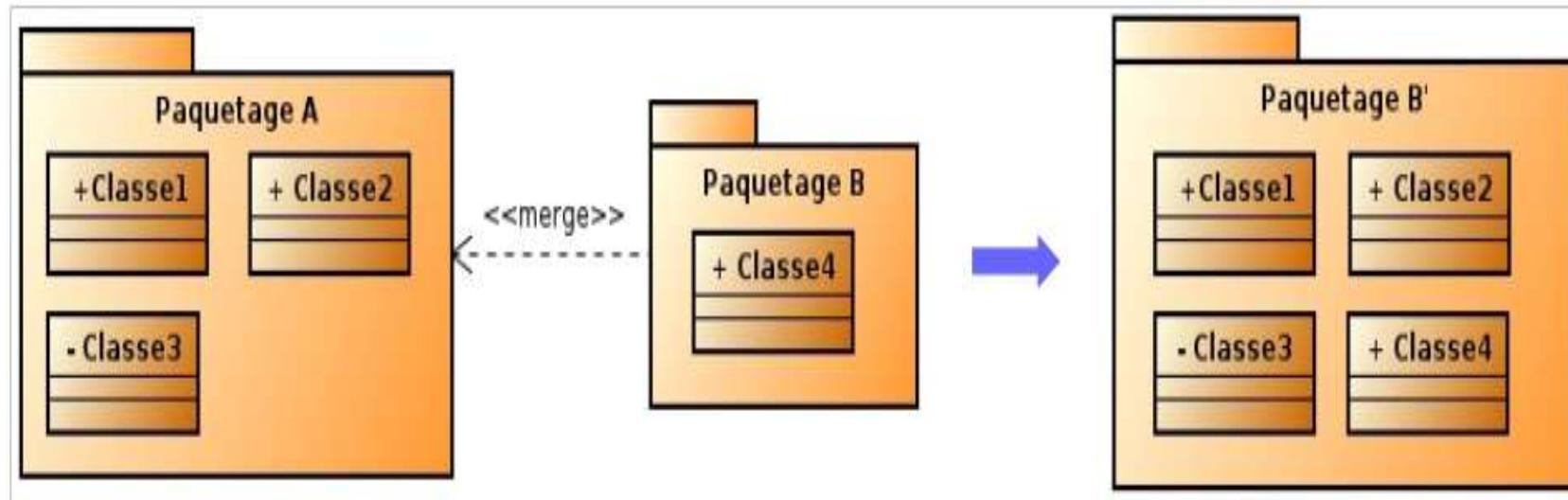


Le paquetage B a accès à **Classe1** et **Classe2** (pas à **Classe3** qui a une visibilité de type **privée**). **Classe1** et **Classe2** ont une visibilité de type **privé** dans paquetage B. Le paquetage C a accès à **Classe4** (pas à **Classe1** et **Classe2** qui ont une visibilité de type **privée** dans paquetage B).

Dépendances entre paquetages

4) Dépendances de type « *merge* » : Elle correspond à la fusion de 2 paquetages en un seul.

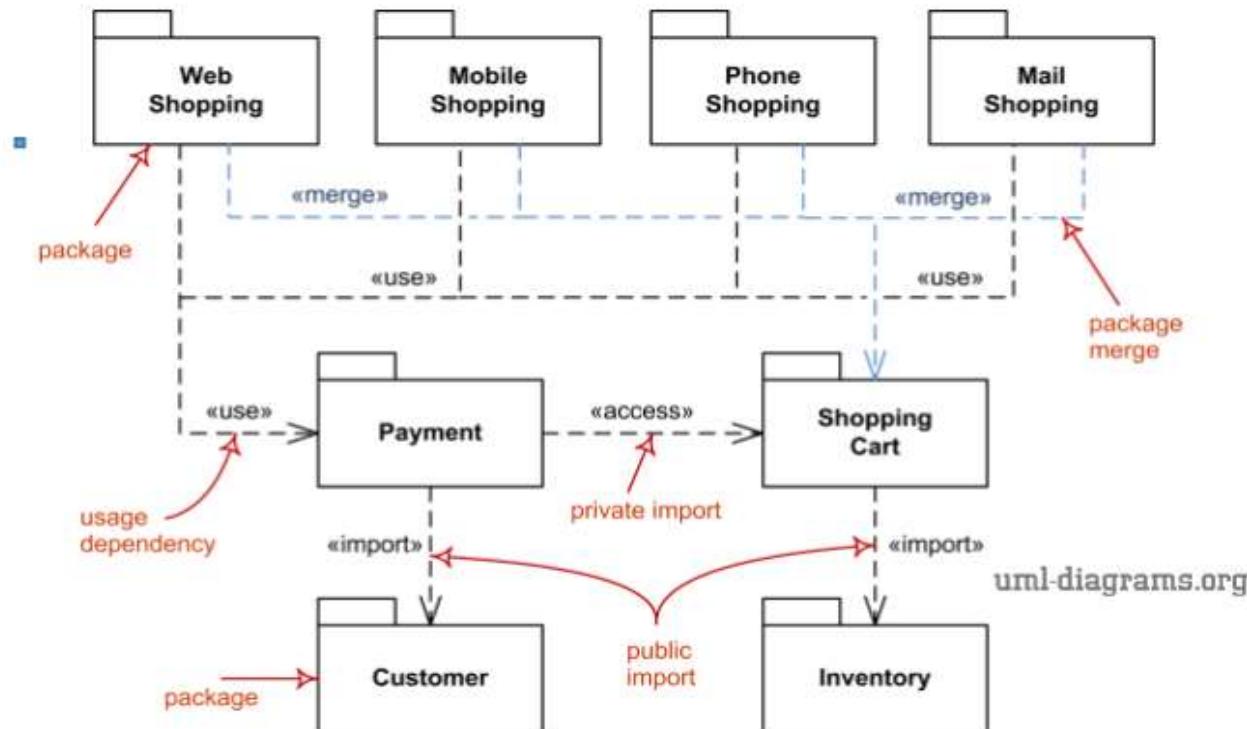
- Le paquetage A est fusionné dans le paquetage B.



-le paquetage A n'est pas modifié alors que le paquetage B est écrasé pour accueillir la fusion des 2 paquetages

Dépendances entre paquetages

Exemple



- Les packages Web Shopping, Mobile Shopping, Phone Shopping, et Mail Shopping merge Shopping Cart package.
- Les 4 packages use Payment package.
- Les deux packages Payment et ShoppingCart import les autres packages.

DIAGRAMME DE PACKETAGES

Exercice d'application:

Proposez un diagramme de paquetage pour deuxième étude de cas (Gestion des vols)de la série 3

DIAGRAMME DE STRUCTURE COMPOSITE

Un diagramme de structure composite est un diagramme structurel UML qui:

- ✓ Le diagramme de structure composite a pour premier objectif de décrire précisément un objet composé.
- ✓ Un tel diagramme n'a pas vocation à remplacer le diagramme des classes mais à le compléter.
- ✓ Permet de visualiser un classificateur structuré donné, en définissant ses classes de configuration, ses interfaces, ses packages et les liens qui les unissent dans le détail.
- ✓ Permet la décomposition de la structure interne de plusieurs classes, interfaces ou composants, et leurs interactions.

DIAGRAMME DE STRUCTURE COMPOSITE

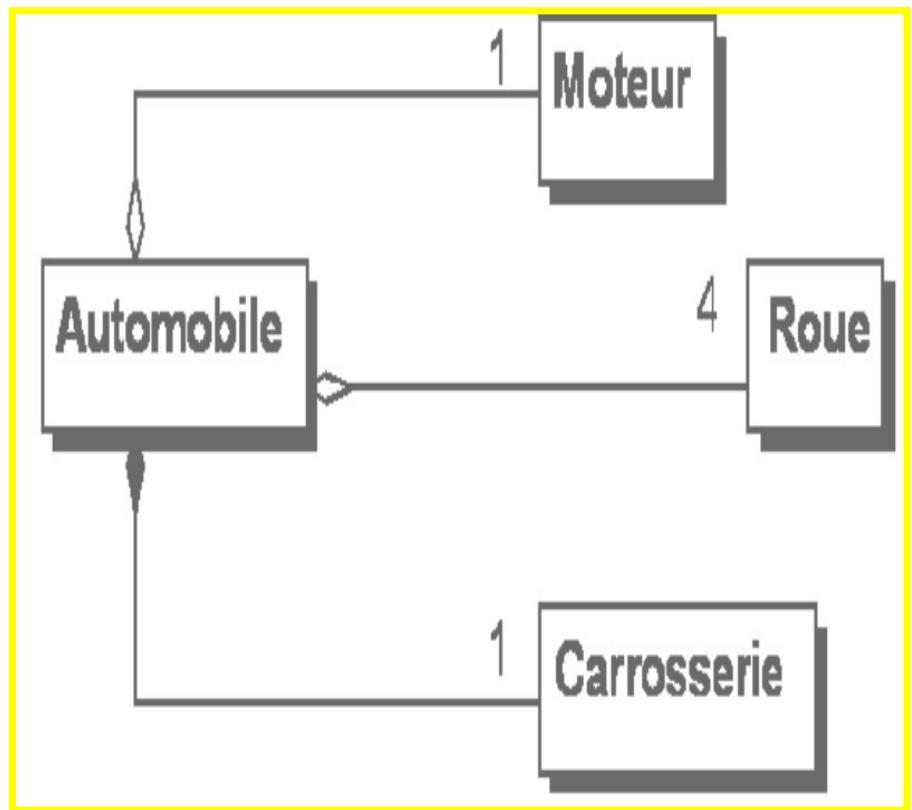


Diagramme de Classe

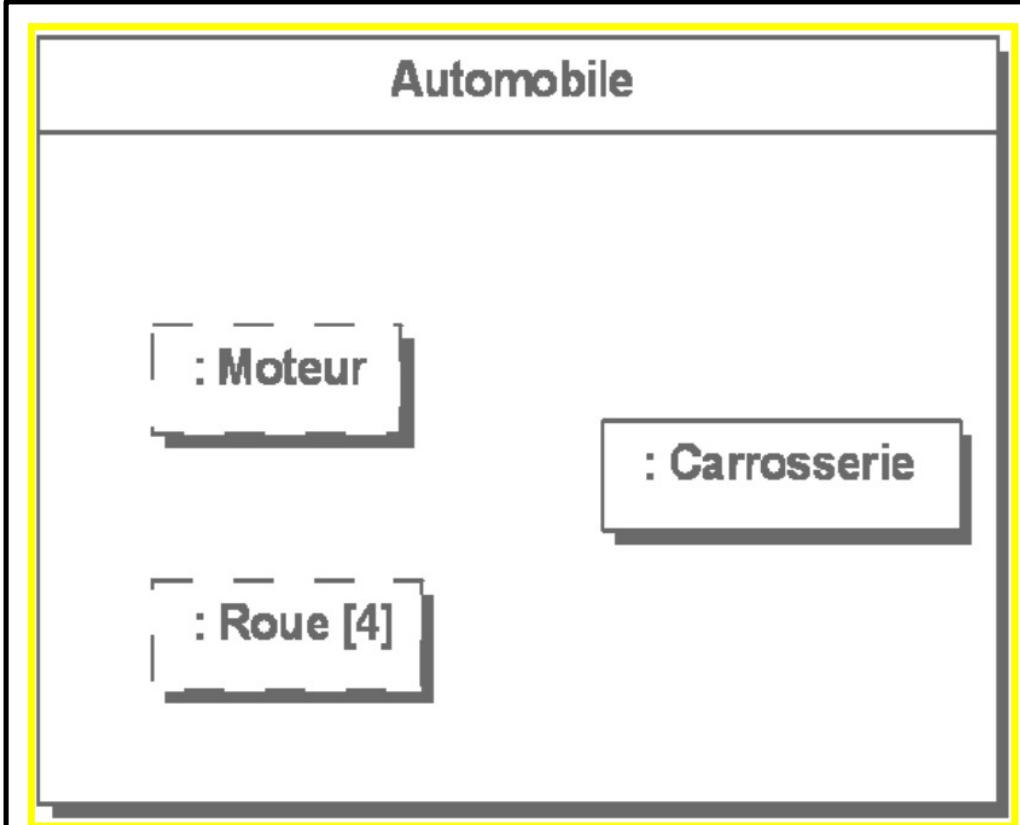


Diagramme de Structure Composite

DIAGRAMME DE STRUCTURE COMPOSITE

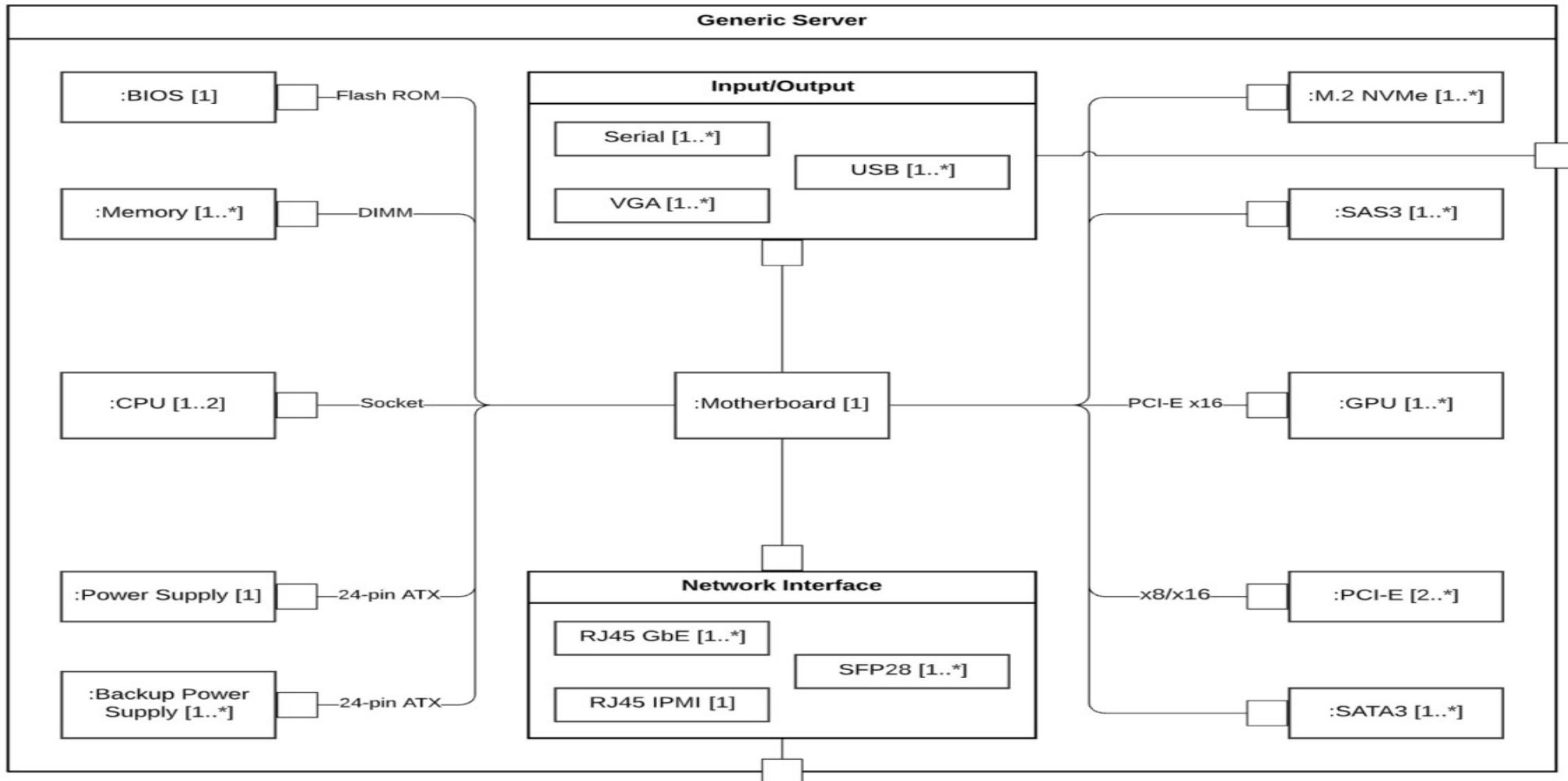


DIAGRAMME DE STRUCTURE COMPOSITE

Composants de base d'un diagramme de structure composite:

1. Classificateur structuré :
 - Classe, souvent abstraite, dont le comportement peut-être déterminé suivant ses interactions avec les différentes parties,
2. Partie :
 - Rôle joué à l'exécution par une instance ou une collection d'instances de classes,
3. Port :
 - Point d'interaction utilisé pour connecter les classificateurs structurés avec leurs parties et leur environnement.
4. Connecteur :
 - Lien entre au minimum 2 entités,
5. Collaboration :
 - Structure purement abstraite,
 - Représentée comme un ovale pointillé contenant le rôle que l'instance peut jouer dans la collaboration.

DIAGRAMME DE STRUCTURE COMPOSITE

Composants de base d'un diagramme de structure composite:

Symbol	Nom	Description
	Partie	Agit comme une instance d'exécution des classes ou des interfaces
	Port	Agit comme un point d'interaction entre une instance de classifieur (ou son comportement) et son environnement
	Interface	Définit le comportement que l'exécutant accepte de respecter
—	Raccord	Représente les liens de communication entre les parties

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

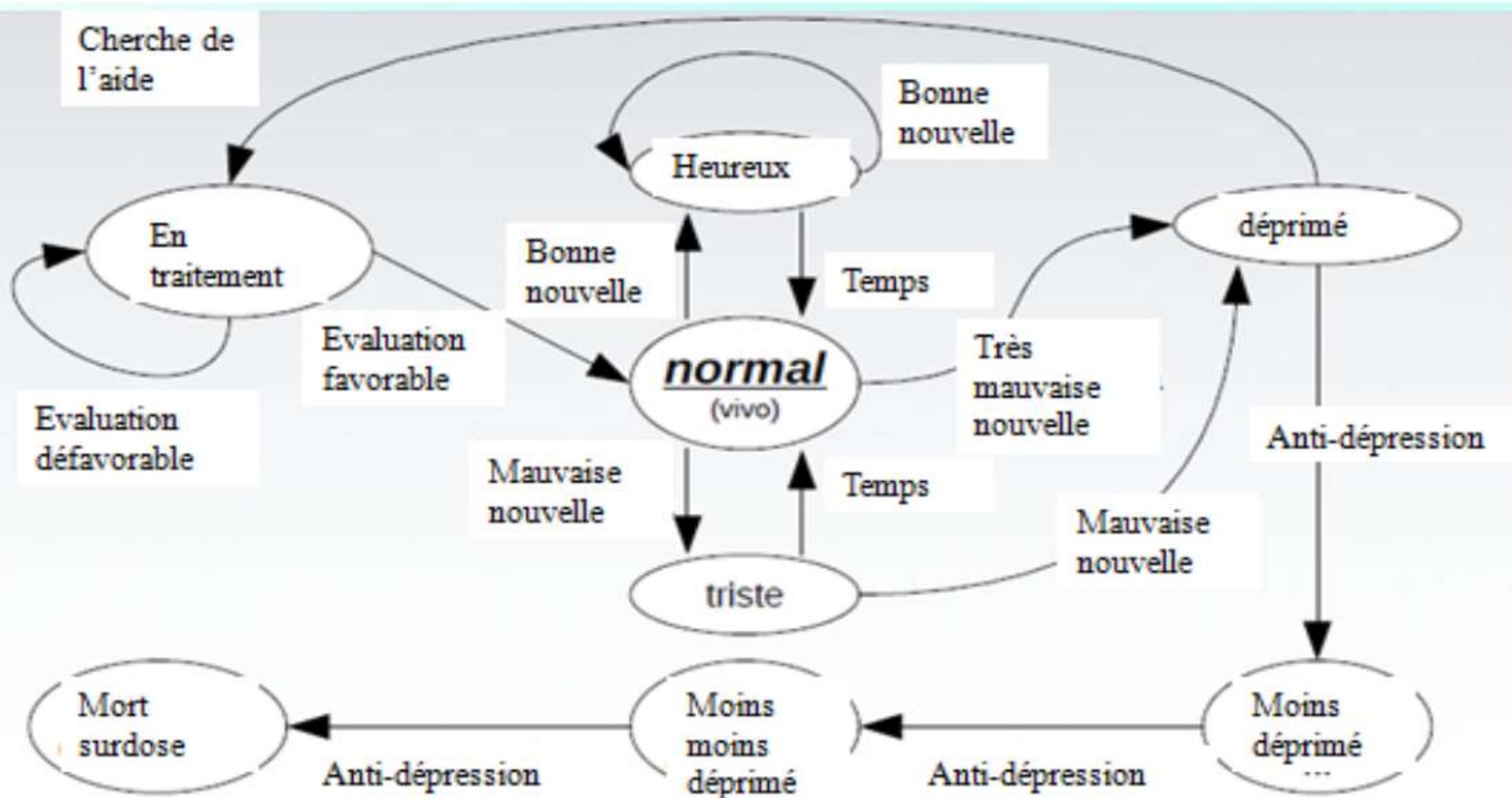


DIAGRAMME D'ETATS-TRANSITIONS

Définitions:

- Il représente les différents états (situations) dans lesquels peut se trouver l'objet ainsi que la façon dont cet l'objet passe d'un état à l'autre en réponse à des événements.
- 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 orienté composé des :
 - 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'ETATS-TRANSITIONS

Composants

- Trois composants :
 - Les états
 - Les transitions
 - Les événements
- Ces 3 composants sont indissociables pour décrire le comportement dynamique d'un objet.
 - Un objet est dans un état donné quelconque (état source)
 - Attend un événement
 - La réception d'un événement déclenche une transition
 - Le déclenchement d'une transition permet à l'objet de passer à un autre état (état destination)



L'objet : un élément **passif** contrôlé par les événements en provenance de son environnement

DIAGRAMME D'ETATS-TRANSITIONS

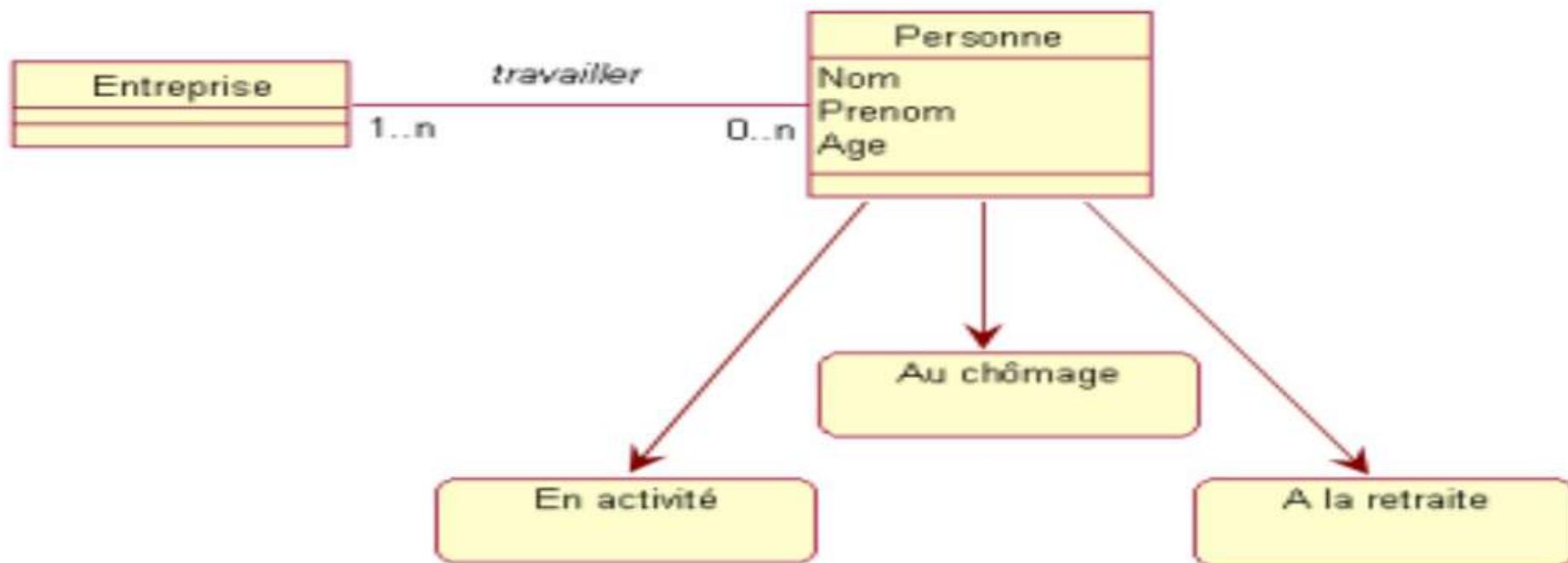
NOTION D'ETAT:

- 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



DIAGRAMME D'ETATS-TRANSITIONS

Exemple : soit le diagramme de classe suivant modélisant les personnes qui travaillent dans des entreprises :



Il y a 3 états qu'une instance de Personne peut prendre:

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

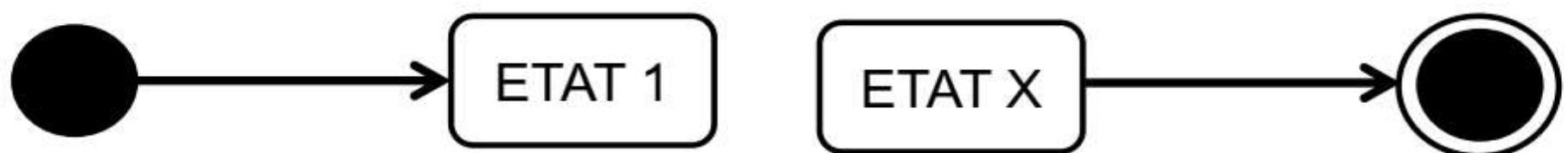


DIAGRAMME D'ETATS-TRANSITIONS

NOTION 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

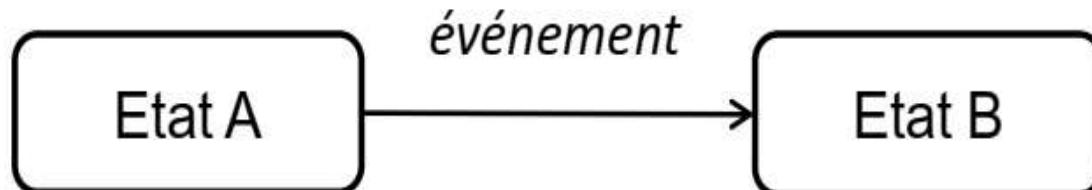


DIAGRAMME D'ETATS-TRANSITIONS

NOTION D'EVENEMENT:

- 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:

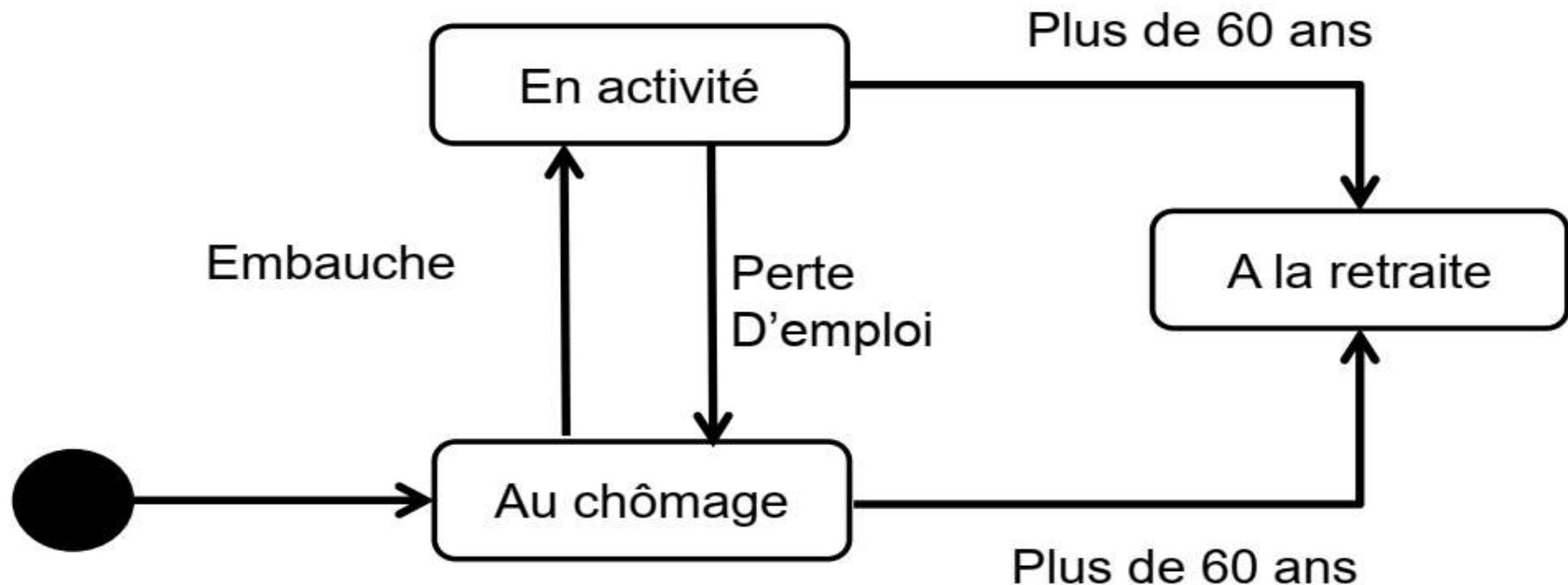
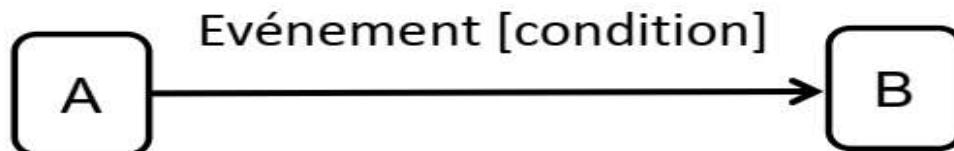


DIAGRAMME D'ETATS-TRANSITIONS

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.



Exemple



Exemple 2

Remarque : Une transition sans nom d'événement est appelée **transition automatique**, elle est déclenchée lorsque l'activité de l'état source est terminée.

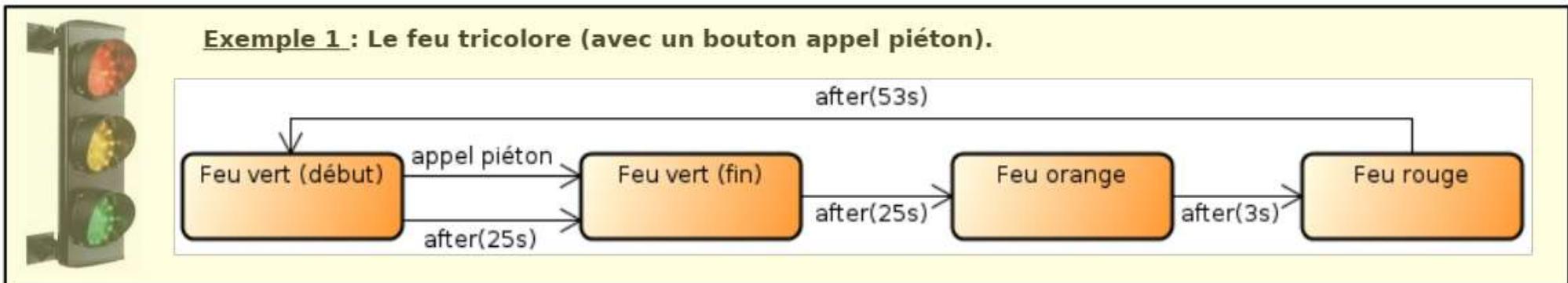


DIAGRAMME D'ETATS-TRANSITIONS

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

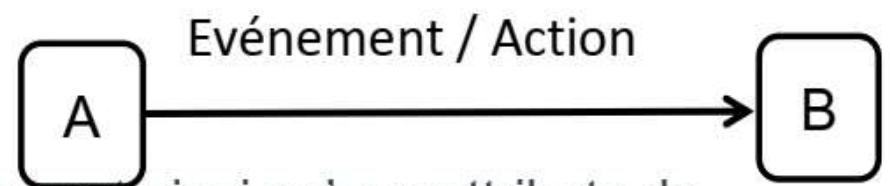
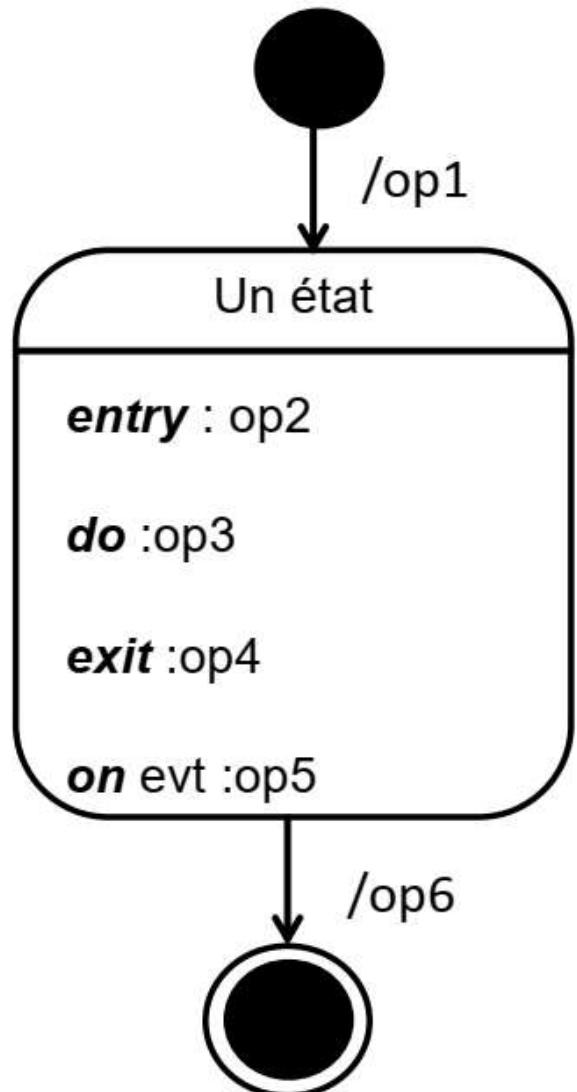


DIAGRAMME D'ETATS-TRANSITIONS



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 **sorite** de l'état (op4)
 - l'action **associée** aux **événements internes** (op5)
 - l'action **associée** à la **transition** de la sortie de l'état (op6)

OPERATIONS, ACTIONS ET ACTIVITES :

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

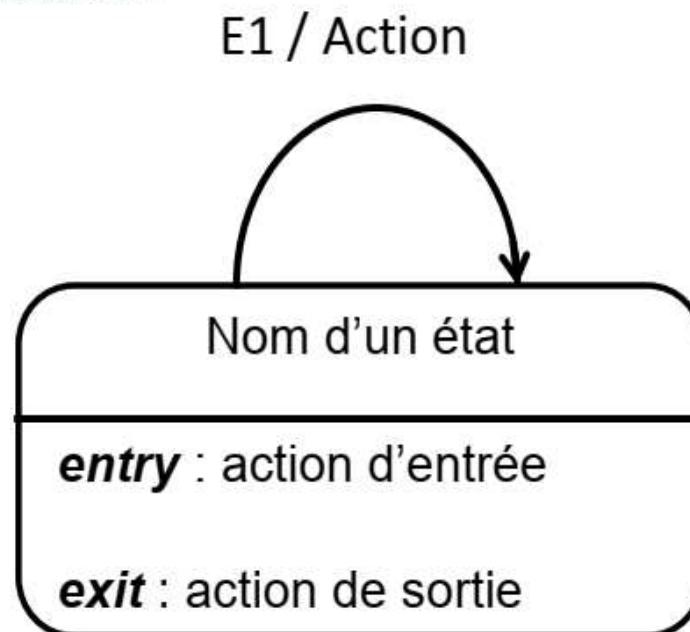


DIAGRAMME D'ETATS-TRANSITIONS

Exemple DAB:

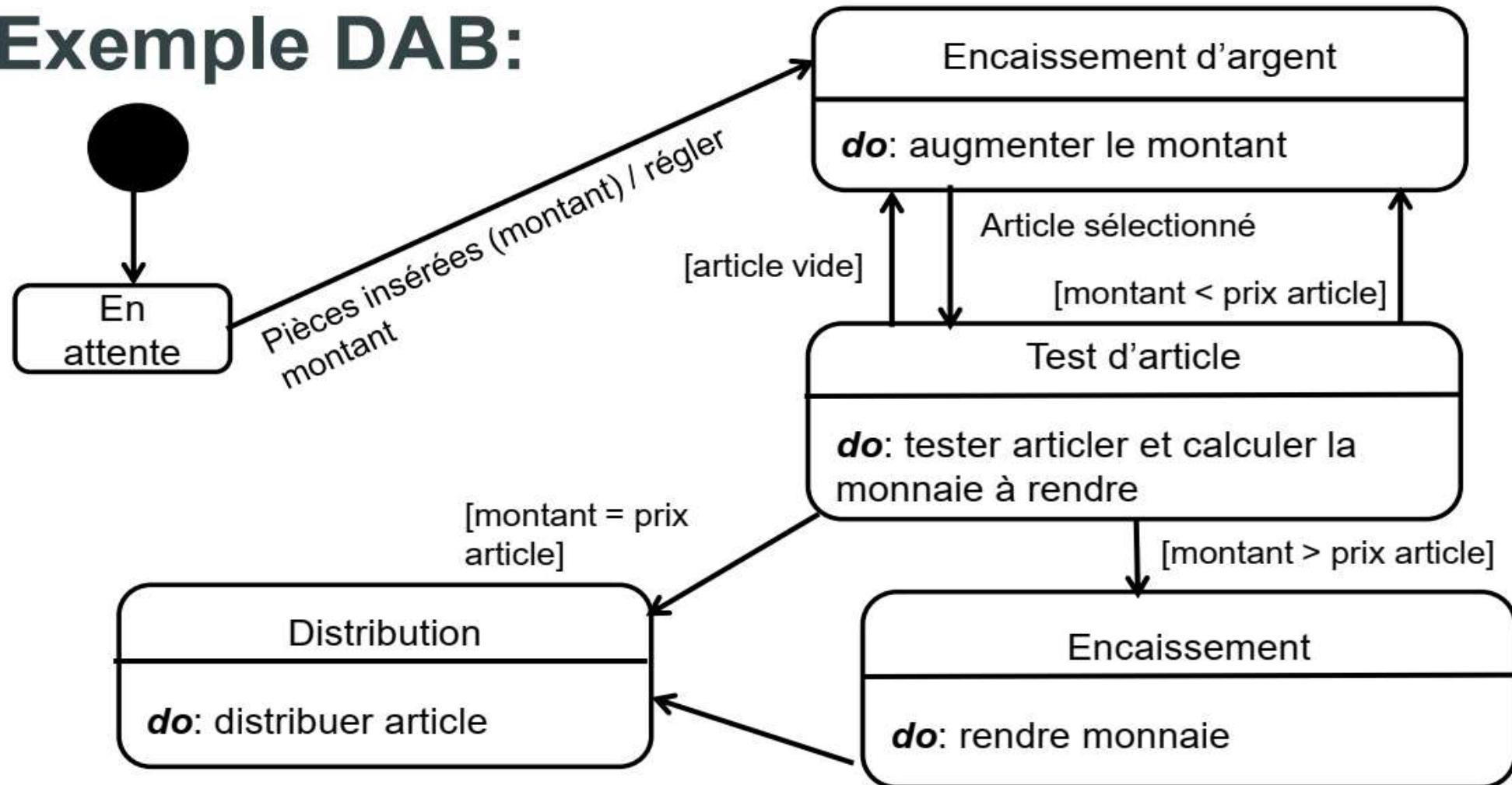


DIAGRAMME D'ETATS-TRANSITIONS

Généralisation d'états:

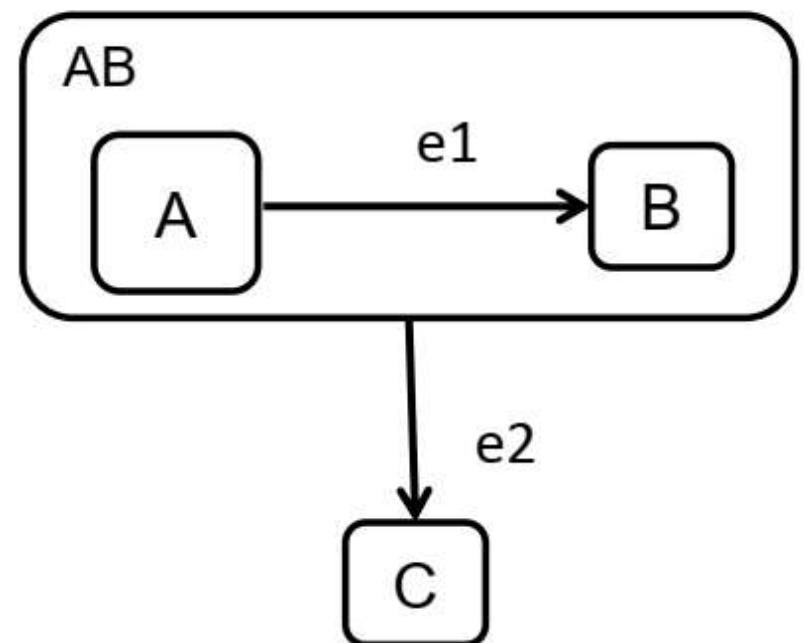
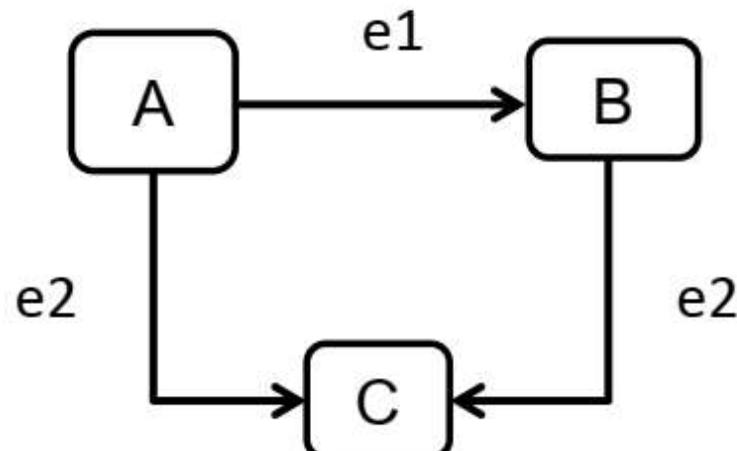


DIAGRAMME D'ETATS-TRANSITIONS

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

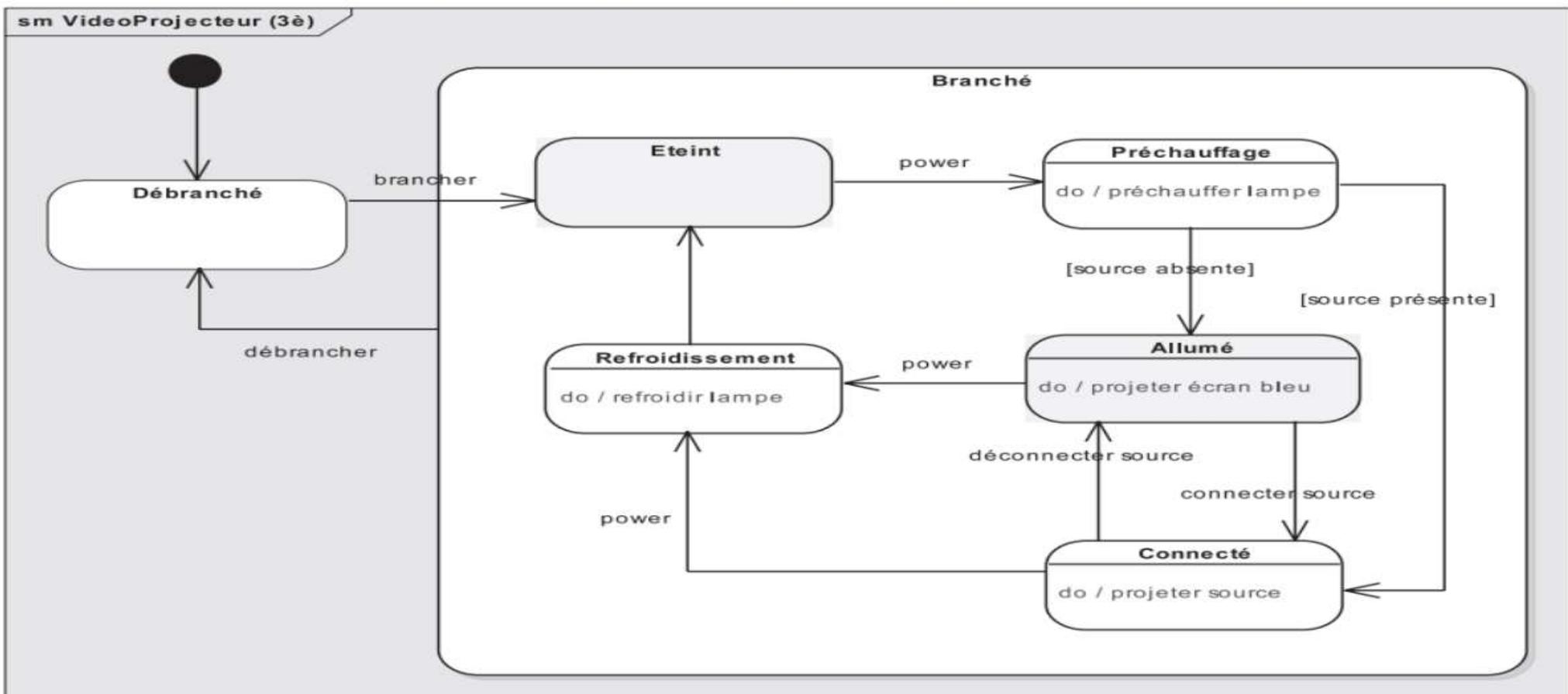


DIAGRAMME D'ETATS-TRANSITIONS

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)

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 → 2 → 3** pour une accélération, et **3 → 2 → 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

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

- *Les diagrammes d'activité affichent le flux de travail d'un point de départ à un point d'arrivée en détaillant les nombreux chemins de décision existant dans la progression des événements contenus dans l'activité*
- *Dans la phase de conception, les diagrammes d'activités sont particulièrement adaptés pour une plus grande description des cas d'utilisation.*
Plus précisément, ils viennent illustrer et consolider la description textuelle.
- 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

DIAGRAMME D'ACTIVITES

Définitions;

Action: Une action est le plus petit traitement qui puisse être exprimé en **UML**. Une action a une incidence sur l'état du système ou en extrait une information.

Une action peut être, par exemple :

- une affectation de valeur à des attributs ;
- un accès à la valeur d'une propriété structurelle (attribut ou terminaison d'association) ;
- la création d'un nouvel objet ou lien ;
- un calcul arithmétique simple ;
- l'émission d'un signal ;
- la réception d'un signal ;
- une mise en attente...

DIAGRAMME D'ACTIVITES

- **Activités (activity)** : définit un comportement décrit par un séquencement organisé d'unités dont les éléments simples sont les actions. Le flot d'exécution est modélisé par des nœuds reliés par des arcs (**transitions**). Le flot de contrôle reste dans l'activité jusqu'à ce que les traitements soient terminés.

DIAGRAMME D'ACTIVITES

Composants:

1°) Nœud d'activités:

Un nœud d'activité est un type d'élément abstrait permettant de représenter les étapes le long du flot d'une activité. Il existe trois familles de nœuds d'activités :

- *les nœuds d'exécutions (executable node en anglais)* ;
- *les nœuds objets (object node en anglais)* ;
- *et les nœuds de contrôle (control nodes en anglais)*.

Représentation graphique des nœuds d'activité : De la gauche vers la droite, nous trouvons : le nœud représentant une action, qui est une variété de nœud exécutable, un nœud objet, un nœud de décision ou de fusion, un nœud de bifurcation ou d'union, un nœud initial, un nœud final et un nœud final de flot.

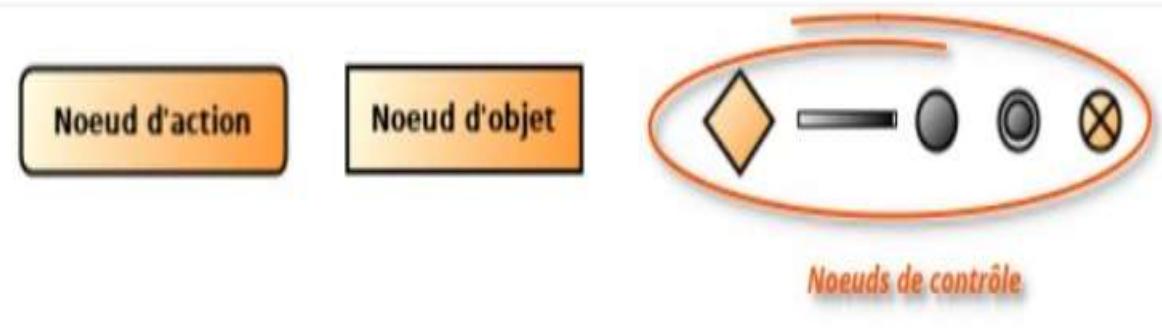


DIAGRAMME D'ACTIVITES

2°) Transition: Le passage d'une activité vers une autre est matérialisé par une transition.

Elles sont déclenchées dès que l'activité source est terminée et provoquent automatiquement et immédiatement le début de la prochaine activité à déclencher (l'activité cible).

Les transitions spécifient l'enchaînement des traitements et définissent le flux de contrôle.



DIAGRAMME D'ACTIVITES

3°) Les nœuds d'actions:

Saisir code

- Un nœud d'action est un nœud d'activité exécutable qui constitue l'unité fondamentale de fonctionnalité exécutable dans une activité.
- L'exécution d'une action représente une transformation ou un calcul quelconque dans le système modélisé.
- Les actions sont généralement liées à des opérations qui sont directement invoquées.
- Un nœud d'action doit avoir au moins un arc entrant.

DIAGRAMME D'ACTIVITES

Dans l'exemple ci-contre, **nous détectons l'arrivée du train**; cette action représente l'action "accept event" c'est-à-dire que le système reçoit le signal de l'arrivée du train.

Deuxièmement, "faire clignoter les feux" est une action "send signal", cela veut dire que le système envoie un signal qui est transmis à un objet cible sans attendre que ce dernier ait bien reçu ce signal.

Ensuite, l'action "time event" est un événement temporel déclenché après l'écoulement d'une certaine durée.

Enfin, "abaisser la barrière" est une action "send signal", un message est envoyé et transmis à la cible.

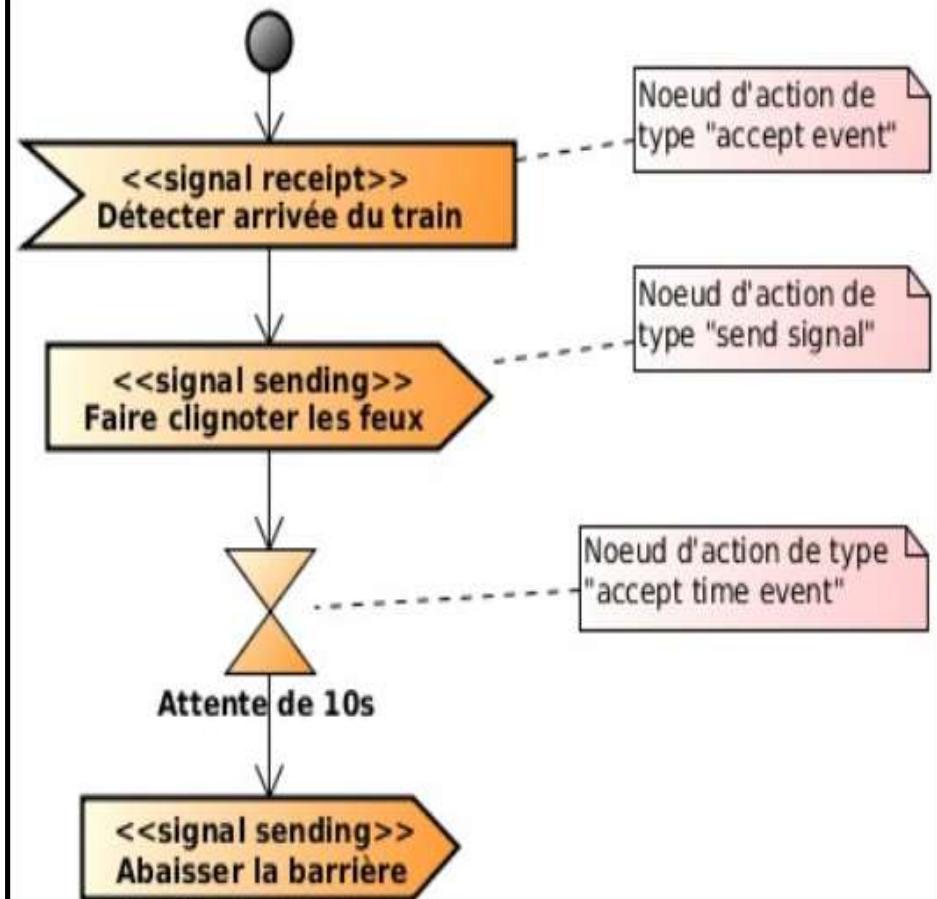


DIAGRAMME D'ACTIVITES

4°) Les nœuds de contrôle :

Un nœud de contrôle est un nœud d'activité abstrait utilisé pour coordonner les flots entre les nœuds d'une activité. Il existe plusieurs types de nœuds de contrôle :

- **nœud initial (initial node en anglais)** ;
- **nœud de fin d'activité (final node en anglais)** ;
- **nœud de fin de flot (flow final en anglais)** ;
- **nœud de décision (decision node en anglais)** ;
- **nœud de fusion (merge node en anglais)** ;
- **nœud de bifurcation (fork node en anglais)** ;
- **nœud d'union (join node en anglais)**.

DIAGRAMME D'ACTIVITES

4.1. Nœud initial :

Un nœud initial est un nœud de contrôle à partir duquel le flot débute lorsque l'activité enveloppante est invoquée. Une activité peut avoir plusieurs nœuds initiaux. Un nœud initial possède un arc sortant et pas d'arc entrant.

Graphiquement, un nœud initial est représenté par un petit cercle plein.



4.2. Nœud final:

Un nœud final est un nœud de contrôle possédant un ou plusieurs arcs entrants et aucun arc sortant. Il existe deux types de nœud finaux :

➤ **Nœud de fin d'activité** : Lorsque l'un des arcs d'un nœud de fin d'activité est activé, l'exécution de l'activité enveloppante s'achève et tout nœud ou flot actif au sein de l'activité enveloppante est abandonné. Graphiquement, un nœud de fin d'activité est représenté par un cercle vide contenant un petit cercle plein.



➤ **Nœud de fin de flot** : Lorsqu'un flot d'exécution atteint un nœud de fin de flot, le flot en question est terminé, mais cette fin de flot n'a aucune incidence sur les autres flots actifs de l'activité enveloppante.

Graphiquement, un nœud de fin de flot est représenté par un cercle vide barré d'un X



DIAGRAMME D'ACTIVITES

4.3°) Nœud de décision /Branchements conditionnel



Un nœud de décision est un nœud de contrôle qui permet de faire un choix entre plusieurs flots sortants. Il possède un arc entrant et un ou plusieurs arcs sortants.

Condition de garde [entre crochets]. Il est possible d'associer une condition à une transition, qui est alors appelée condition de garde. Pour que la transition soit franchie, il faut que la condition soit remplie. C'est une expression booléenne qui doit être vraie pour que la transition soit déclenchée.

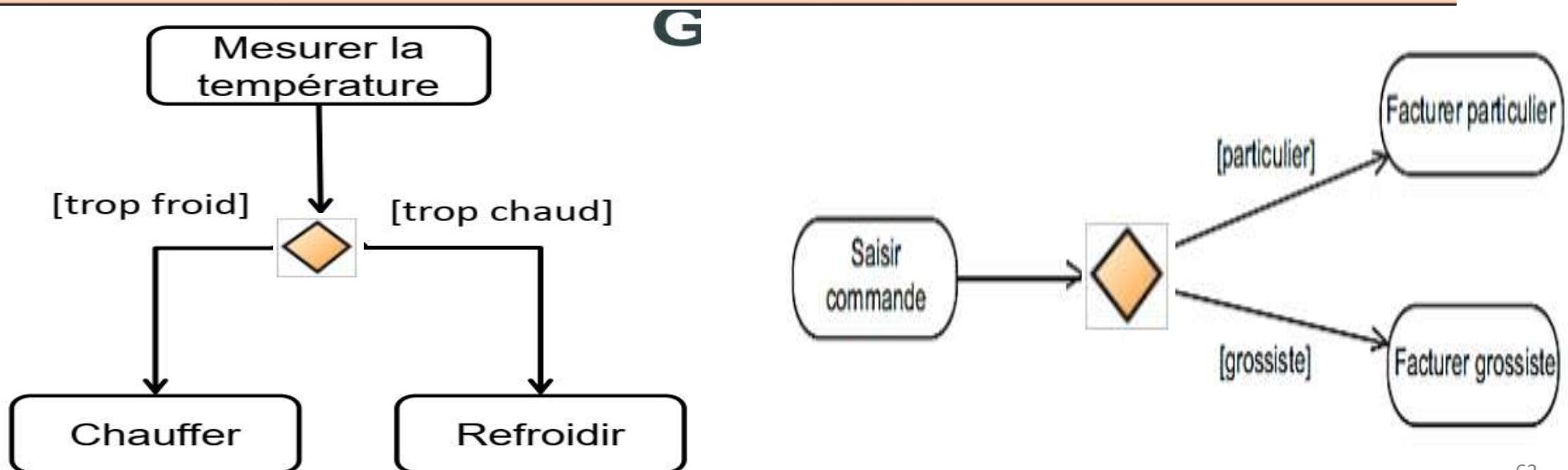


DIAGRAMME D'ACTIVITES

4.4°) Nœud de fusion/convergence :



Un nœud de fusion est un nœud de contrôle qui rassemble plusieurs flots alternatifs entrants en un seul flot sortant.

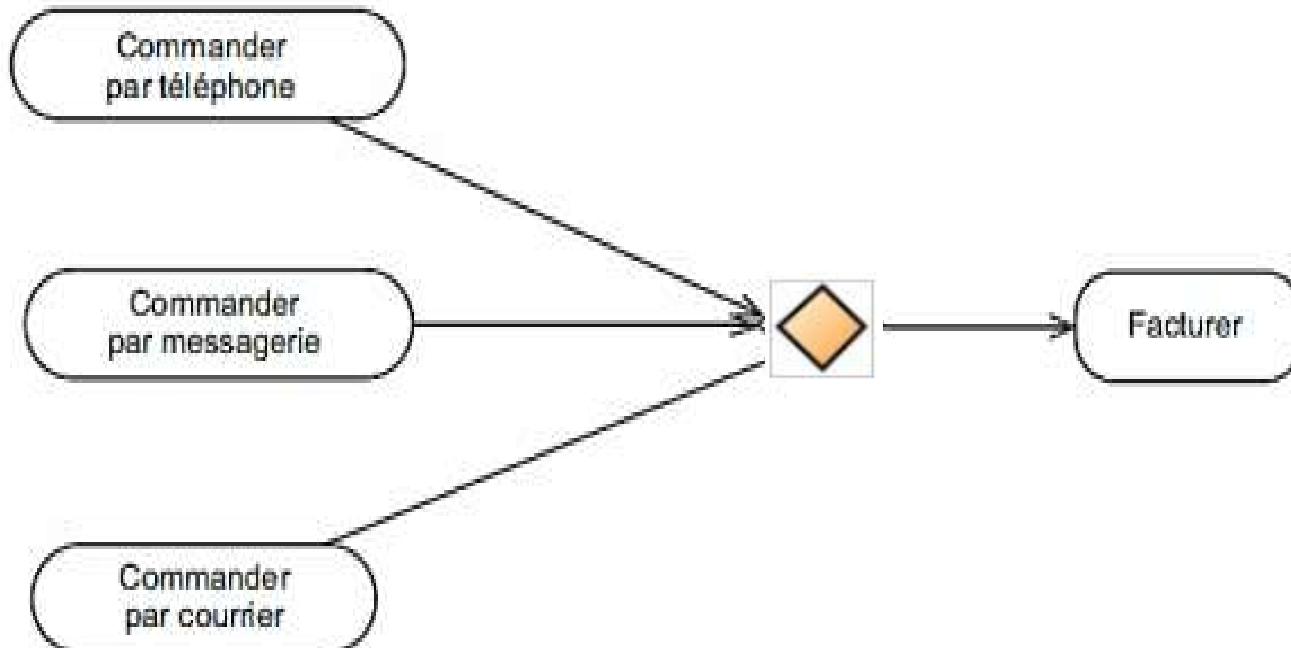


DIAGRAMME D'ACTIVITES

4.5°) Nœud de bifurcation/débranchement /Parallélisme :

Un nœud de bifurcation, est un nœud de contrôle qui sépare un flot en plusieurs flots concurrents. Un tel nœud possède donc un arc entrant et plusieurs arcs sortants.

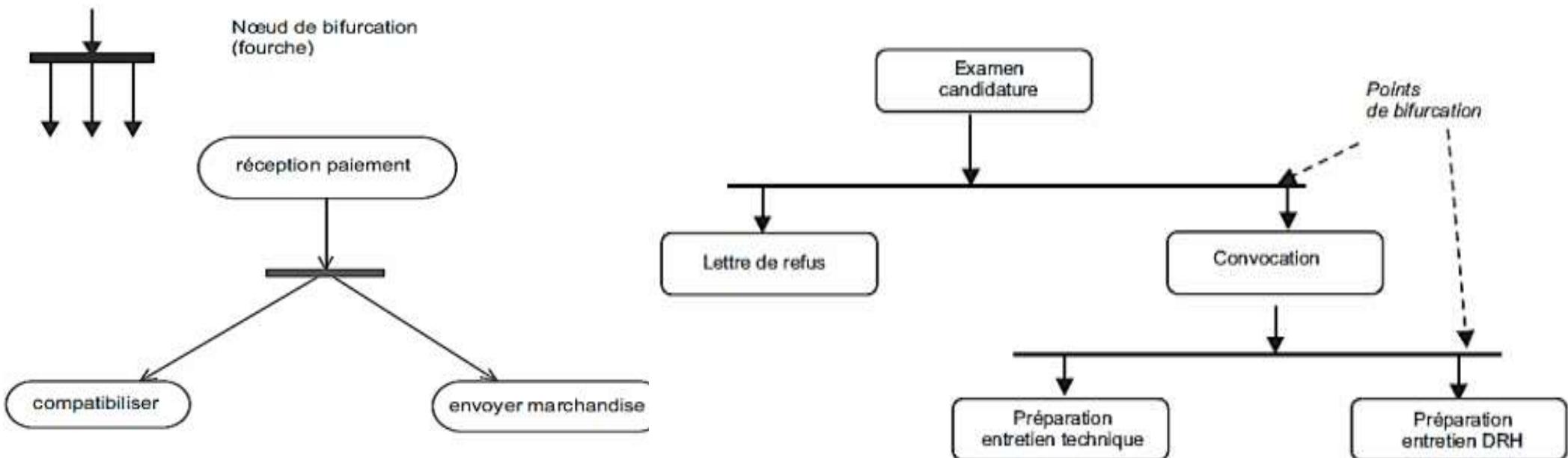


DIAGRAMME D'ACTIVITES

4.6) Nœud jonction/Synchronisation :

Un nœud de jonction, est un nœud de contrôle qui synchronise des flots multiples.

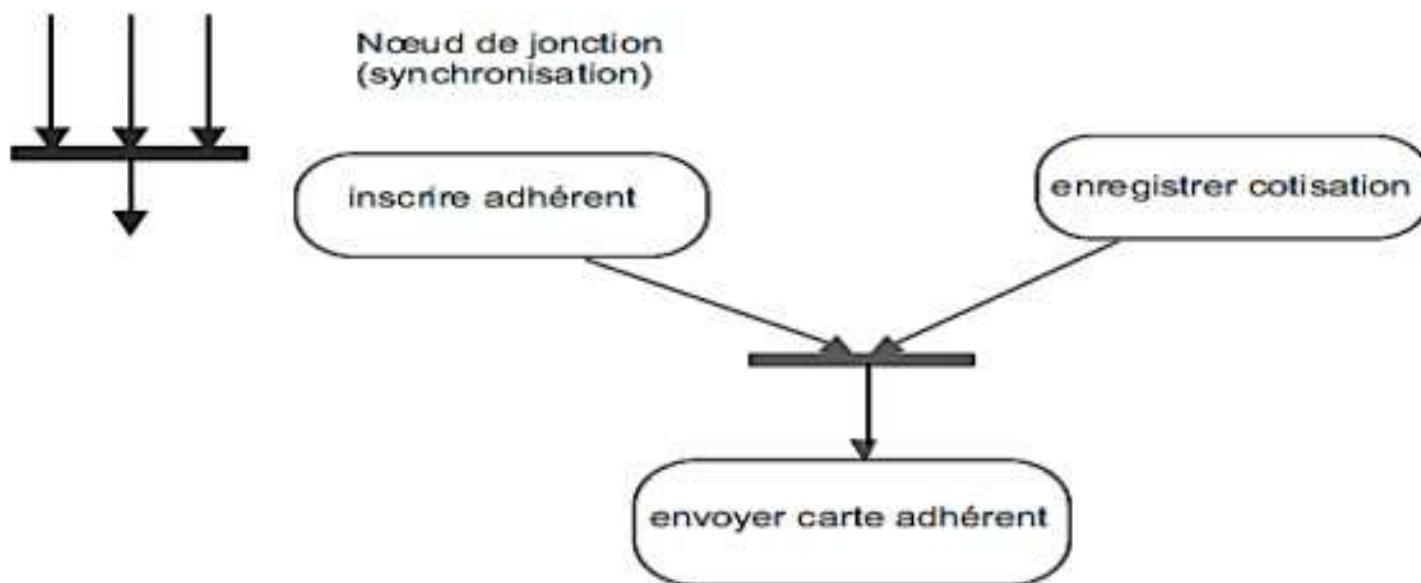


DIAGRAMME D'ACTIVITES

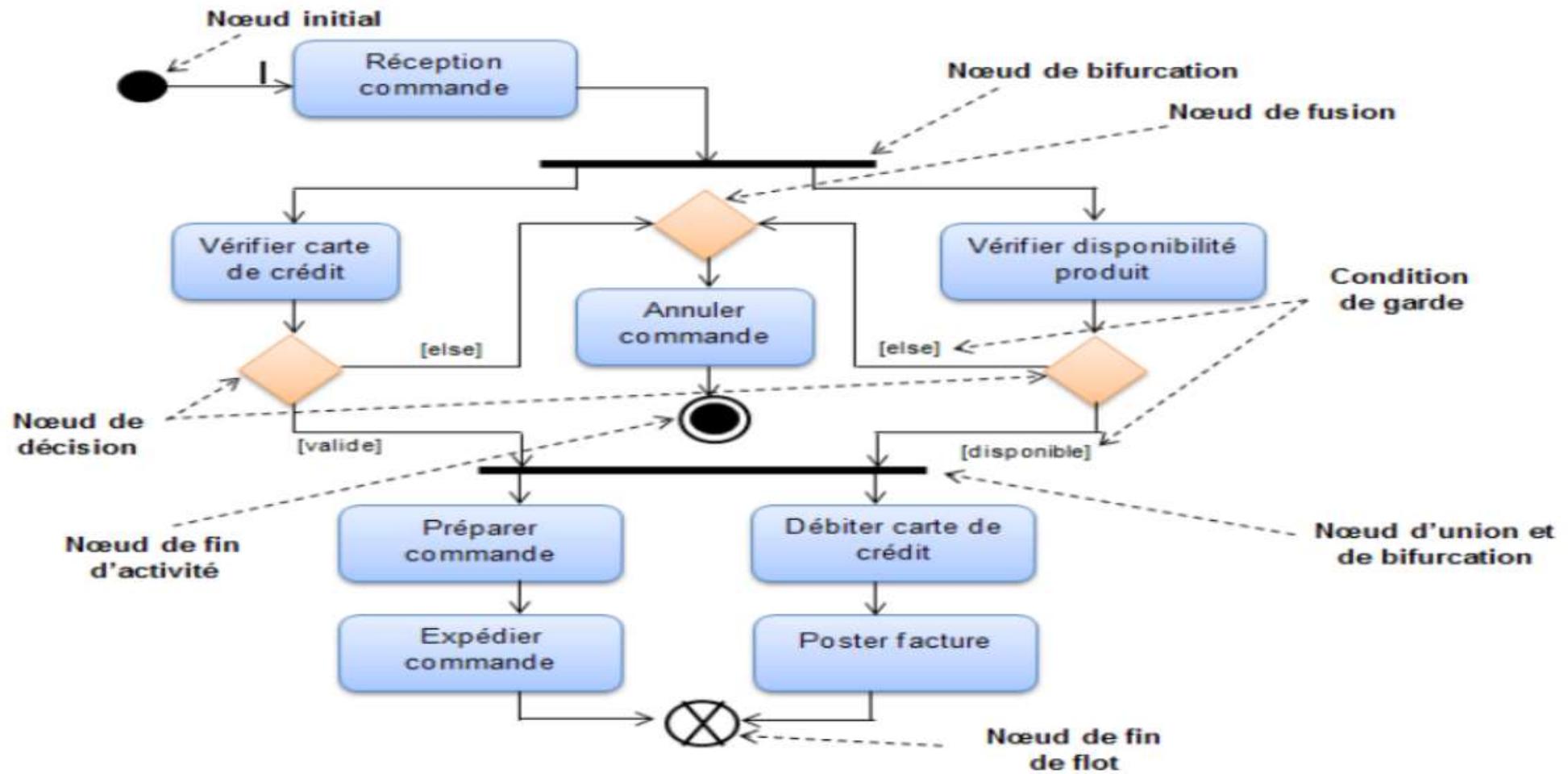


DIAGRAMME D'ACTIVITES

DECOUPAGE (COULOIR D'ACTIVITE – SWIM LANE):

- Pour montrer les différentes responsabilités au sein d'un mécanisme ou d'une organisation, on schématisé des couloirs d'activités.
- Chaque activité est allouée à un couloir correspondant à la ressource concernée : partenaire, travailleur
- Chaque partition montre ainsi quelles actions sont exécutées par une classe ou par une unité organisationnelle.

COULOIR D'ACTIVITE

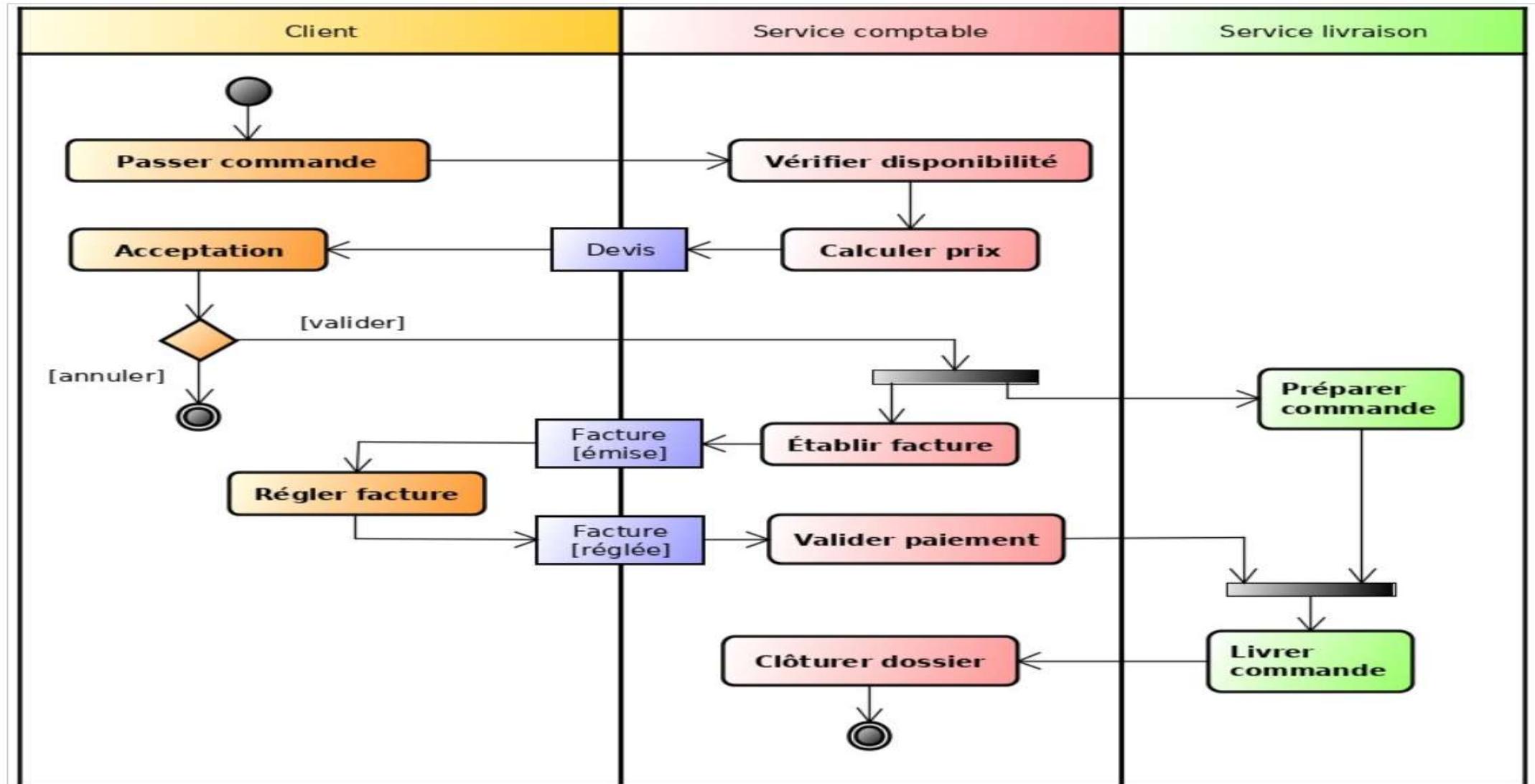


DIAGRAMME D'ACTIVITES

Exercice 1: 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.

DIAGRAMME D'ACTIVITES

Exercice 2 *Créer un diagramme d'activité, en indiquant les différentes partitions, pour le use case « Créer une fiche de réparation » décrit comme suit;*

Pour créer une fiche de réparation, le chef d'atelier saisit les critères de recherche de voitures dans le système.

Le logiciel de gestion des réparations lui donne la liste des voitures correspondant aux critères entrés.

Si la voiture existe, le chef d'atelier va sélectionner la voiture. Le logiciel va, ensuite, fournir les informations sur le véhicule. Si la voiture est sous garantie, le chef devra saisir la date de demande de réparation.

Si la voiture n'existe pas, le chef va saisir les informations concernant ce nouveau véhicule.

Dans tous les cas, le chef d'atelier devra saisir la date de réception et de restitution.

Si le dommage de la voiture est payé par l'assurance, le logiciel va fournir une liste d'assurances au chef d'atelier. Ce dernier sélectionnera l'assurance adéquate.

Enfin, le logiciel enregistre la fiche de réparation.

Diagramme de Composants

- Le diagramme de composants permet de représenter les composants logiciels d'un système ainsi que les liens existants entre ces composants.
- Le diagramme de composants décrit les éléments physiques qui réalisent les classes et leurs relations dans l'environnement de réalisations.
- Un composant est un élément logiciel autonome, remplaçable et réutilisable qui **fourni** ou **reçoit** un service bien précis.: fichiers sources (.java, .cpp, .h, .cs...) librairies (dll, jar...), exécutables...
- Les composants **fournissent** des services via des **interfaces**. Un composant peut être remplacé par n'importe quel autre composant compatible c'est-à-dire ayant les mêmes interfaces.
- Un composant peut évoluer indépendamment des applications ou des autres composants qui l'utilise à partir du moment où les interfaces sont respectées.

Diagramme de Composants

Ex : parallèlement avec les composants d'un ordinateur.

Un ordinateur est un ensemble de composants modulaires qui **fournissent** et **reçoivent** des services (carte mère, carte graphique, disque dur, clavier, écran...). Chacun de ces composants est remplaçable par un autre composant (pas forcément identique) à condition qu'il ait des **interfaces compatibles**.

ATTENTION :

En UML les composants ne sont pas des éléments matériels mais des **éléments logiciels** qui par contre seront installés sur des éléments matériels (ce que nous verrons lorsque nous aborderons le diagramme de déploiement).

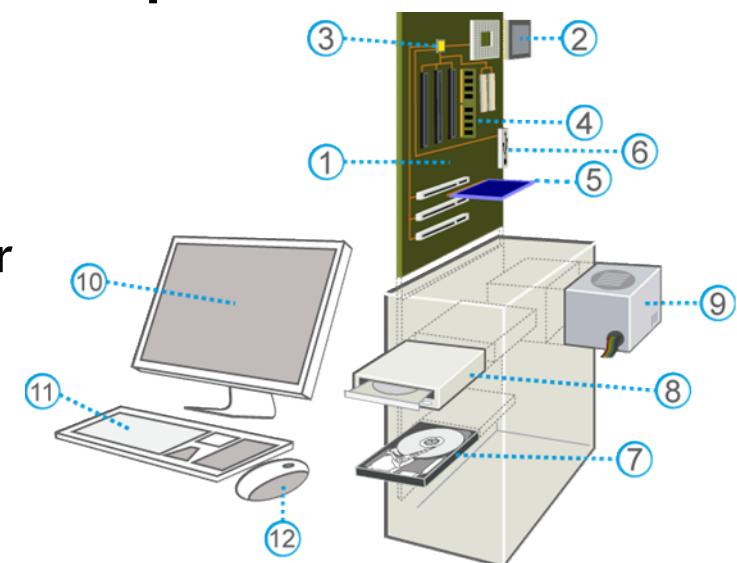
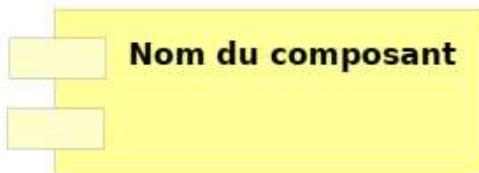


Diagramme de Composants

Un composant est représenté par :

- **Nom du composant** : Permet de distinguer un composant des autres composants. Il peut être un nom simple ou un nom composé qui indique le paquetage auquel appartient le composant.
- **Stéréotypes** : Spécifient un composant qui désigne :
 - « executable » : un programme pouvant s'exécuter sur un nœud ;
 - « library » : une bibliothèque statique ou dynamique ;
 - « table » : une table de base de données;
 - « file » : un fichier contenant du code source ou des données;
 - « document » : un document.



Un rectangle dans lequel figure :

- Le nom du composant.
- 2 petits rectangles l'un au dessus
- de l'autre à cheval du côté gauche.



Un rectangle dans lequel figure :

- Le nom du composant.
- Le symbole en haut à droite.



Un rectangle dans lequel figure :

- Le stéréotype <<component>>.
- Le nom du composant.

Diagramme de Composants

- **Les interfaces:**

Il existe deux types d'interface :

- **Les interfaces requise** : Ce sont des interfaces qui fournissent un service au composant et dont il a besoin pour fonctionner. Elles sont reliées au composant par une flèche en pointillées sur laquelle figure le stéréotype <<use>>.
- **Les interfaces fournies** : Ce sont des interfaces par lesquels le composant fourni lui-même un service. sont reliées au composant par une flèche en pointillées sur laquelle figure le stéréotype <<realize>>

Diagramme de Composants

- **Les interfaces:**

Graphiquement, elles sont représentées comme suit ;

1. Intégrées dans la représentation du composant:

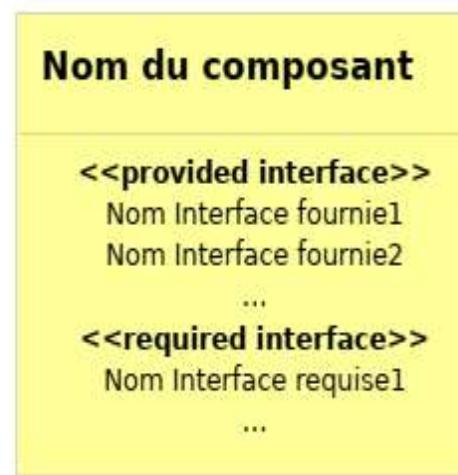
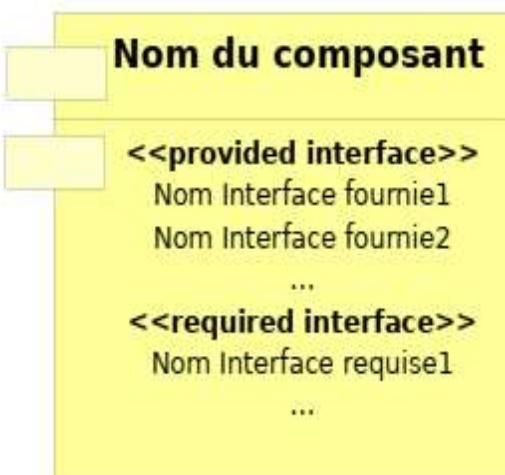
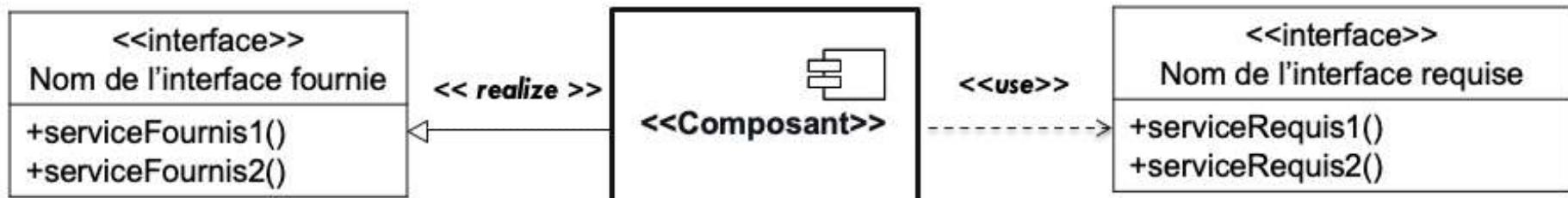


Diagramme de Composants

- 2. Dans un classeur séparé du composant dans lequel sont listés les différents services.



- 3. En utilisant des connecteurs d'assemblage

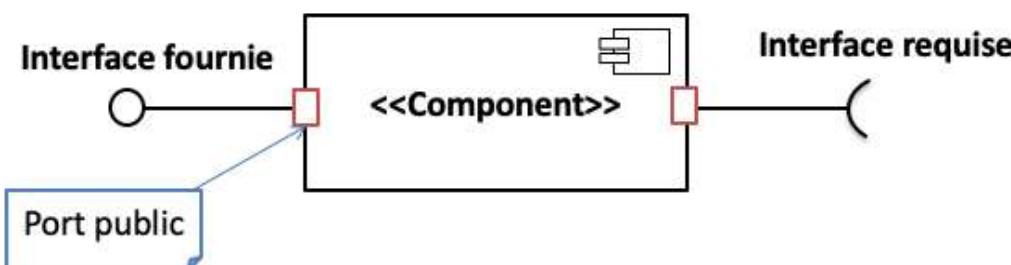
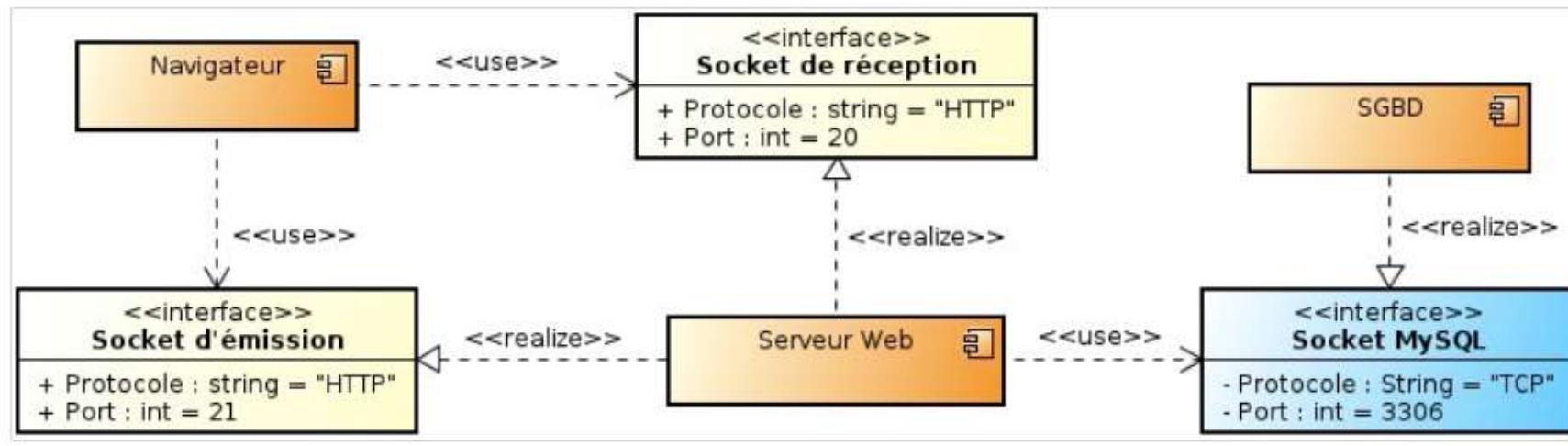


Diagramme de Composants

Exemple : Transfert des données par internet



Ou

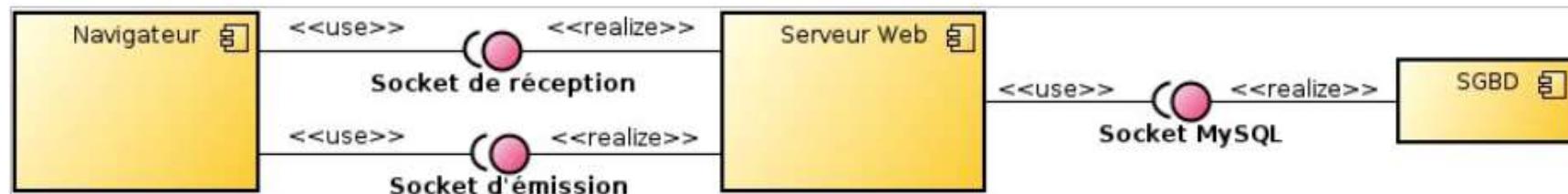


Diagramme de Composants

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 librairies dynamiques qui sont nécessaires lors de l'exécution d'un programme.

Diagramme de Composants

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

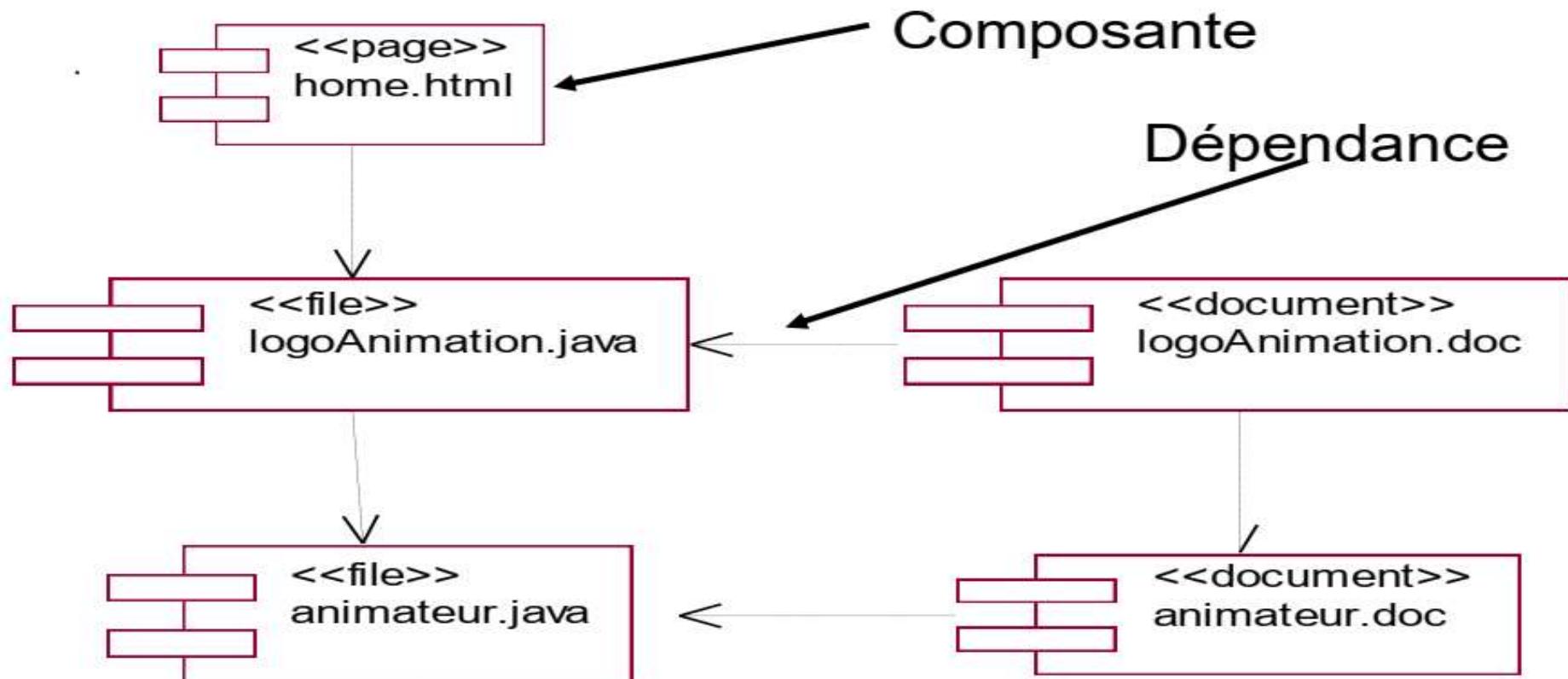
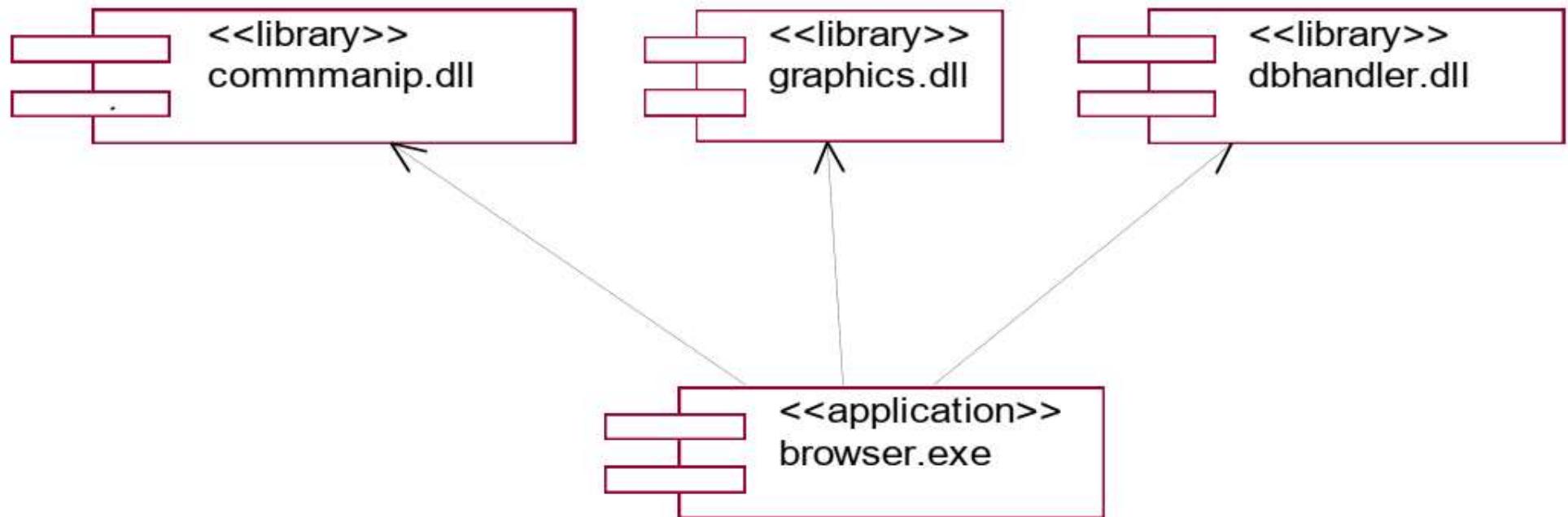


Diagramme de Composants

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



Exercices d'application

Exercice 1:

1. Le composant fraudeAgent.dll contient les classes agentDesFraudes, reglementDesFraudes et RechercheDePattern.

Question: Représentez le diagramme de composants correspondant.

1. Le composant image.java dépend de l'interface ImageObserver du composant component.java.

Question : Représentez le diagramme de composants.

Diagramme de déploiement

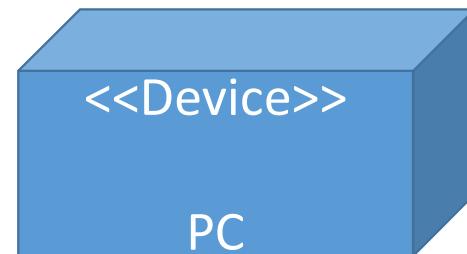
Le diagramme de déploiement permet de représenter :

1. La disposition physique des ressources matérielles qui constituent le système et montre la répartition des composants (élément logiciels) qui s'exécutent sur ces matériels.
2. Les chemins de communication entre les différentes ressources matérielles.

Diagramme de déploiement

- **Les éléments du diagramme de déploiement:**

1°) **Les nœuds :** Un nœud est une ressource matérielle du système.



Un nœud est représenté par un parallélépipède rectangle dans lequel figure son nom.



Diagramme de déploiement

- **Les éléments du diagramme de déploiement:**

- Un nœud possède des attributs (quantité de mémoire, vitesse de processeur, marque, type...) que nous pouvons spécifier à l'intérieur du parallélépipède.



Diagramme de déploiement

□ Les éléments du diagramme de déploiement:

- Pour montrer qu'un composant est affecté sur un nœud, il faut :
 - ✗ Soit en plaçant le composant dans le nœud.
 - ✗ Soit en le reliant à l'aide d'une relation de dépendance (flèche en pointillées) stéréotypée «**support**» orientée du composant vers le nœud .

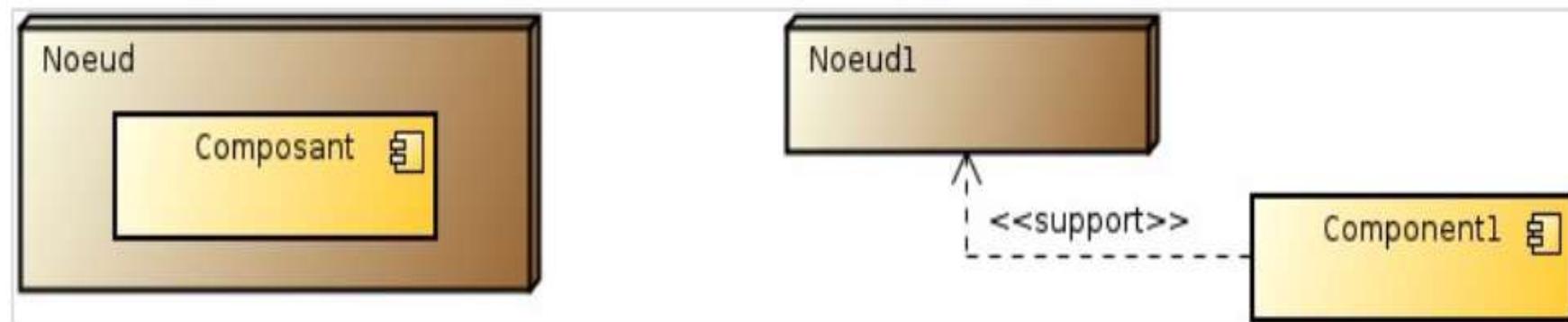


Diagramme de déploiement

- **Les éléments du diagramme de déploiement:**

2°) **Les chemins de communications**

Les différents nœuds qui apparaissent dans le diagramme de déploiement sont connectés entre eux par des lignes qui symbolisent un support de communication → Ce sont les chemins de communications. ***Le chemin de communication est donc un lien qui permet de modéliser de façon simpliste la communication entre 2 nœuds (liaison Ethernet, USB, série...).***

Il est possible de faire figurer sur ce lien :

- les cardinalités ;
- des contraintes entre accolades (pour indiquer par exemple qu'un accès est sécurisé) ;
- le type de réseau et/ou son débit en l'indiquant comme un stéréotype...



Exemple réel: L'application Smarteam

L'architecture de l'application est constituée des ressources suivantes:

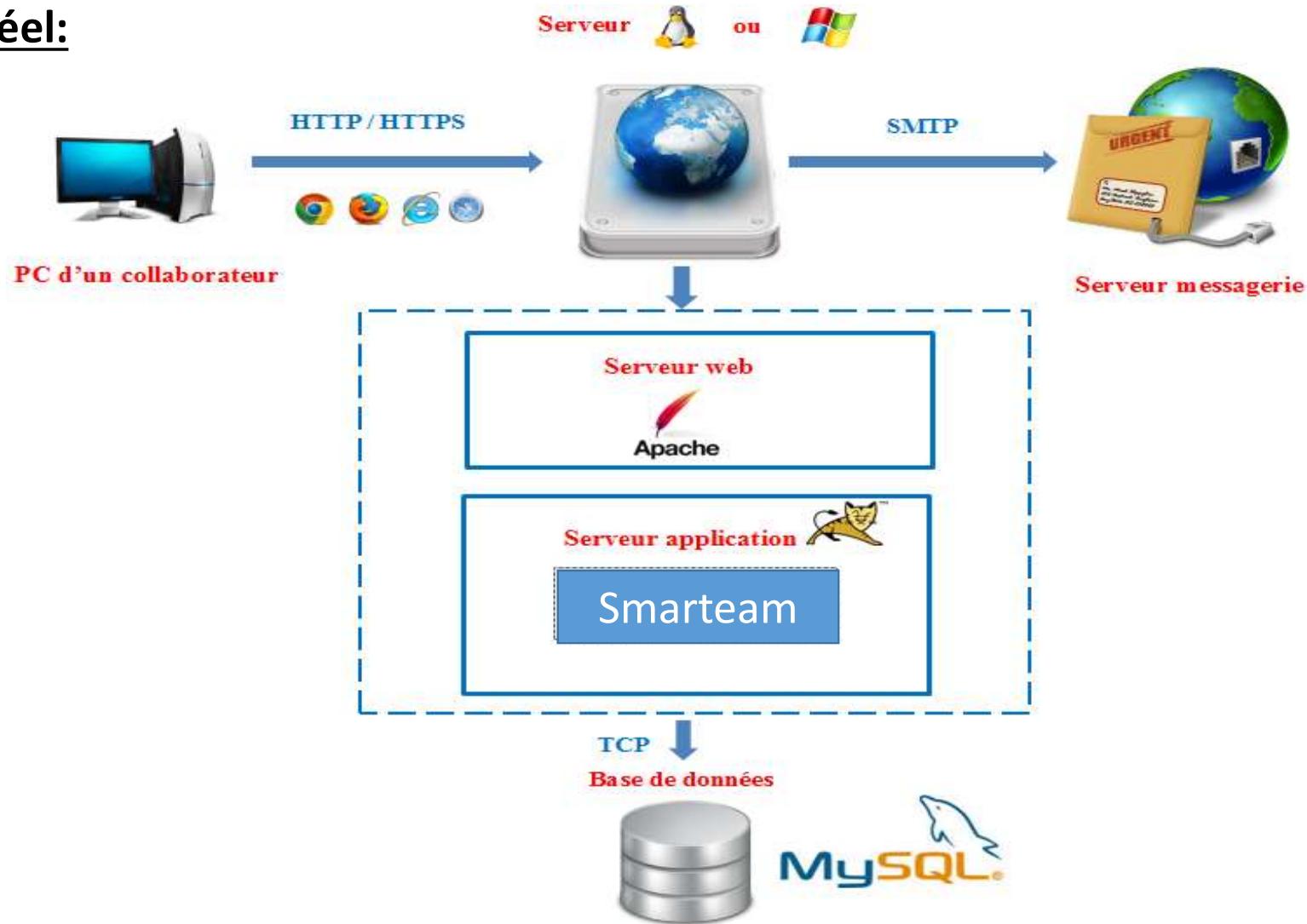
- **Client web** : Il représente les navigateurs web avec lesquels se connectent les différents collaborateurs à l'application.
- **Serveur UNIX / WINDOWS** : Il représente le serveur dans lequel est déployé l'application. Ce serveur héberge les deux serveurs
 - le serveur web Apache
 - le serveur application Tomcat

Les deux serveurs sont fusionnés en Apache-Tomcat version x. x . Le serveur UNIX ou WINDOWS doit être accessible aux utilisateurs sur un réseau LAN ou WAN.

Comme une installation préalable de la JVM JDK x.x compatible avec Tomcat est exigée.

- **SGBD** : Système de gestion de base de données MySQL x.x assurant le stockage des données
- **Serveur SMTP** : Il achemine les emails générés par l'application à destination des utilisateurs lors de la création de leurs profils.

Exemple réel:



Architecture Technique de l'application smarteam

Exemple réel: L'application Smarteam

L'architecture de l'application est constituée des ressources suivantes:

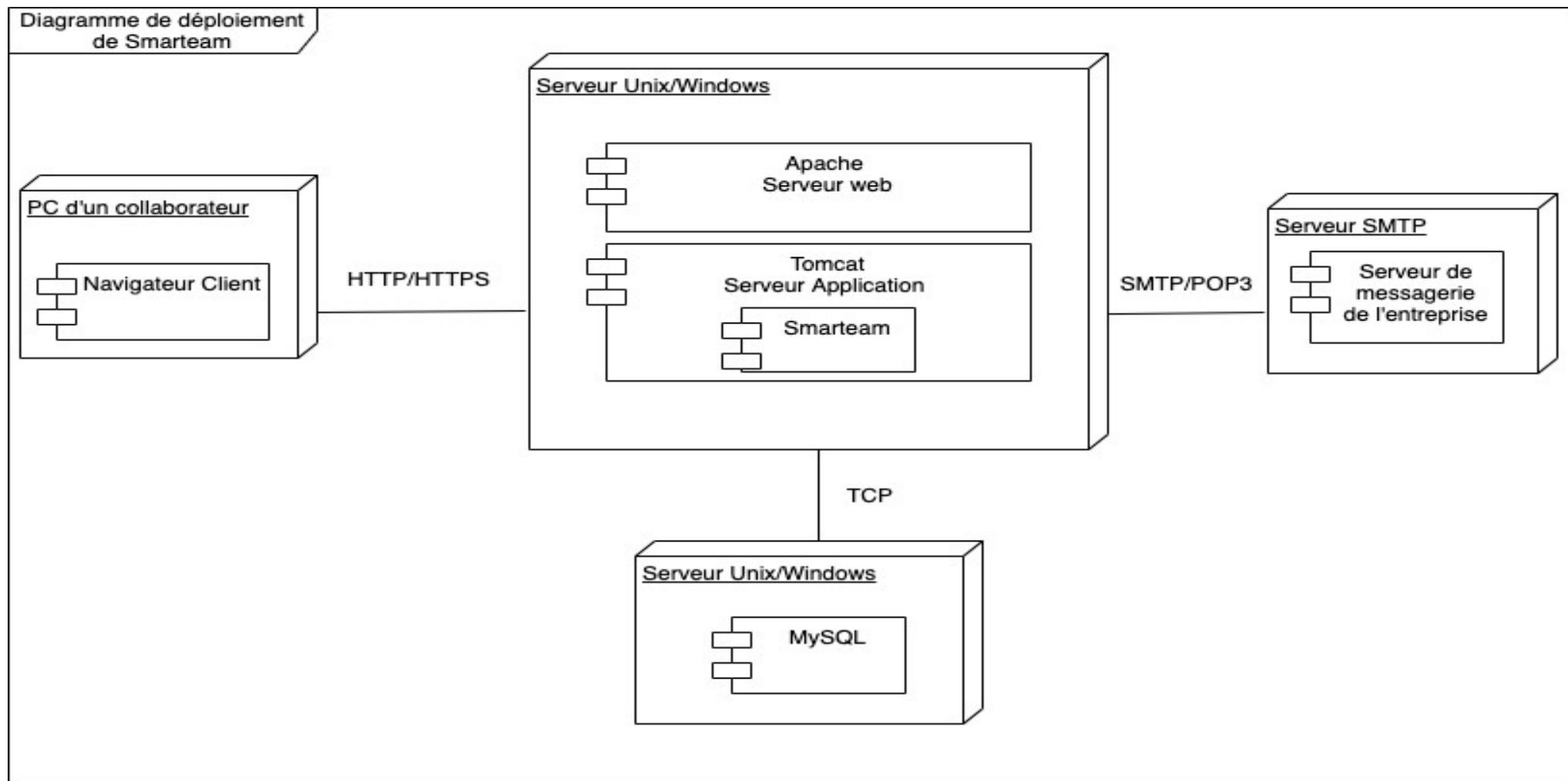
- **Client web** : Il représente les navigateurs web avec lesquels se connectent les différents collaborateurs à l'application.
- **Serveur UNIX / WINDOWS** : Il représente le serveur dans lequel est déployé l'application. Ce serveur héberge les deux serveurs
 - le serveur web Apache
 - le serveur application Tomcat

Les deux serveurs sont fusionnés en Apache-Tomcat version x. x . Le serveur UNIX ou WINDOWS doit être accessible aux utilisateurs sur un réseau LAN ou WAN.

Comme une installation préalable de la JVM JDK x.x compatible avec Tomcat est exigée.

- **SGBD** : Système de gestion de base de données MySQL x.x assurant le stockage des données
- **Serveur SMTP** : Il achemine les emails générés par l'application à destination des utilisateurs lors de la création de leurs profils.

Exemple réel:



L'architecture technique de l'application smarTeam sous le diagramme de déploiement

Exercices d'application

Exercice 2:

Une architecture matérielle est constituée d'un serveur, d'un kiosque et d'une console. Le serveur utilise une tour de disques durs montés en RAID. Le serveur communique avec le kiosque par une liaison Ethernet 100bT et la console par une liaison RS232C. Les caractéristiques du serveur sont :

- Processeur Pentium
- Mémoire vive : 4 Go
- Ecran 17 pouces

On peut aussi préciser les modules exécutables sur chaque noeud :

- sur le kiosque : user.exe
- sur le serveur : dbadmin.exe et tkmstr.exe
- sur la console : admin.exe et cong.exe

Représentez le diagramme de déploiement correspondant.

Diagramme de communication

- Diagramme de collaboration (UML 1.x)
- Diagramme de communication (UML 2.0)
- Est un diagramme de comportement, dynamique.
- Il montre les interactions/messages entre les différents objets/Acteurs
- Est une représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets/Acteurs.

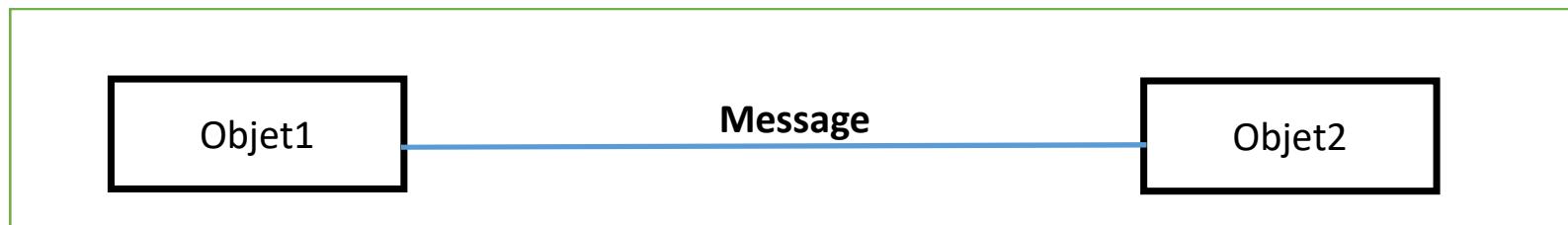


Diagramme de communication

Les éléments d'un diagramme de communication;

1. Les objets/Acteurs
2. Les liens
3. Les flèches
4. Les messages

Diagramme de communication

1. Les objets



2. Les liens: Un lien montre la connexion entre les objets/Acteurs



3. Les flèches: indiquent la direction dans laquelle le message circule

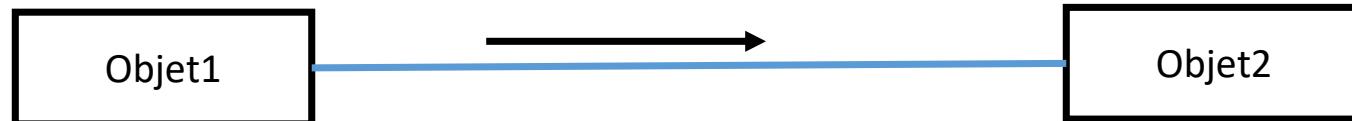
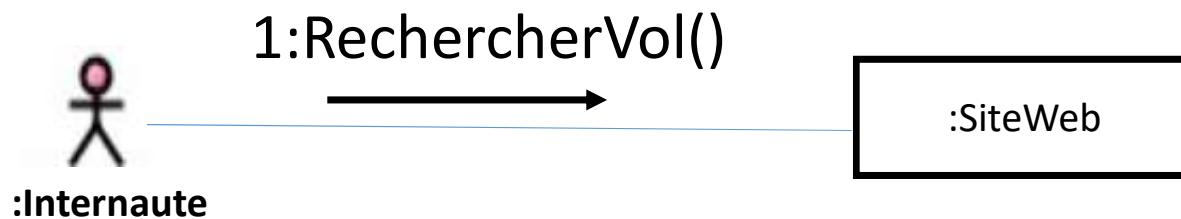


Diagramme de communication

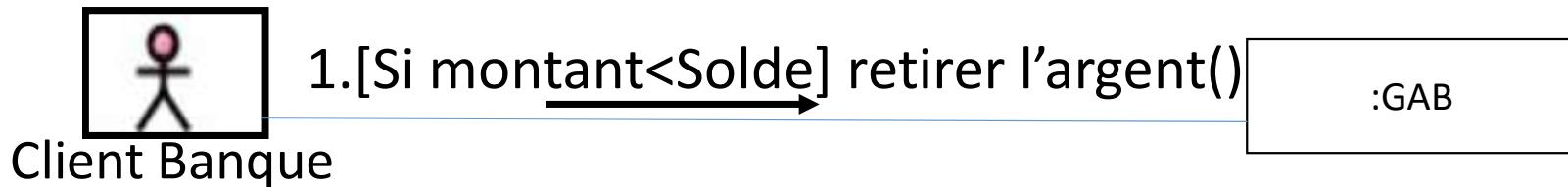
4. Message: Déclenche l'exécution d'une opération de l'objet destinataire



[garde][séquence][* itération][valeur retour:=]signatureMessage(arguments)

Diagramme de communication

4.1 Message: [Garde]; c'est la condition d'envoie du message



4.2 Message: [arguments];



Diagramme de communication

4.3 Message: [séquence]; est le numéro de séquence du message.

On numérote les messages par envoi et sous-envoi désignés par des chiffres séparés par des points : ainsi l'envoi du message 1.4.4 est postérieur à celui du message 1.4.3, tous deux étant des conséquences (i.e. des sous-envois) de la réception d'un message 1.4.

La simultanéité d'un envoi est désignée par une lettre : les messages 1.1a et 1.1b sont envoyés ***en même temps***.

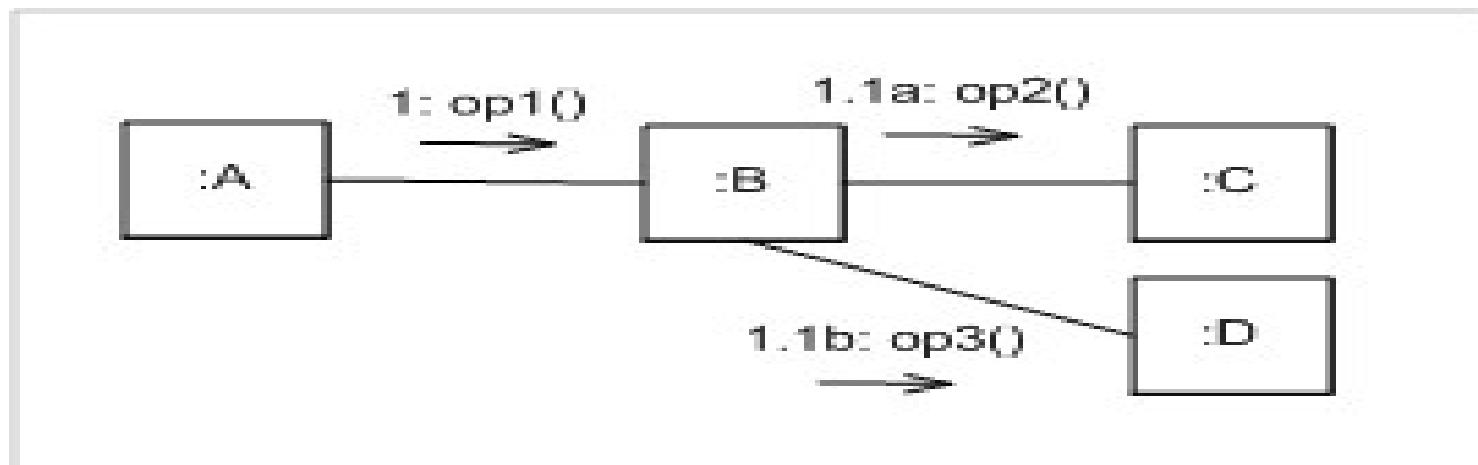


Diagramme de communication

4.4 Message: [***itération**]; spécifie l'envoi de plusieurs messages, on utilise (*) pour désigner un message récurrent, envoyé un certain nombre de fois.

4.5 Message: [**valeur retour:=**] est la valeur de retour du message, qui sera par exemple transmise en paramètre à un autre message.

4.6 Message : **signatureMessage** est le nom du message.

4.7 Message: [**paramètre**] désigne les paramètres (optionnels) du message.

Diagramme de communication

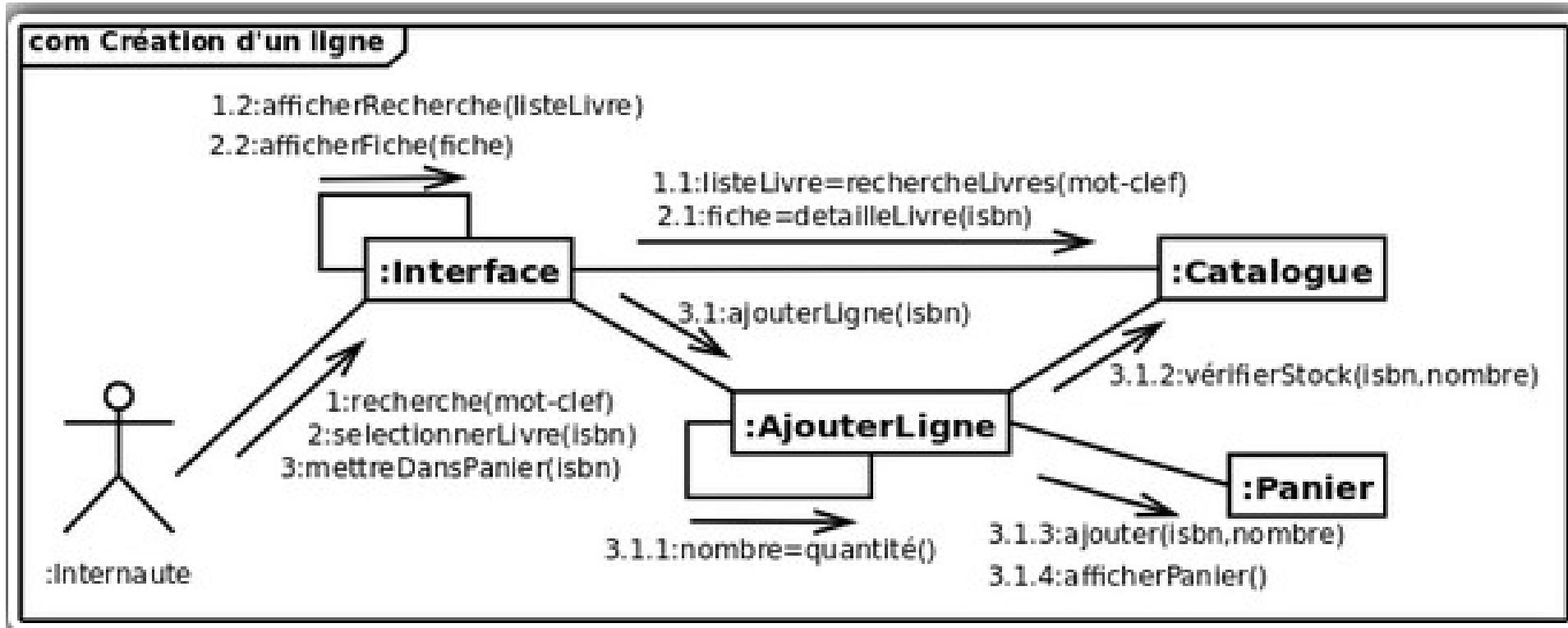


Diagramme de communication illustrant la recherche puis l'ajout, dans son panier virtuel, d'un livre lors d'une commande sur Internet

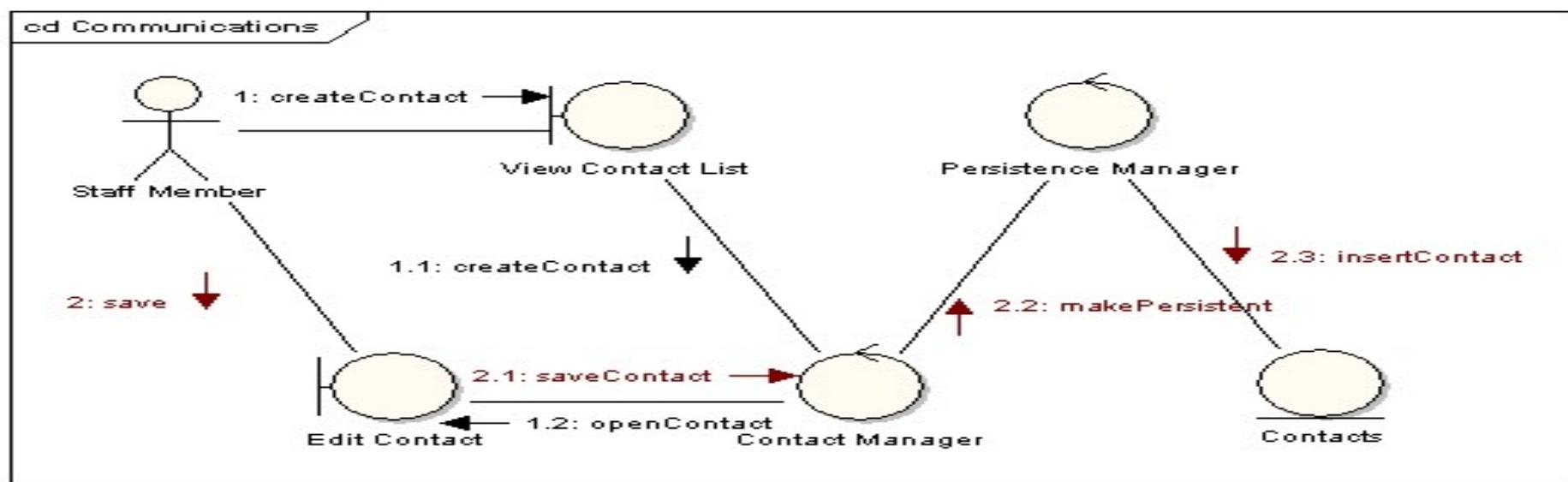
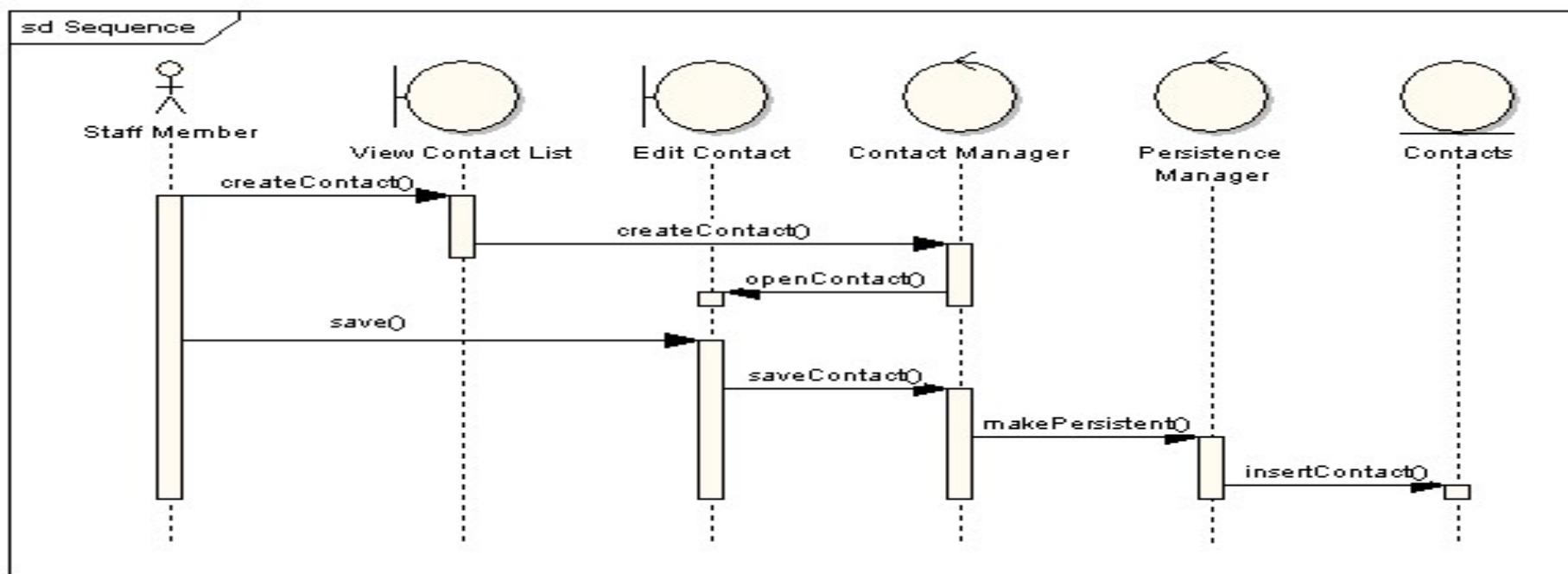


Diagramme de communication

Exercice d'application: Proposez un diagramme de communication équivalent au diagramme de séquence suivant

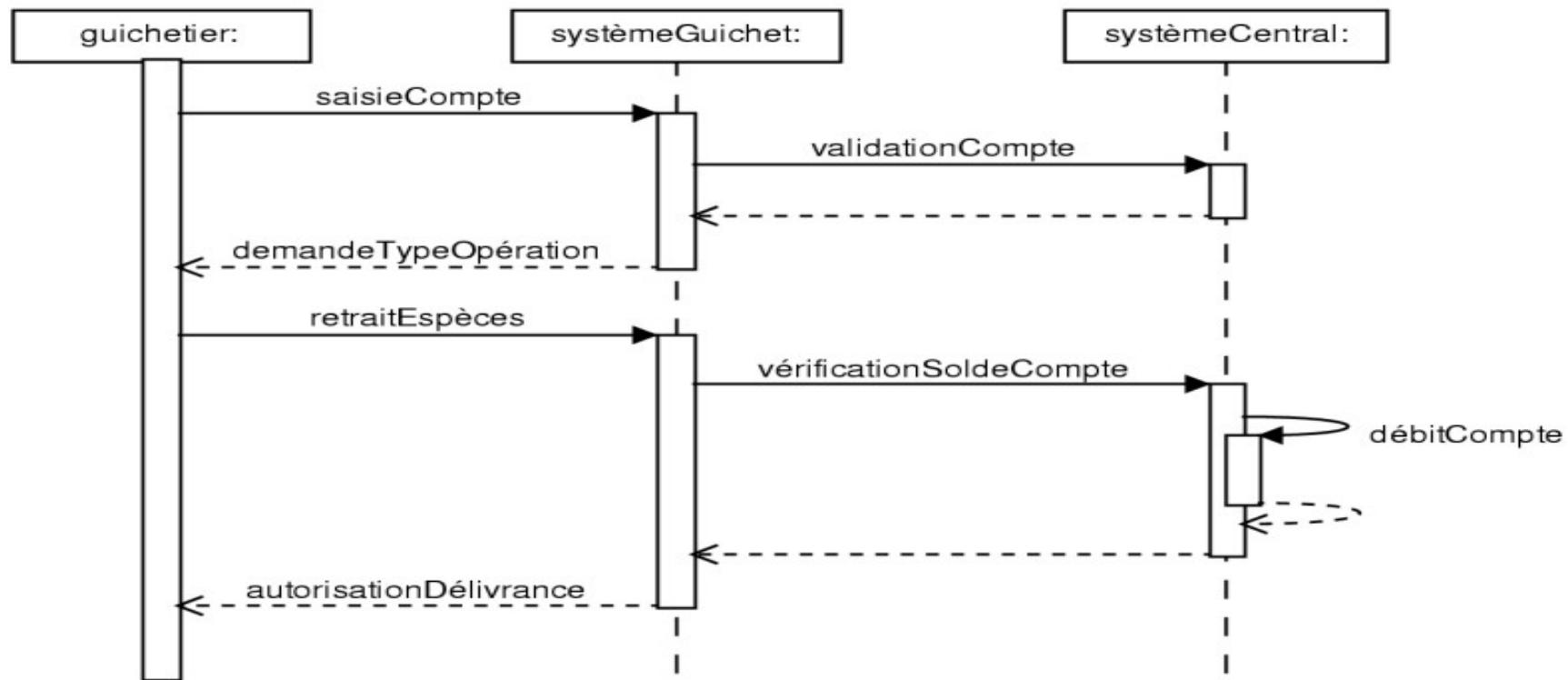


Diagramme d'interaction Globale

Le diagramme d'interaction globale représente une fusion entre le diagramme d'activité et le diagramme de séquence.

Il permet d'organiser des interactions, représentées par exemple par des diagrammes de séquence, au moyen des nœuds de contrôle du diagramme d'activité : décision, parallélisme, etc.

C'est donc une sorte de diagramme d'activité dans lequel les actions sont remplacées par des interactions schématisées via un diagramme de séquence.

Diagramme d'interaction Globale

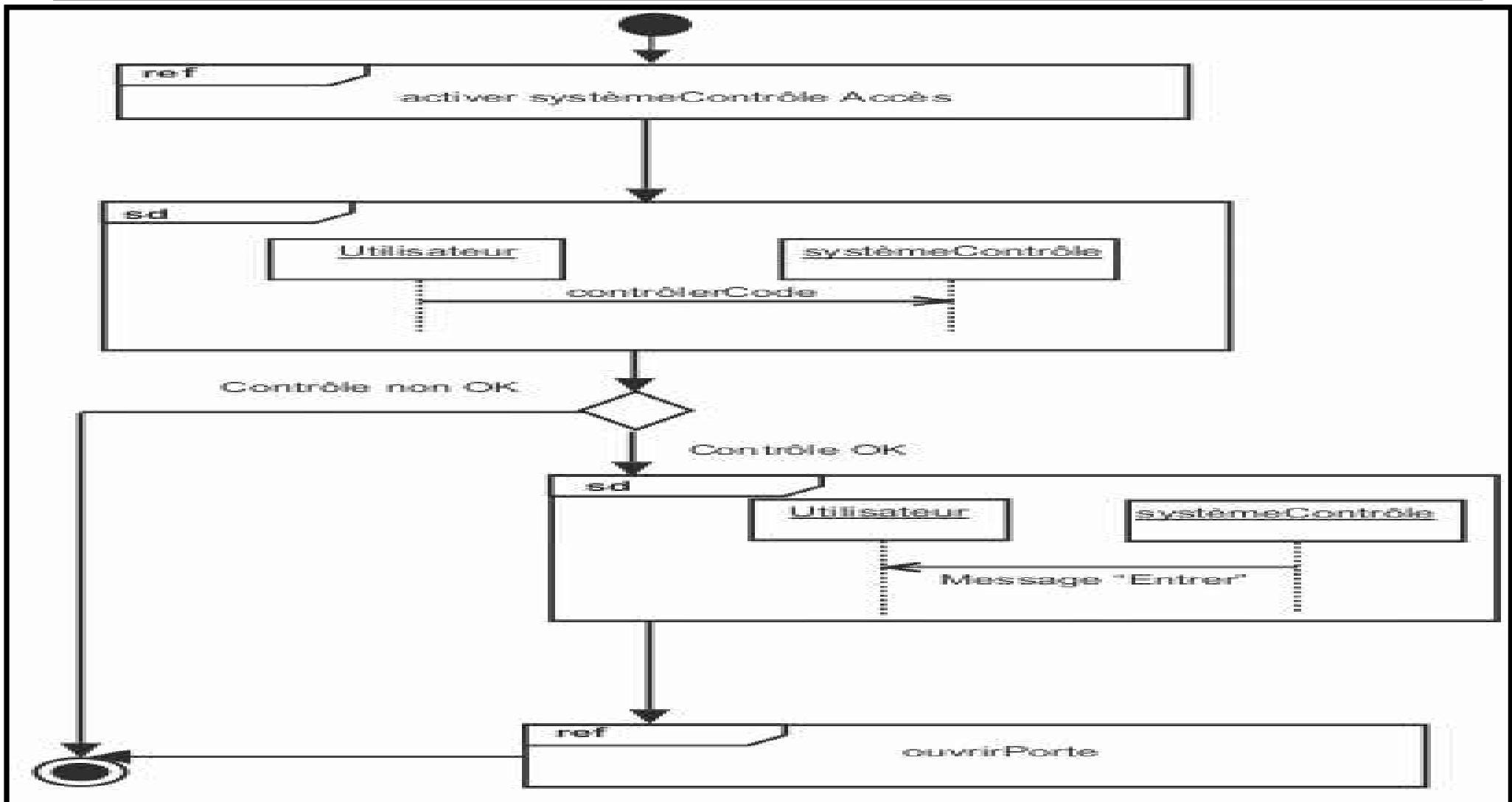


Diagramme de temps

- Introduit en UML 2, le diagramme de temps montre les changements d'état d'un objet quand ceux-ci dépendent exclusivement du temps.
- Le diagramme indique alors la durée minimale et/ou maximale de chaque état à l'aide de contraintes temporelles.
- Le diagramme de temps est l'un des quatre diagrammes d'interaction, est donc est un diagramme de comportement.
- Il consiste en un graphique ou des vagues qui représentent l'état d'une ligne de vie à un moment précis.

Diagramme de temps

Les composants majeurs d'un diagramme de temps

1. Line de vie:

- La ligne de vie représente une entité (elle ne représente qu'une seule entité), qui fait partie de l'interaction.
- Il représente le nom du classificateur ou l'instance qu'il représente. Une ligne de vie peut être placée dans un cadre de diagramme.

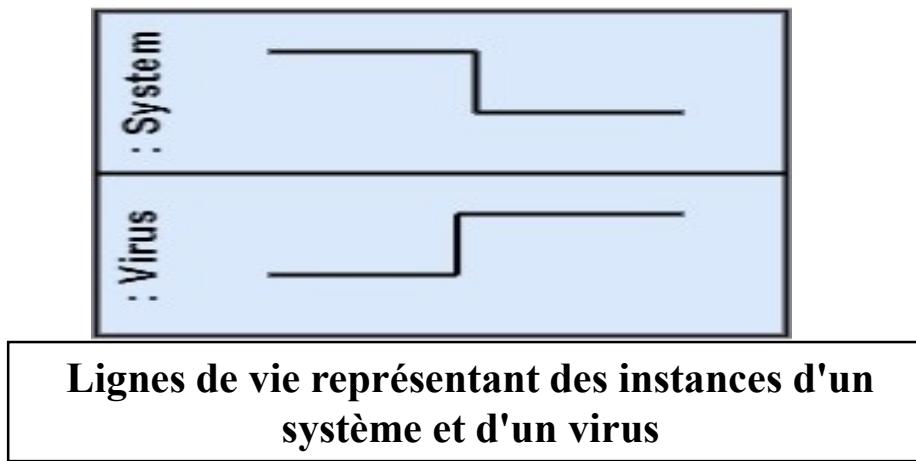
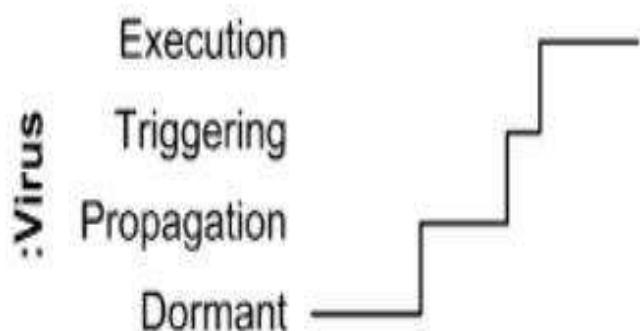


Diagramme de temps

Les composants majeurs d'un diagramme de temps

2. La Chronologie de l'état ou de la condition :

Les chronologies de condition et d'état sont utilisées pour représenter la coopération des attributs et des états de classificateur.



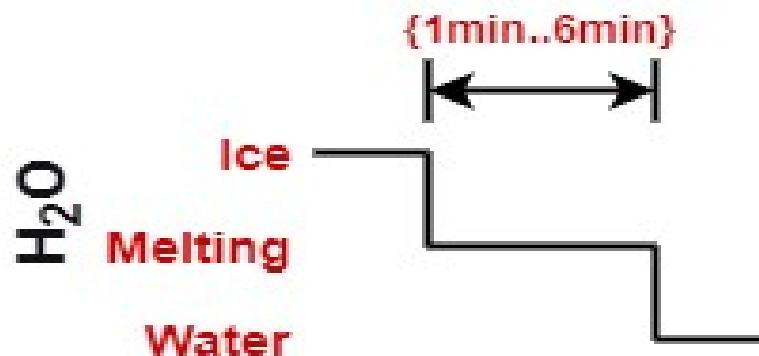
Timeline shows Virus changing its state between Dormant, Propagation, Triggering and Execution state

Diagramme de temps

Les composants majeurs d'un diagramme de temps

3. Contrainte de durée:

- Est une contrainte d'intervalle, qui fait référence à un intervalle de durée.
- Il permet de déterminer si la contrainte est satisfaite pendant une durée ou non.
- La trace négative définit les contraintes violées, ce qui signifie que le système est défaillant.



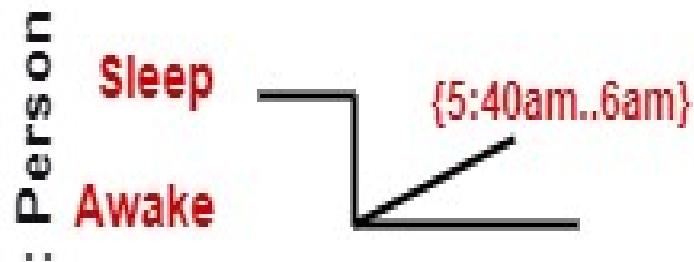
La glace devrait fondre dans l'eau en 1 à 6 minutes

Diagramme de temps

Les composants majeurs d'un diagramme de temps

4. Contrainte de temps :

- une contrainte d'intervalle de temps pendant lequel quelque chose doit être accompli par le participant.
- L'association graphique est principalement représentée par une petite ligne entre un intervalle de temps et une spécification d'occurrence.



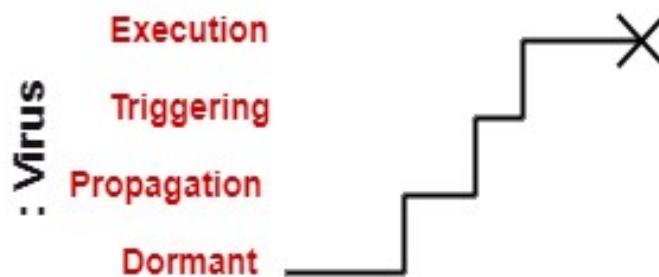
Une personne doit se réveiller entre 5h40 et 6h

Diagramme de temps

Les composants majeurs d'un diagramme de temps

5. L'occurrence de destruction :

- Une occurrence de message qui détruit le participant individuel et représente la fin de la ligne de vie de ce participant.
- Il peut par la suite détruire d'autres objets appartenant à la composition de cet objet, de sorte que rien ne se produise après l'événement de destruction sur une ligne de vie donnée.
- Il est représenté par une croix à la fin d'une chronologie.



La ligne de vie du virus est interrompue

Diagramme de temps

Exemple: Les étapes de la croissance humaine. En conséquence, il n'a qu'une seule ligne de vie.

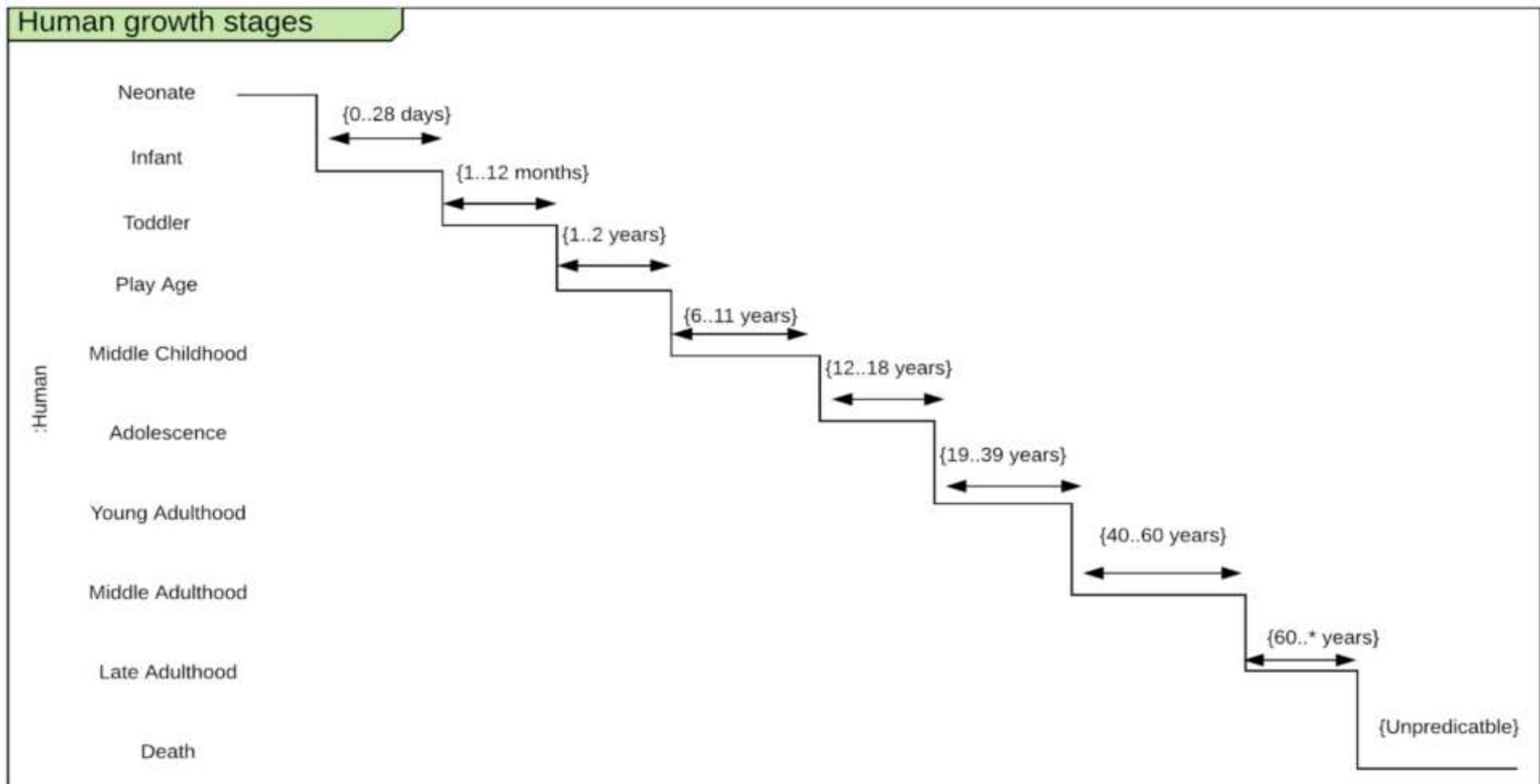


Diagramme de temps

Exercice d'application

1. **Aucune déficience, état normal:** La mémoire et les capacités cognitives semblent normales.
2. **Oubli à l'âge normal:** On le voit surtout chez les personnes âgées de 65 ans qui éprouvent des plaintes subjectives de difficultés cognitives et/ou fonctionnelles, ce qui signifie qu'elles ont du mal à se souvenir du nom et des 5 à 10 dernières années d'histoire.
3. **Confusion précoce, troubles cognitifs légers:** Cela pose un problème dans la récupération des mots, la planification , l'organisation, les objets égarés ainsi que l'oubli d'un nouvel apprentissage, qui à son tour affecte l'environnement.
4. **Confusion tardive, Alzheimer léger:** En cela, une personne oublie les événements et les conversations les plus récents. La personne se souvient de lui-même et de sa famille, mais rencontre des problèmes lors de l'exécution de tâches séquentielles telles que cuisiner, conduire, etc. Sa durée est d'environ deux ans,

Diagramme de temps

Exercice d'application

5. Démence précoce, Alzheimer modéré : En cela, la personne ne peut pas gérer de manière autonome. Il a du mal à se souvenir des détails passés et des coordonnées. Il dure environ 1,5 ans

6. Démence moyenne, Alzheimer modérément sévère: Cela conduit à une prise de conscience insuffisante des événements actuels et la personne est incapable de se souvenir du passé. Cela entraîne une incapacité chez les personnes à prendre un bain et à s'habiller de manière indépendante. Il dure environ 2,5 ans environ

7. Démence tardive ou grave, retard de croissance C'est une capacité intellectuelle sévèrement limitée. En cela, une personne communique soit par des mots courts, soit par des cris, ce qui entraîne une détérioration de la santé en fermant le système corporel. Sa durée est de 1 à 2,5 ans

Diagramme de temps

Diagramme de temps