

Centre universitaire
Abdelhafid Boussouf
Mila

Faculté des sciences et de la
technologie

Département math et
informatique



Génie logiciel

Chapitre 5

Diagrammes UML: Vue dynamique

Mme. S.HEDJAZ

V. Diagrammes UML: Vue dynamique



01

Diagramme d'interaction (séquence)

02

Diagramme d'interaction (communication)

03

Diagramme d'activité

04

Diagramme d'état transition



Diagramme d'interaction (Séquence)

1- Objectif:

- Le diagramme de séquence fait parties des diagrammes dynamique et plus précisément des **diagrammes d'interactions**
- Il permet de représenter des **échanges** entre les différents **objets et acteurs** du système en fonction du **temps** (représentation temporelle)
- Le diagramme de séquence décrit les interactions entre un groupe d'objets en montrant, de façon séquentielle, **les envois de message** qui interviennent entre les objets. Le diagramme peut également montrer **les flux de données** échangées lors des envois de message



Diagramme d'interaction (Séquence)

2- Éléments de base:

- Objets (instances)
- Acteurs
- Messages (cas d'utilisation, appels d'opération)

Représentation graphique de la chronologie des échanges de messages avec le système ou au sein du système

- ✓ « Vie » de chaque objet représentée verticalement
- ✓ Échanges de messages représentés horizontalement



Diagramme d'interaction (Séquence)

2- a- Objet:

- Dans un diagramme de séquence, l'objet à la même représentation que dans le diagramme d'objets. C'est-à-dire un rectangle dans lequel figure le nom de l'objet

objet1 : Classe1

2- b- Acteur:

- Les diagrammes de séquences représentant les échanges entre les objets mais aussi les échanges avec **les acteurs**, nous trouverons aussi la représentation du **stick man** (qui peut être considéré comme un objet).



: Acteur



Diagramme d'interaction (Séquence)

2- c- Ligne de vie:

- A chaque objet est associé une ligne de vie (en trait pointillés à la verticale de l'objet) qui peut être considéré comme un axe temporel (le temps s'écoule du haut vers le bas)
- La ligne de vie indique les périodes d'activité de l'objet (généralement, les moments où l'objet exécute une de ces méthodes).
- Lorsque l'objet est détruit, la ligne de vie s'achève par un croix.

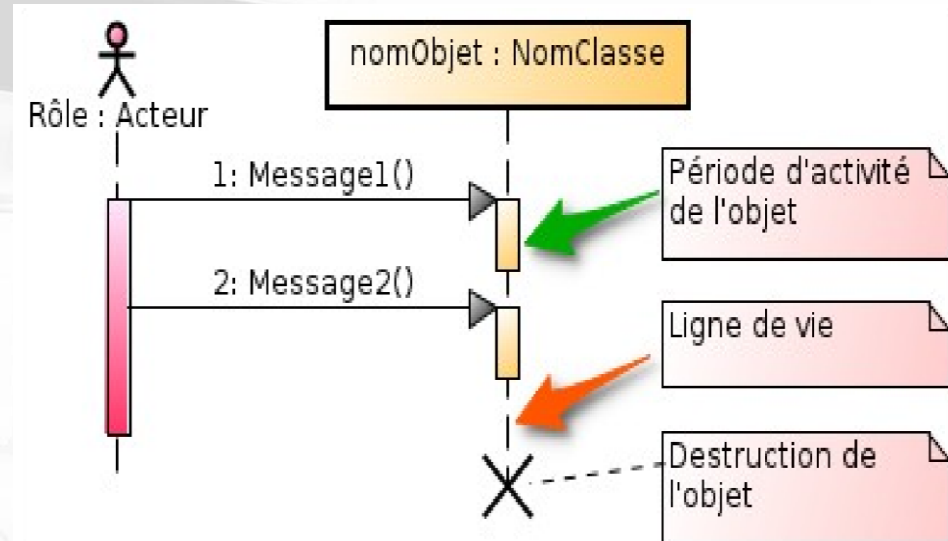




Diagramme d'interaction (Séquence)

2- d- Les messages:

- **Un message** définit une **communication** particulière entre des lignes de vie. Ainsi, un message est une communication d'un **objet** vers un **autre objet**.
- La réception d'un message est considérée par l'objet récepteur comme un événement qu'il faut traiter (ou pas).
- Plusieurs types de messages existent, les plus communs sont :
 - ✓ L'invocation d'une opération : **message synchrone** (appel d'une méthode de l'objet cible).
 - ✓ L'envoi d'un signal : **message asynchrone** (typiquement utilisé pour la gestion événementielle).
 - ✓ **La création** ou la **destruction d'une instance** de classe au cours du cycle principal.



Diagramme d'interaction (Séquence)

2- d- 1- Les messages synchrones:

- La réception d'un message synchrone doit provoquer chez le **destinataire** le **lancement d'une de ses méthodes** (qui souvent porte le même nom que le message).
- **L'expéditeur** du message **reste bloqué** pendant toute l'exécution de la méthode et **attend donc la fin** de celle-ci avant de pouvoir lancer un nouveau message. C'est le message le plus fréquemment utilisé.
- Représenté par une flèche avec un **triangle plein** à son extrémité

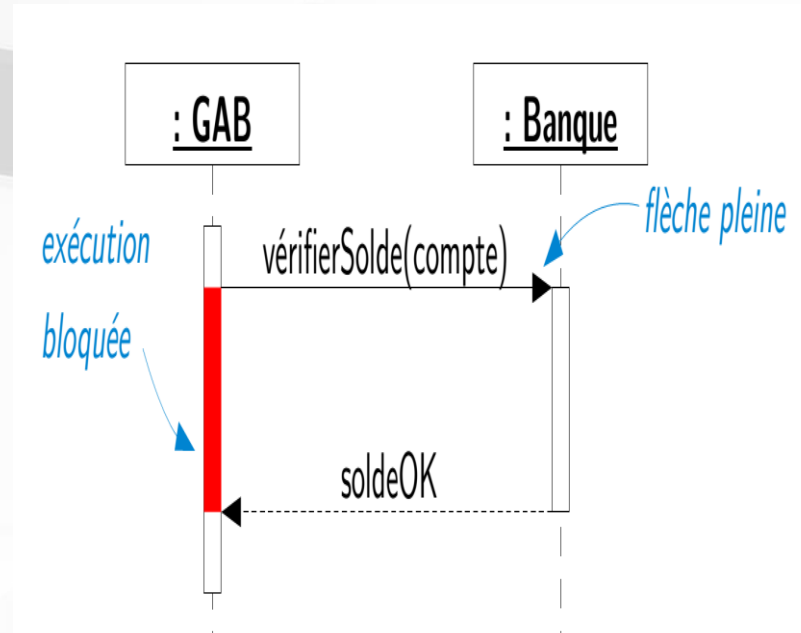




Diagramme d'interaction (Séquence)

2- d- 2- Les messages asynchrone:

- Dans le cas d'un message **asynchrone**, l'expéditeur **n'attend pas la fin de l'activation** de la méthode invoquée chez le destinataire.
- Un message asynchrone représenté par une **flèche simple**

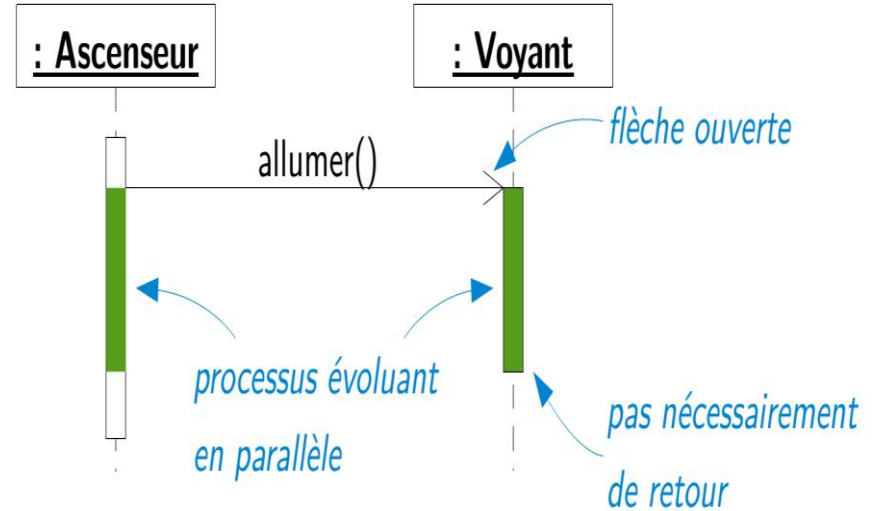




Diagramme d'interaction (Séquence)

2- d- 3- Les messages de retours:

- Si une méthode qui a été activée (par un message) doit retourner des valeurs à la fin de son activation, cela se fait par **un message retour**.
- Le **message de retour** n'est donc pas un appel de méthode (il ne provoque donc pas l'activation d'un objet)
- Le message retour porte souvent **le nom de l'élément retourné**.
- Le message de retour représenté par une **simple flèche en pointillés**

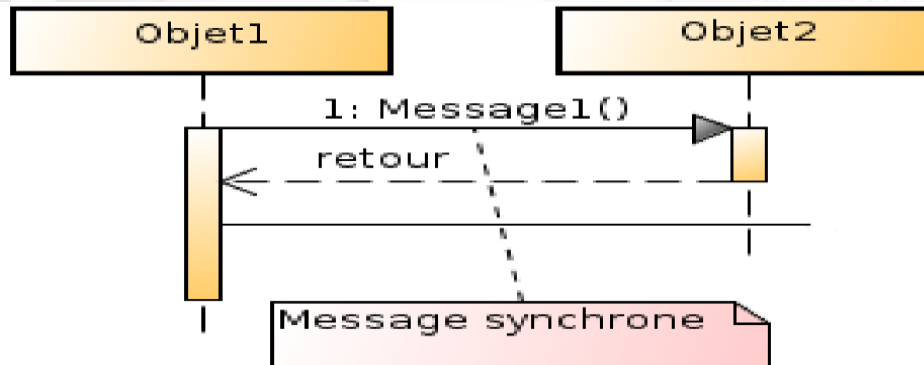




Diagramme d'interaction (Séquence)

2- d- 4- Les messages de création et de destruction:

- La **création d'un objet** est matérialisée par un message spécifique, **appel d'un constructeur**, généralement accompagné du stéréotype « **create** » qui pointe sur le début (le sommet) de la ligne de vie de l'objet créé
- La **destruction d'un objet** est représentée par une **croix à la fin de sa ligne de vie**. Souvent l'objet est détruit suite à la réception d'un message mais ce n'est pas obligatoire. Dans ce cas là, il porte le stéréotype « **destroy** ».

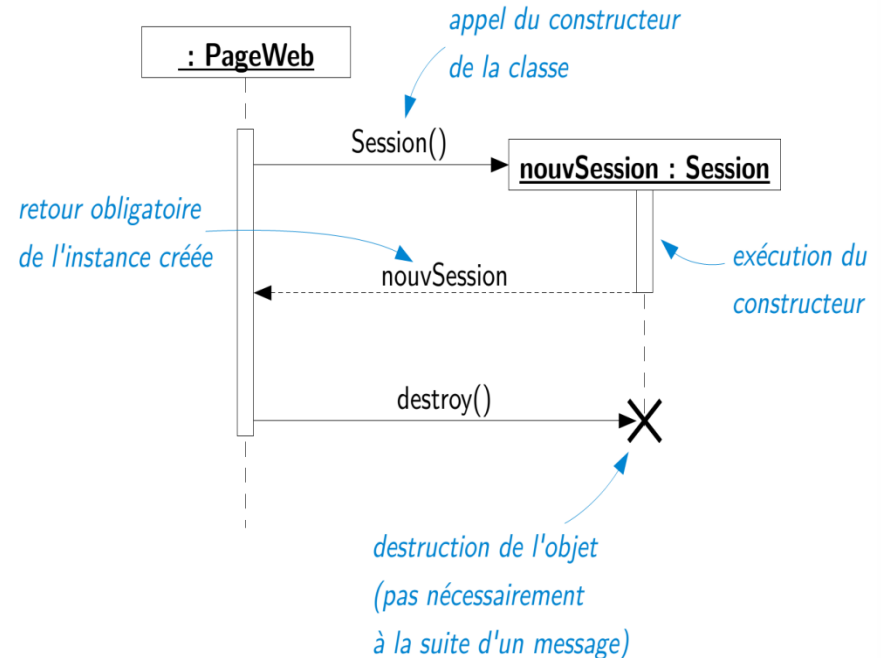




Diagramme d'interaction (Séquence)

2- e- Formalisme des message synchrones et asynchrones:

- **Son nom** (qui est le nom de la méthode appelée ou du signal envoyé). Nous pouvons lui adjoindre facultativement :
 - ✓ Une **numérotation** devant le nom message (séparé du nom du message par 2 point " : ")
 - ✓ Les **paramètres** passés à la méthode ou au signal (entre parenthèses après le nom du message)

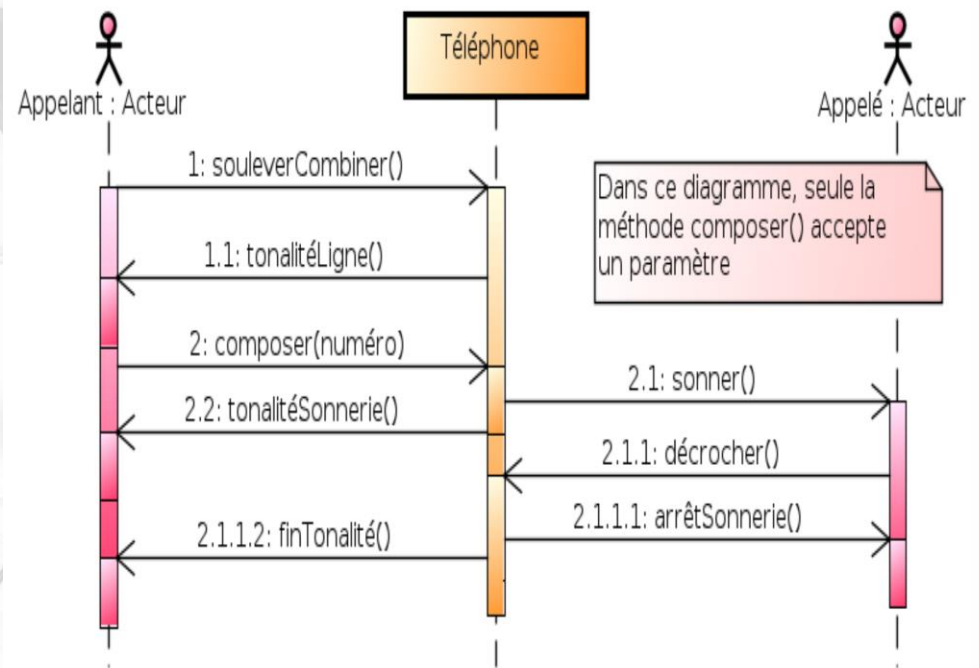




Diagramme d'interaction (Séquence)

2- f- Messages récursifs:

- Un **objet peut s'envoyer un message à lui-même** (utilisation d'une autre méthode du même objet).
- Cela se représente là aussi par un **dédoublage de la bande d'activation**.

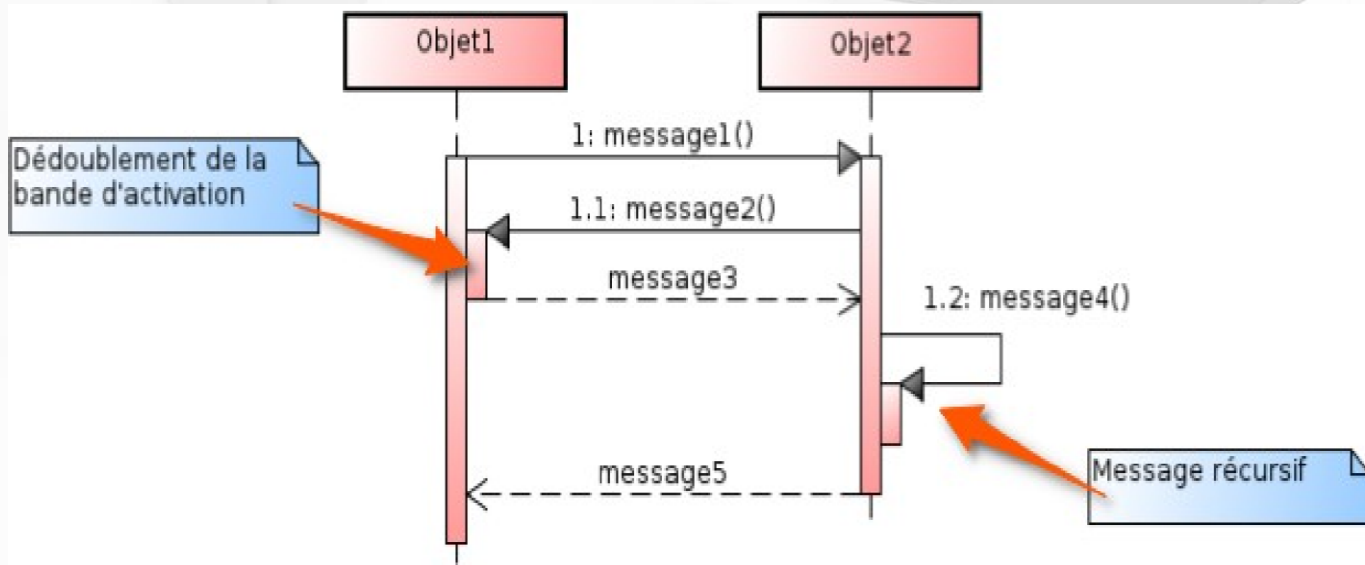




Diagramme d'interaction (Séquence)

3- Cadre d'interaction:

- Un **cadre d'interaction** est une partie du diagramme de séquence associé à une **étiquette**.
- Elle contient un **opérateur** qui en détermine la modalité d'exécution. Les principales modalités sont le **branchement conditionnel** et la **boucle**.

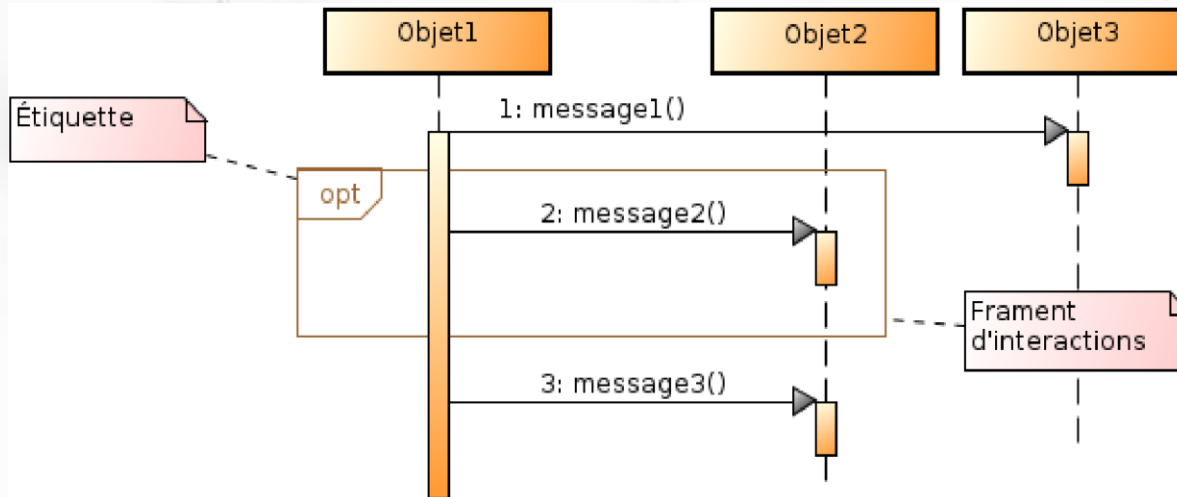




Diagramme d'interaction (Séquence)

3-a- Cadre d'interaction avec « opt » :

- L'opérateur option (**opt**) comporte un opérande et une **condition de garde associée**. Le sous fragment s'exécute **si la condition de garde est vraie** et ne s'exécute pas dans le **cas contraire**.

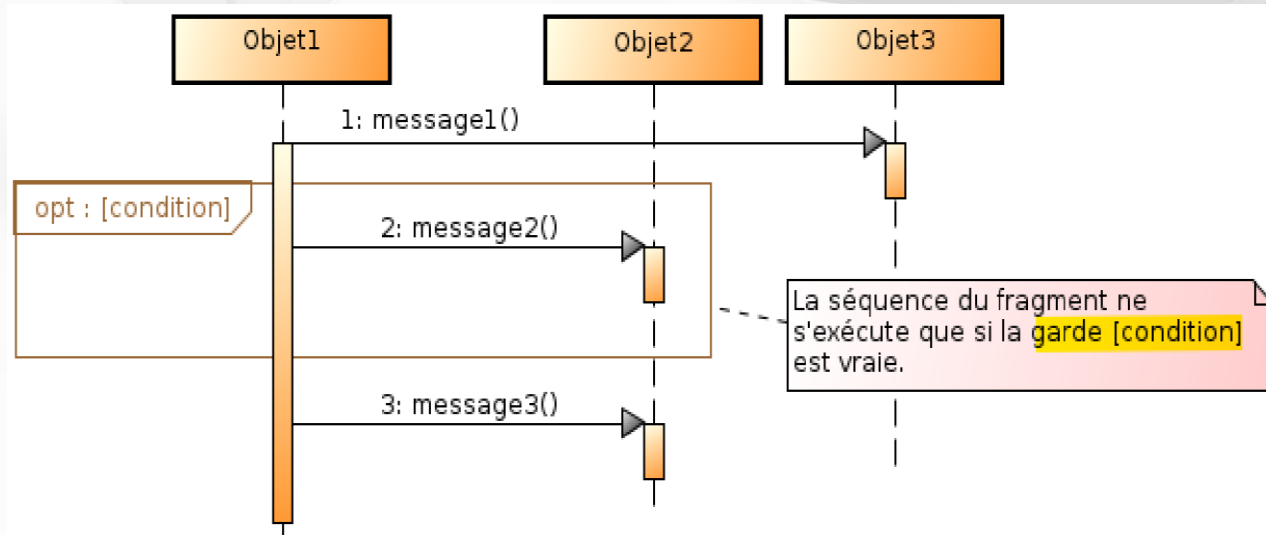




Diagramme d'interaction (Séquence)

3-b- Cadre d'interaction avec « alt » :

- L'opérateur alternatives (alt) est un opérateur conditionnel possédant plusieurs opérandes séparés par des pointillés. C'est l'équivalent d'une **exécution à choix multiples**.
- Chaque opérande détient une condition de garde. **Seul le sous-fragment** dont la condition est **vraie est exécuté**.
- La condition **else** est exécutée que si aucune autre condition n'est valide.

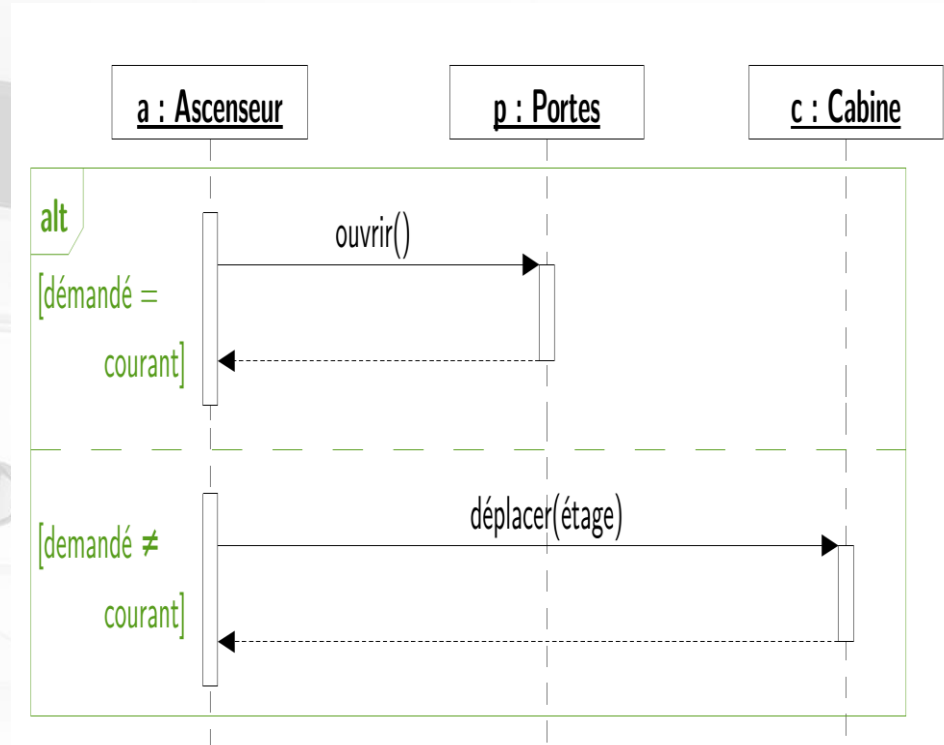
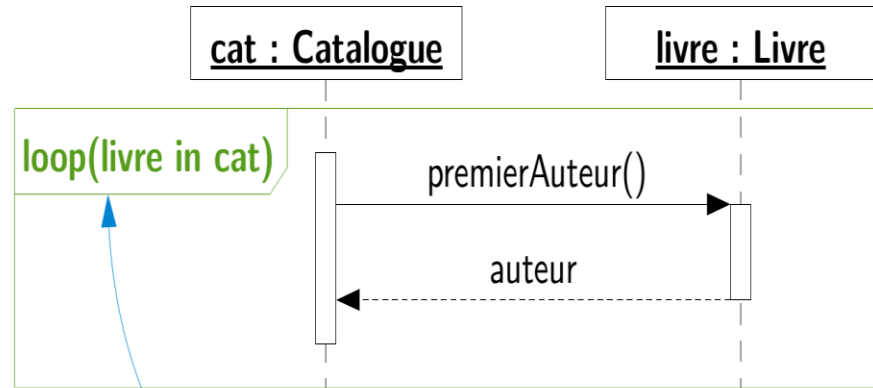




Diagramme d'interaction (Séquence)

3-c- Cadre d'interaction avec « loop » :

- La boucle est réalisée par l'opérateur de boucle (loop) suivi des paramètres min, max et d'une condition de test
- Le contenu du cadre est exécuté min fois, puis tant que la condition de test est vérifiée et tant que le nombre maximal d'exécutions de la boucle ne dépasse pas max.
- Chaque paramètre est optionnel



*objets sur lesquels
répéter la boucle*



Diagramme d'interaction (Séquence)

3-d- Cadre d'interaction avec « ref »:

- L'opérateur de boucle (ref) permet de faire appel à un autre diagramme de séquence.

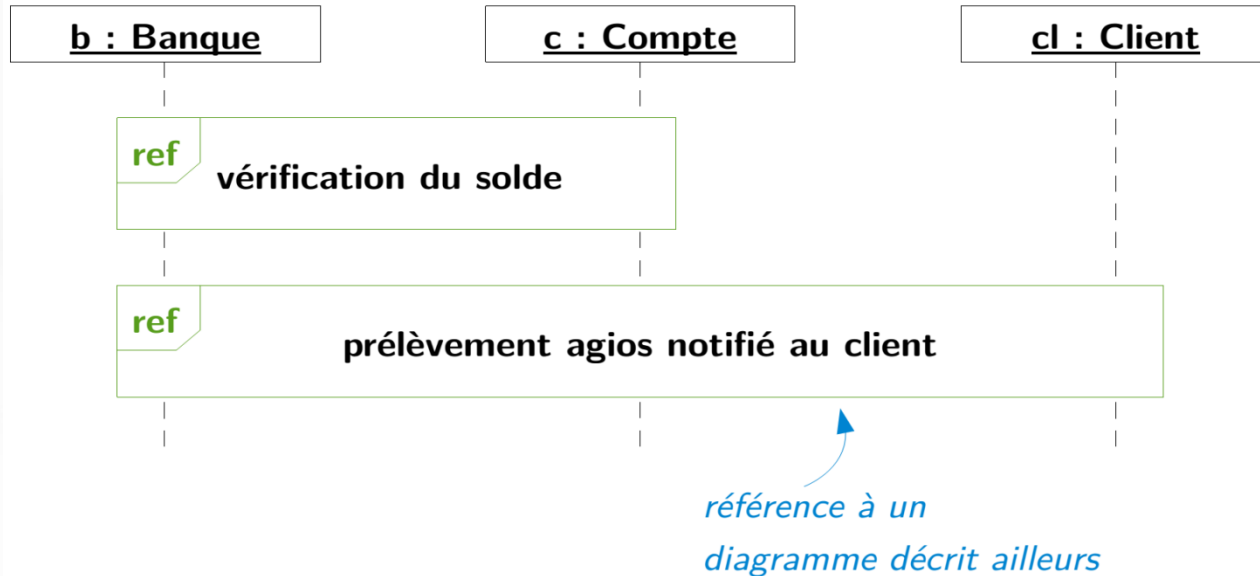




Diagramme d'interaction (Séquence)

4- Exemple: diagramme de cas d'utilisation et diagramme de classe

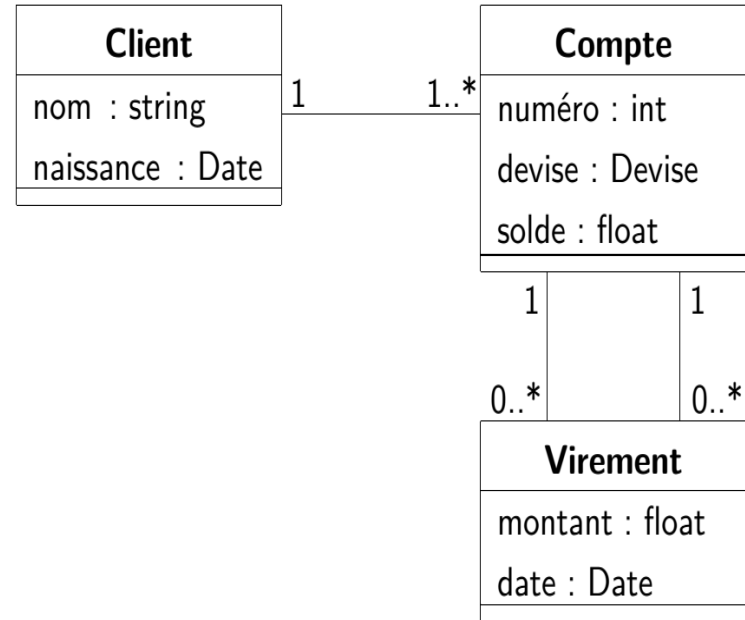
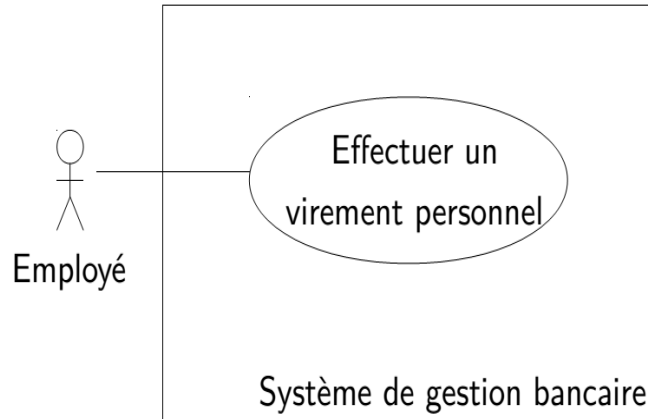




Diagramme d'interaction (Séquence)

4- Exemple: diagramme de séquence correspondant

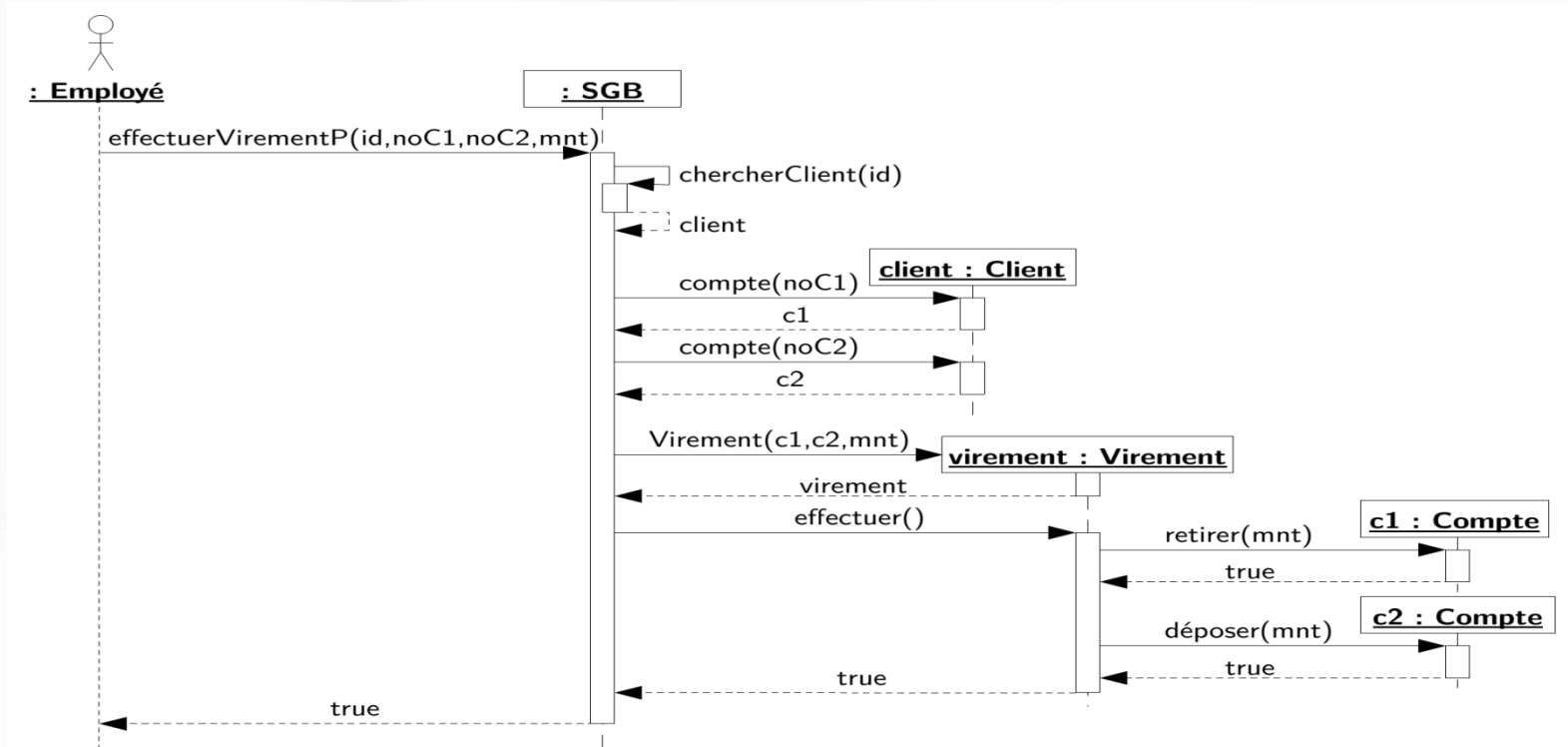




Diagramme d'interaction (Séquence)

4- Exemple: diagramme de classe finale

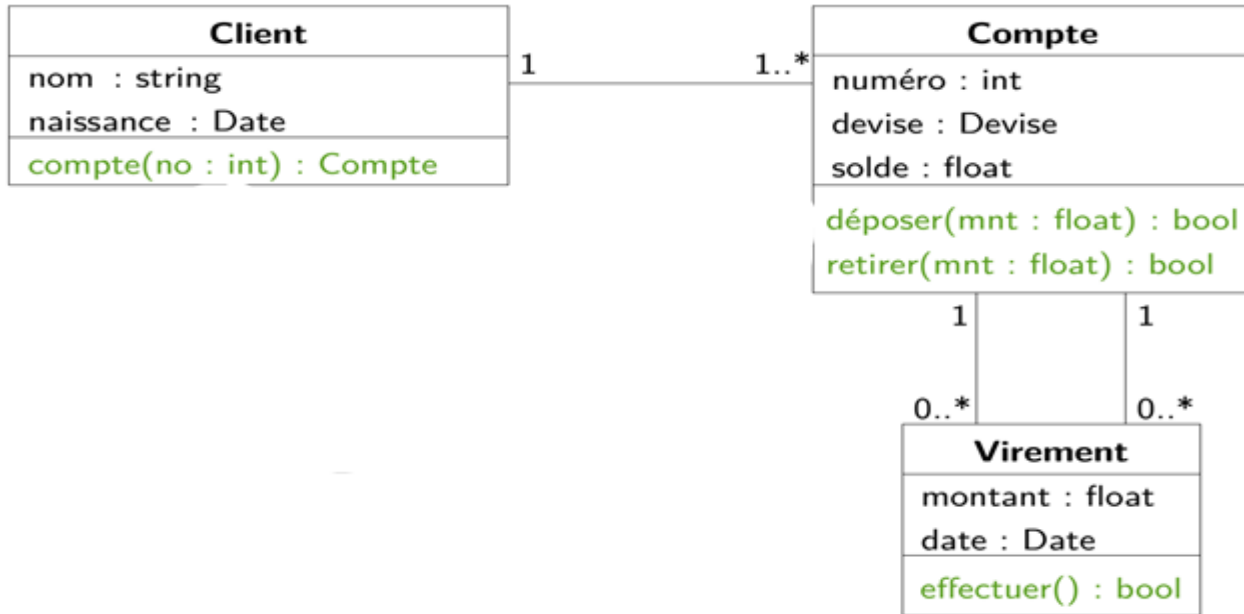




Diagramme d'interaction (Communication)

1- Objectif:

- Un **diagramme de communication**, anciennement appelé un **diagramme de collaboration**, est un schéma d'interaction. Il affiche des informations similaires à des diagrammes de séquence, mais son **objectif principal est de relations d'objet**.
- Le **diagramme de communications** représente les **interactions entre objets** en mettant moins en évidence l'aspect temporel mais en faisant ressortir les relations entre objets, ce modèle montre **les différents messages** qui se propagent **d'un objet à l'autre**
- Les **envois de messages** sont placés le long des liens inter objets. Les messages sont **obligatoirement numérotés**, la numérotation composée étudiée dans le cadre des diagrammes de séquence pouvant également être employée



Diagramme d'interaction (Communication)

2- Éléments de base:

- Objets (instances)
 - Liens d'interactions
 - Messages
-
- Sur les diagrammes de communication, les objets sont présentés avec des connecteurs d'association entre eux.
 - Les messages sont ajoutés à des associations et des flèches montrent aussi courts pointant dans la direction du flux de messages.
 - La séquence des messages est présentée à travers un système de numérotation.



Diagramme d'interaction (Communication)

2- a- Objets:

- Dans un diagramme de communication, l'objet à la même représentation que dans le diagramme d'objets. C'est-à-dire un rectangle dans lequel figure le nom de l'objet

nomObjet:NomType

:NomType

nomObjet:

2- b- Liens:

- Indique un chemin de communication entre deux objets, sur lequel passent les messages

Objet 1

Objet 2



Diagramme d'interaction (Communication)

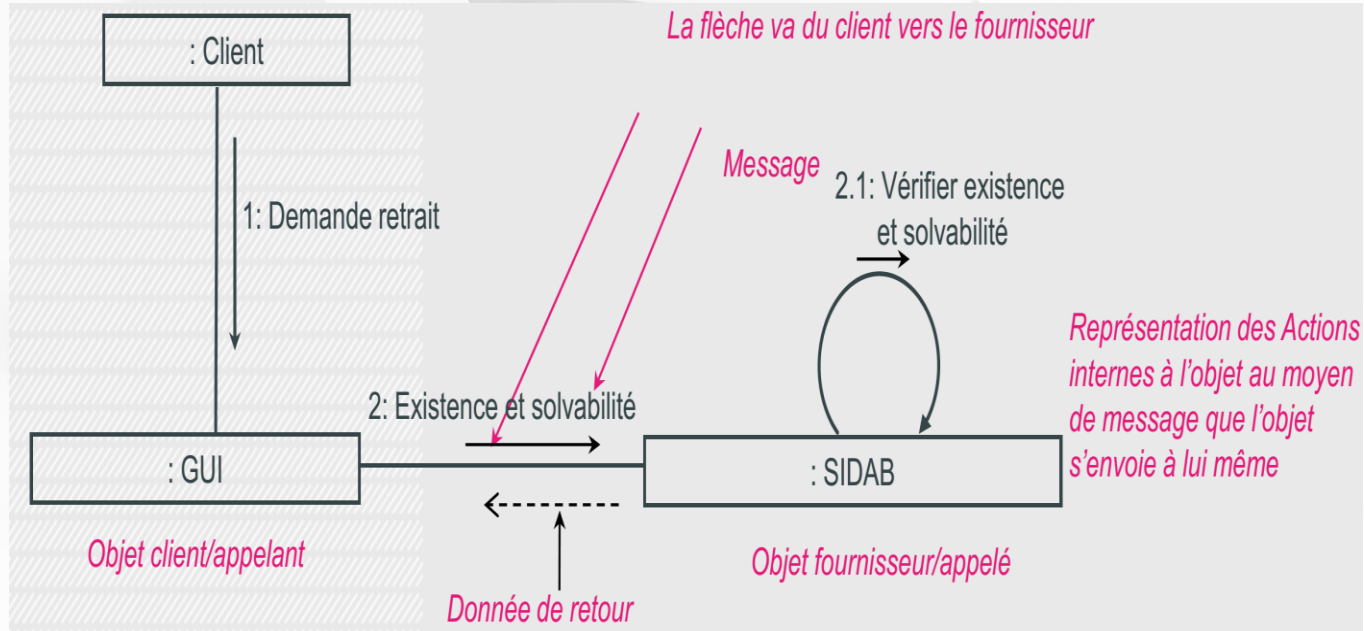
2- c- Messages:

- Dans un diagramme de communication, un message est associé à un lien d'interaction, représenté par une flèche
- **Synchrone** : Un message est envoyé par à un objet à un autre, et le premier objet attends jusqu'à ce que l'action ai finie. Représenté par traits pleins et à l'extrémité pleine
- **Asynchrone** : Un message est envoyé par à un objet à un autre, mais le premier objet n'attends pas la fin de l'action. Représenté par traits pleins et à l'extrémité ouverte
- **Retour** : Le retour explicite d'un objet à qui le message était envoyé. Représenté par une simple flèche en pointillés



Diagramme d'interaction (Communication)

2- c- Messages: exemple:



02

Diagramme d'interaction (Communication)

2- c- Messages:

- Les messages pouvant être accompagné de plusieurs informations:
 - ✓ Ordre: **numérotation**
 - ✓ condition de franchissement: **ajout d'une expression logique** [condition = true]
 - ✓ Boucle: **contrainte de boucle *** [i=0..9]
 - ✓ valeur de retour

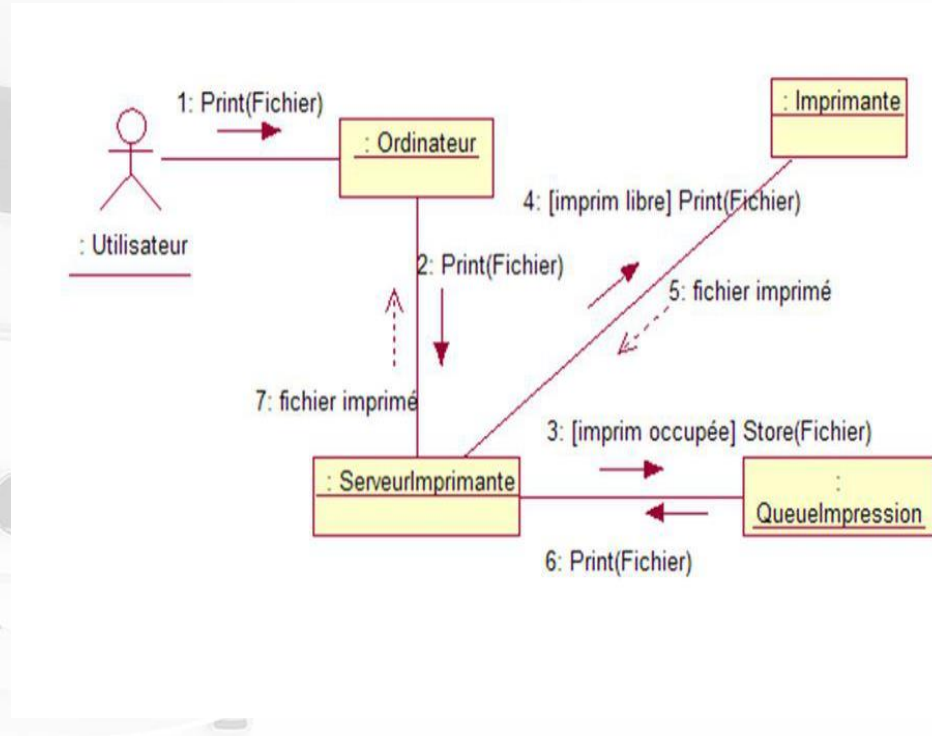




Diagramme d'activité

1- Objectif:

- Le diagramme d'activité est utilisé pour modéliser les aspects dynamiques d'un système. Il s'agit de représenter les opérations d'un processus et leurs conséquences sur les objets (logiciels ou matériels). La modélisation peut être utilisée pour décrire le déroulement d'un cas d'utilisation ou d'une méthode.
- Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Nous pouvons attacher un diagramme d'activités à n'importe quel élément de modélisation afin de visualiser, spécifier, construire ou documenter le comportement de cet élément.
- 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.



Diagramme d'activité

2- Éléments de base:

- **Les activités:** (une activité= une étape d'exécution, état-activité). Une activité représente une exécution d'un mécanisme, un déroulement d'étapes séquentielles. Le passage d'une activité vers une autre est matérialisé par une transition.
- **Les transitions:** qui sont automatiques entre activités, il est inutile également de préciser les événements. Les transitions sont déclenchées par la fin d'une activité et provoquent le début immédiat d'une autre.
- 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.

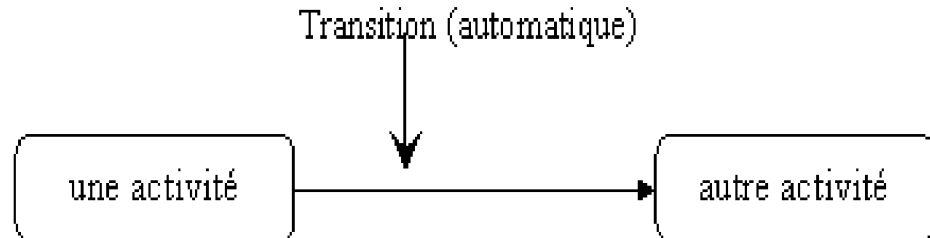




Diagramme d'activité

2- a- 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.
- La notion d'action est à rapprocher de la notion d'instruction élémentaire d'un langage de programmation (comme C++ ou Java). 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 ;
- ...



Diagramme d'activité

2- b- Activité et groupe d'activité:

- Une **activité** définit un comportement décrit par un séquençement 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.
- Une activité est un comportement et à ce titre peut être associée à des paramètres.
- Un groupe d'activités est une **activité regroupant des nœuds et des arcs**. Les nœuds et les arcs peuvent appartenir à **plus d'un groupe**.
- Un diagramme d'activités **est lui-même un groupe d'activités**.



Diagramme d'activité

2- c- Nœuds d'activité:

- 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 ou action
 - ✓ les nœuds objets
 - ✓ et les nœuds de contrôle
- Représentation graphique des nœuds d'activité : de la gauche vers la droite:
 - ✓ Le nœud représentant une action
 - ✓ Un nœud objet,
 - ✓ un nœud de décision ou de fusion,
 - ✓ un nœud de bifurcation ou d'union,
 - ✓ un nœud initial
 - ✓ un nœud final
 - ✓ un nœud final de flot.

Noeud d'action

Noeud d'objet



Noeuds de contrôle



Diagramme d'activité

2- d- Transition:

- Le passage d'une activité vers une autre est matérialisé par une **transition**. Graphiquement les transitions sont représentées par des **flèches en traits pleins** qui **connectent les activités** entre elles.
- 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).





Diagramme d'activité

2- c- 1. Nœud d'actions:

- 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.
- Un nœud d'action doit avoir au moins un arc entrant.
- Graphiquement, un nœud d'action est représenté par un rectangle aux coins arrondis qui contient sa description textuelle.

Etablir les commandes



Diagramme d'activité

2- c- 2. Nœud d'objets:

- Les nœuds d'objet permettent de définir un flot d'objets (flot de données) dans un diagramme d'activités. Chaque nœud représente l'existence d'objet généré par une action dans une activité et utilisé par d'autres actions.
- Graphiquement, un tel nœud d'objet est représenté par un rectangle dans lequel est mentionné le type de l'objet. Des arcs viennent ensuite relier ce nœud d'objet à des activités sources et cibles

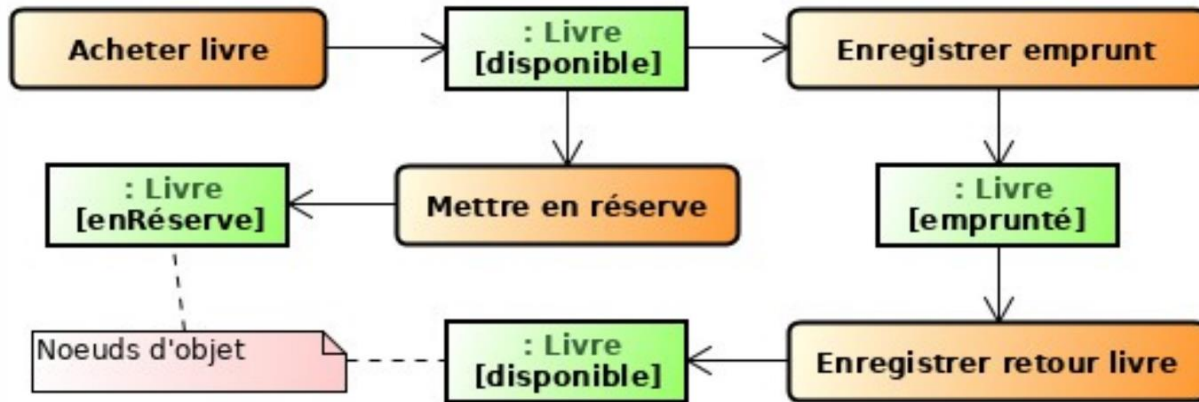




Diagramme d'activité

2- c- 3. Nœud 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 :

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.

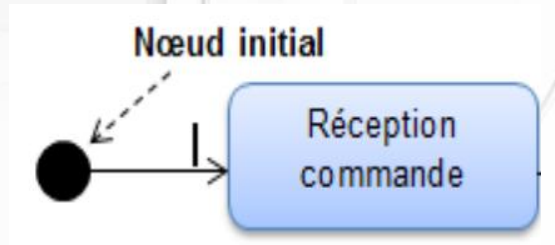




Diagramme d'activité

2. Nœud final:

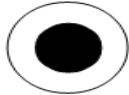

- Un nœud final est un nœud de contrôle possédant un ou plusieurs arcs entrants et aucun arc sortant. Il existe deux types de nœud finaux :
- **Nœud de fin d'activité** : Lorsque l'un des arcs d'un nœud de fin d'activité est activé, l'exécution de l'activité enveloppante s'achève et tout nœud ou flot 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'activité

3. Nœud décision:

- 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 plusieurs arcs sortants. Ces derniers sont généralement accompagnés de conditions de garde pour conditionner le choix.
- Graphiquement, nous représentons un nœud de décision par un losange.

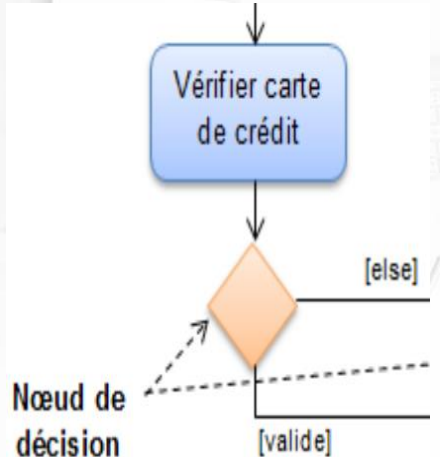




Diagramme d'activité

4. Nœud fusion:

- Un nœud de fusion est un nœud de contrôle qui rassemble plusieurs flots alternatifs entrants en un seul flot sortant. Il n'est pas utilisé pour synchroniser des flots concurrents (c'est le rôle du nœud d'union) mais pour n'accepter un flot parmi plusieurs.
- Graphiquement, nous représentons un nœud de fusion, comme un nœud de décision, par un losange.

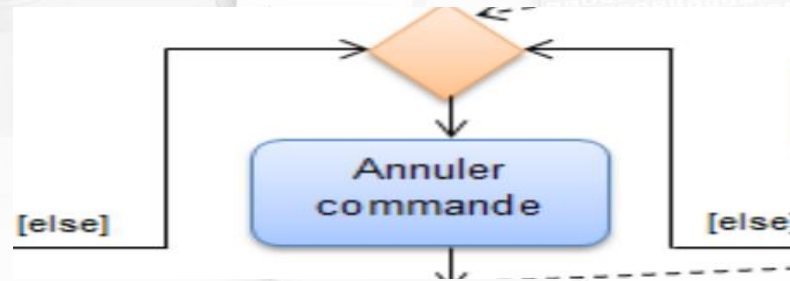




Diagramme d'activité

5. Nœud de bifurcation:

- Un nœud de bifurcation, également appelé nœud de débranchement 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.
- Graphiquement, nous représentons un nœud de bifurcation par un trait plein

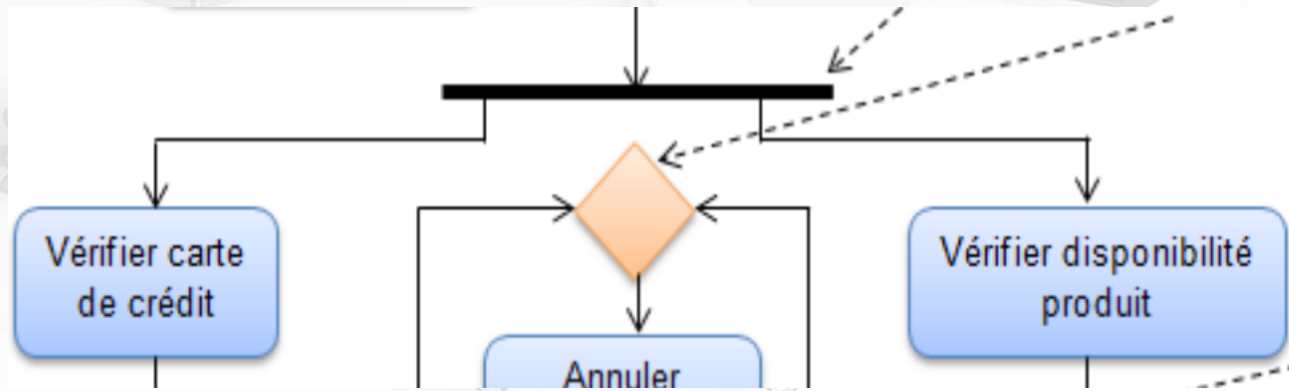




Diagramme d'activité

6. Nœud d'union:

- Un nœud d'union, également appelé nœud de jointure est un nœud de contrôle qui synchronise des flots multiples. Un tel nœud possède donc plusieurs arcs entrants et un seul arc sortant.
- Graphiquement, nous représentons un nœud d'union, comme un nœud de bifurcation, par un trait plein.

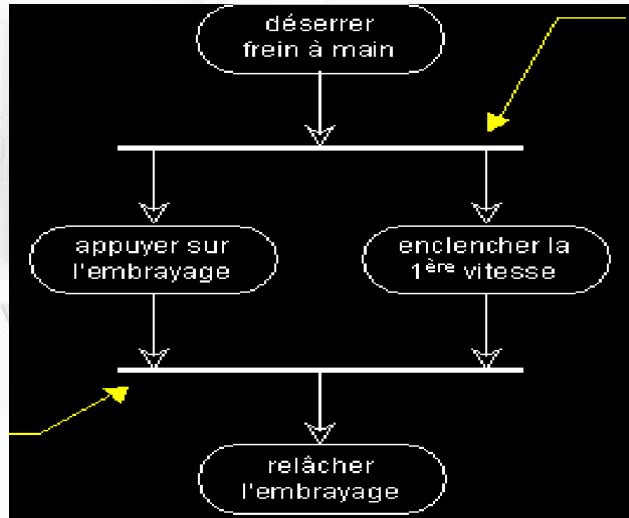




Diagramme d'activité

7. Partitions ou couloirs d'activités:

- Les diagrammes d'activités indiquent ce qui se passe sans préciser qui fait quoi (en terme de programmation, ils ne précisent pas quelle classe est responsable et en terme de processus métier, ils ne précisent pas quelle partie de l'organisation exécute chaque action).
- Il est alors possible de diviser un diagramme d'activités en partitions ou couloirs d'activités. Chaque partition montre ainsi quelles actions sont exécutées par une classe ou par une unité organisationnelle.



Diagramme d'activité

7. Partitions ou couloirs d'activités: exemple

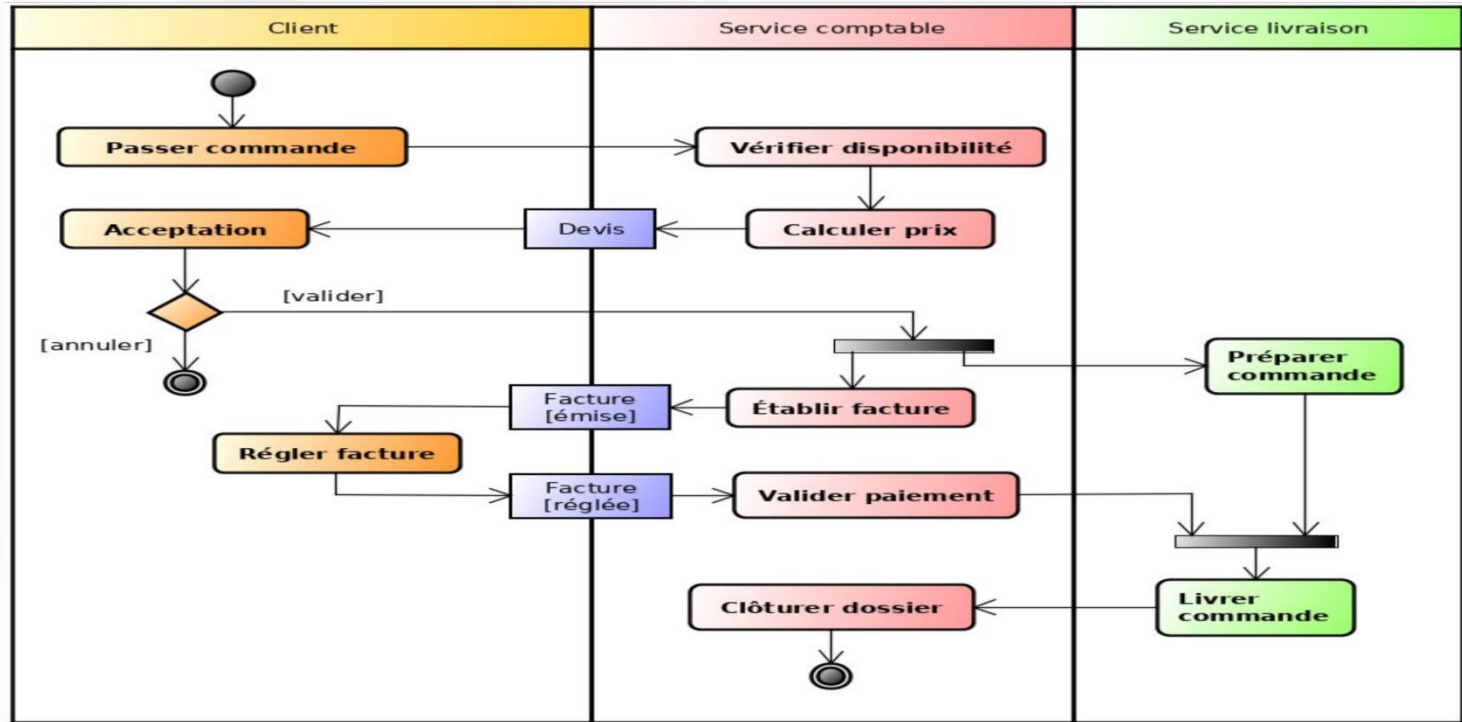




Diagramme d'état transition

1- Définition:

- Décrit le comportement des objets d'une classe au moyen d'un automate d'états associés à la classe
- Il est souhaitable de construire un diagramme d'états-transitions pour chaque classeur (qui, le plus souvent, est une classe) possédant un comportement dynamique important. Un diagramme d'états-transitions ne peut être associé qu'à une seule classe
- Le comportement est modélisé par un graphe :
 - **Nœuds** = états possibles des objets
 - **Arcs** = transitions d'état à état



Diagramme d'état transition

2- Notation graphique:

- Les diagrammes d'états-transitions visualisent des automates d'état finis, du point de vue des états et des transitions.
- Les états sont représentés par des rectangles aux coins arrondis, tandis que les transitions sont représentées par des arcs orientés liant les états entre eux.

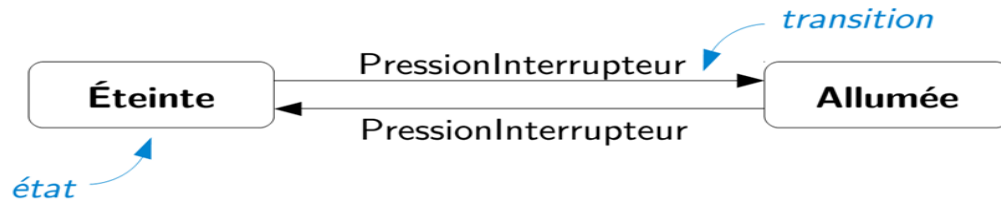
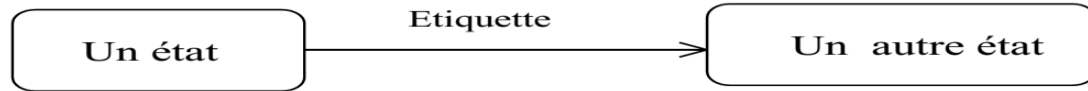




Diagramme d'état transition

3- Etat:

- Un objet peut passer par une série d'états pendant sa durée de vie
- Un état représente une période dans la vie d'un objet pendant laquelle ce dernier attend un événement ou accomplit une activité
- La configuration de l'état global de l'objet est le jeu des états (élémentaires) qui sont actifs à un instant donné
- Le nom de l'état peut être spécifié dans le rectangle et doit être unique dans le diagramme d'états-transitions, ou dans l'état enveloppant



Diagramme d'état transition

3-1. Types état:

1. Etat initial:

- L'état initial est un pseudoétat qui indique l'état de départ, par défaut, lorsque le diagramme d'états-transitions, ou l'état enveloppant, est invoqué
- Lorsqu'un objet est créé, il entre dans l'état initial ●

2. Etat final:

- L'état final est un pseudoétat qui indique que le diagramme d'états-transitions, ou l'état enveloppant, est terminé ●



Diagramme d'état transition

3-1. Types état:

3. Etat intermédiaire:

- Les états peuvent également contenir des actions ; elles sont exécutées à l'entrée ou à la sortie de l'état ou lors de l'occurrence d'un événement pendant que l'objet est dans l'état en question. Comme les classes, les états peuvent être divisés en compartiments qui contiennent :
 - Le nom de l'état ;
 - La liste des actions ou des activités internes effectuées pendant que l'élément est dans cet état.

ÉtatSimple

ÉtatAvecÉvt

event1 [cond1] / action1

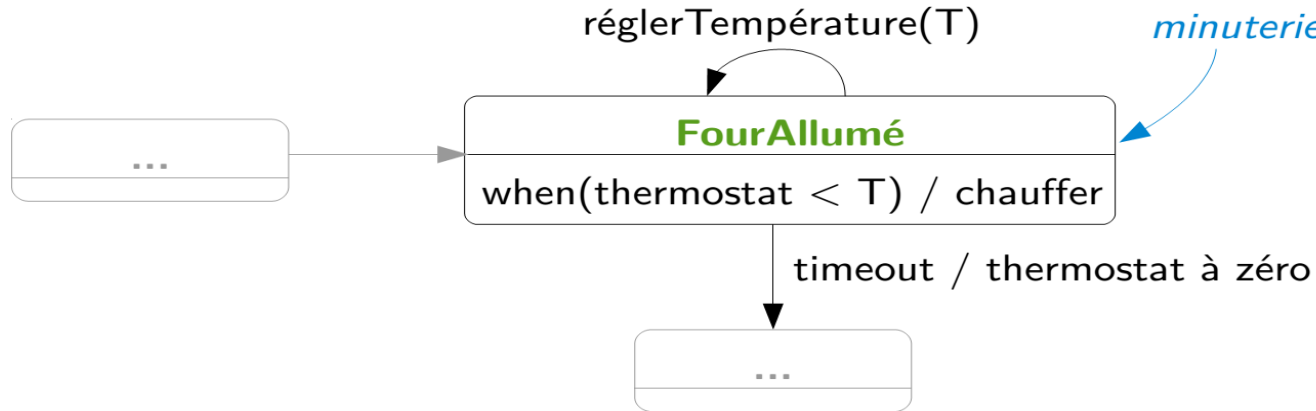
event2 [cond2] / action2



Diagramme d'état transition

3-2. Caractéristiques état:

3.2.1. Conditions:



conditions :
thermostat non nul
minuterie non nulle



Diagramme d'état transition

3-2. Caractéristiques état:

3.2.2. Evènements:

- Fait instantané venant de l'extérieur du système et survenant à un instant donné

Types d'événements :

- **Signal** : réception d'un message asynchrone
- **Appel** d'une opération (synchrone) : liée aux cas d'utilisation, opération du diagramme de classes...
- **Satisfaction d'une condition booléenne** : when(cond), évaluée continuellement jusqu'à ce qu'elle soit vraie
- **Temps**
 - Date relative : when(date = date)
 - Date absolue : after(durée)



Diagramme d'état transition

3-2. Caractéristiques état:

3.2.2. Evènements:

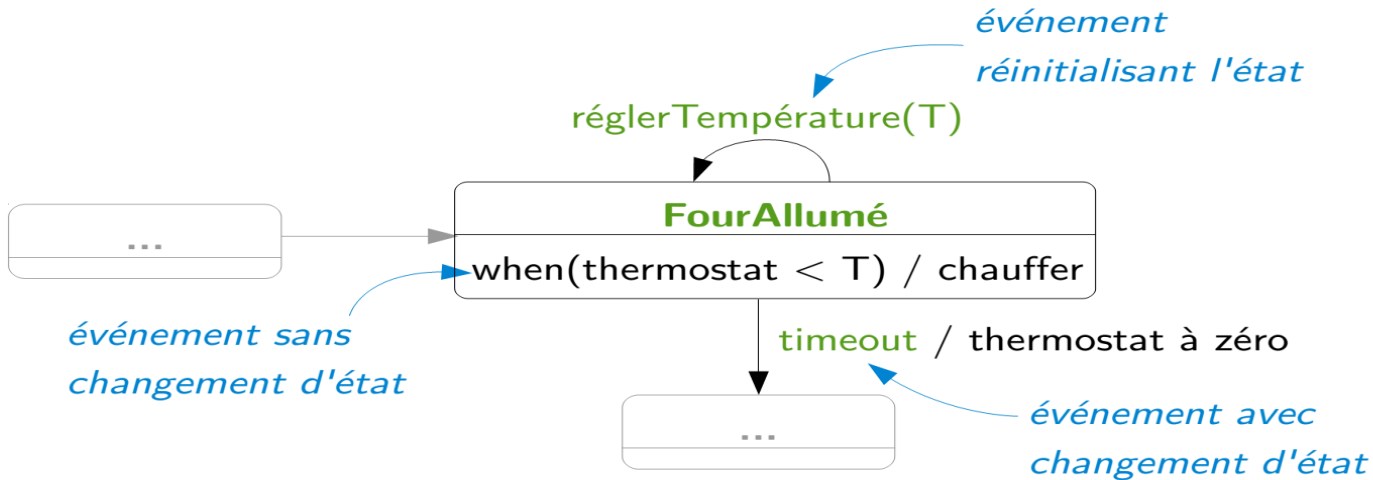




Diagramme d'état transition

3-2. Caractéristiques état:

3.2.3. Action:

- **Action** : Réaction du système à un événement
- **Caractéristiques** : atomique, instantanée, non interruptible
- Exemples d'actions :
 - affectation
 - envoi d'un signal
 - appel d'une opération
 - création ou destruction d'un objet



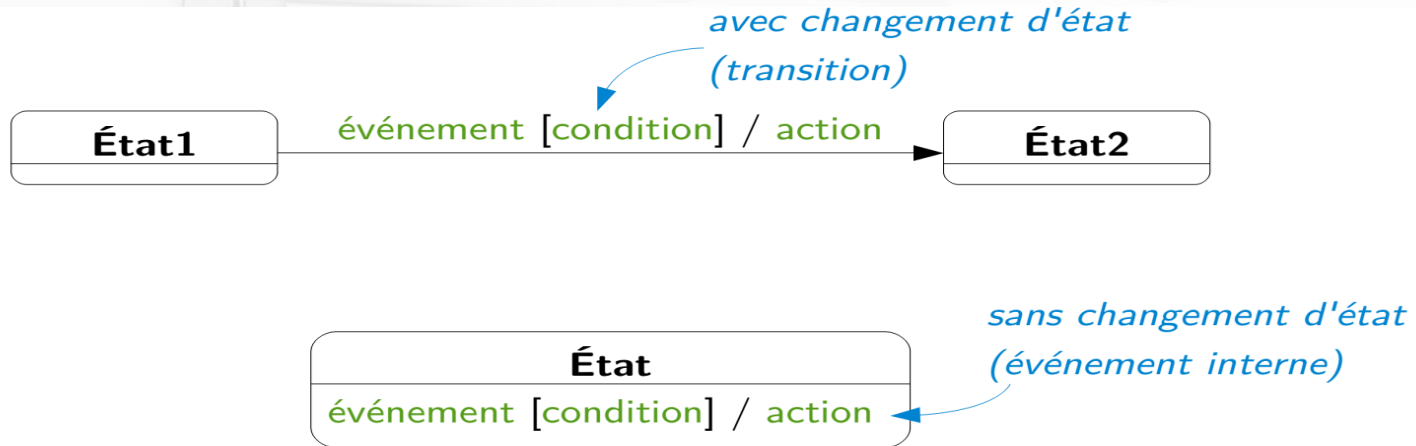
Diagramme d'état transition

3-2. Caractéristiques état:

3.2.3. Action:

événement [condition] / action

Lorsque l'événement se produit, si la condition est vérifiée, alors l'action est effectuée



03

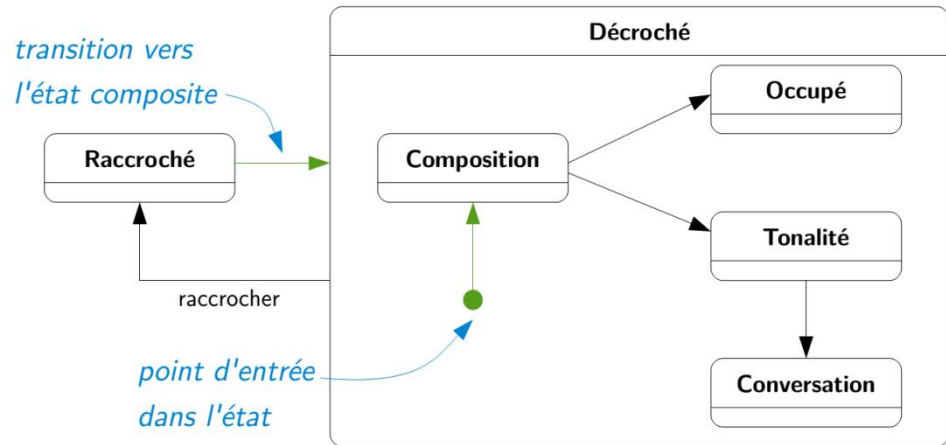
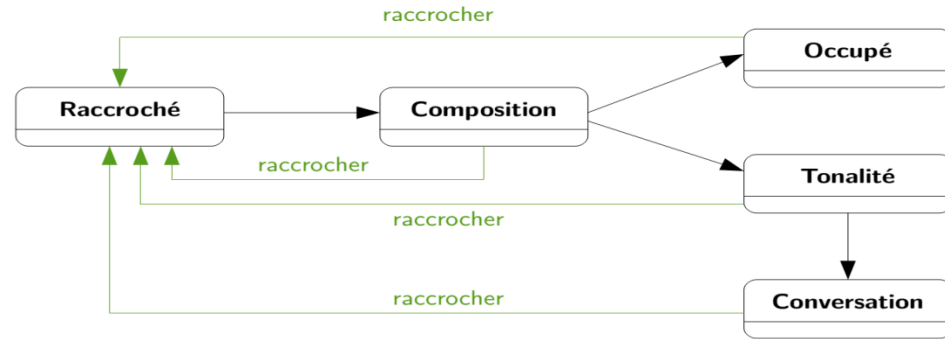
Diagramme d'état transition

3.3. Etat composite:

État regroupant un ensemble d'états

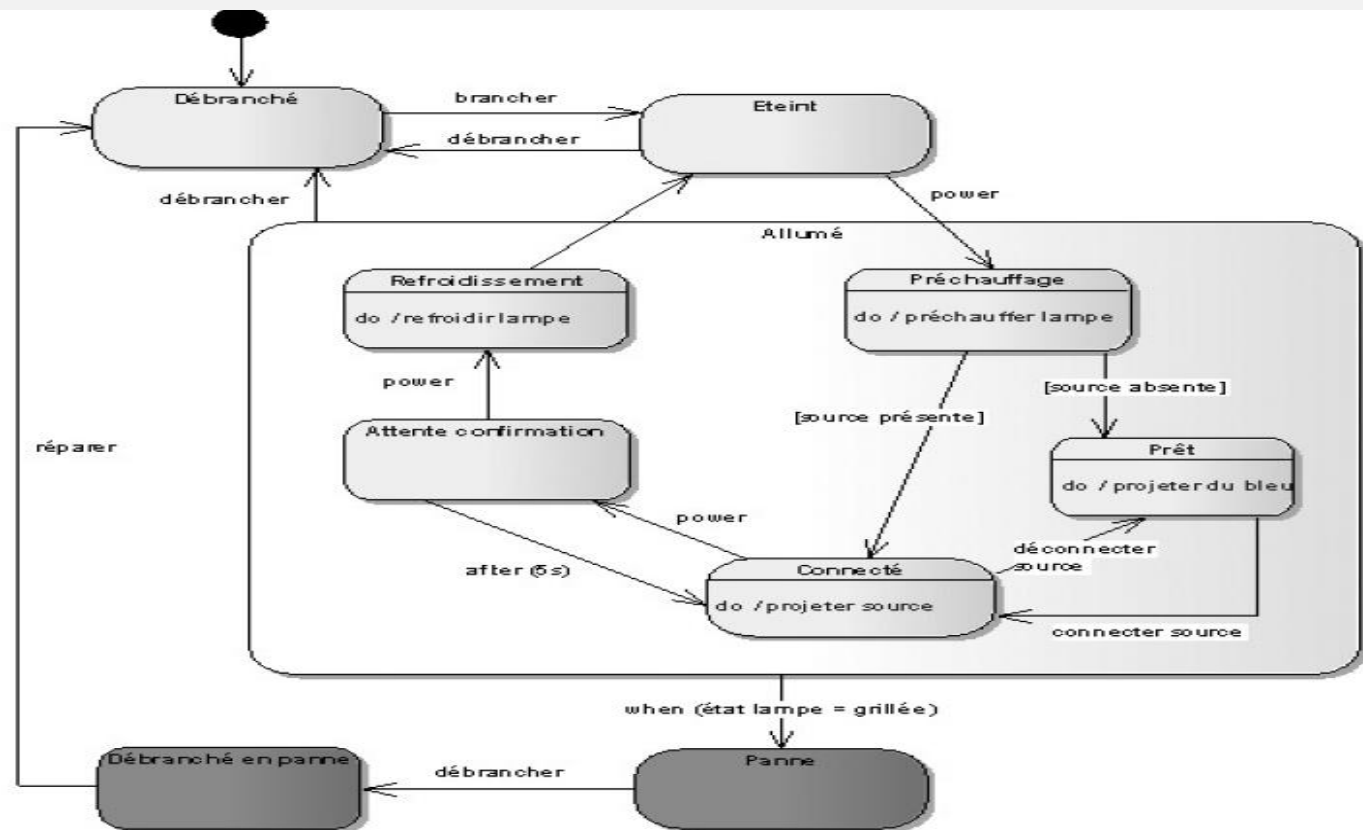
Objectifs :

- Hiérarchiser les états
- Structurer les comportements complexes
- Factoriser les actions



03

Diagramme d'état transition



Bibliographies

- **Uml 2 pratique de la modélisation**, Benoît Charroux, Yann Thierry-Mieg, Aomar Osmani
Ni <https://fr.slideshare.net/nassimamine3994/uml-2-pratique-de-la-modlisation>
- **Uml 2 par la pratique**, Pascal roques
- **Les cahiers du programmeur**, Pascal roques
- **Uml en action**, Pascal roques
-