

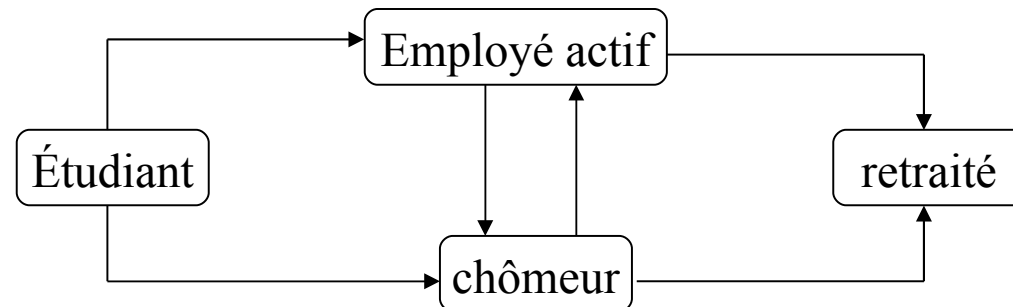
Diagramme d'états (DET)

Dame Samb
UCAD/FST/DMI



Diagramme d'états (DET): Objectifs

- ✧ Un DET représente le cycle de vie des instances d'une classe.
- ✧ Il décrit tous les états possibles d'un objet, les transitions possibles entre ces états et les actions associées aux transitions.
- ✧ Un DET doit être attaché à tous les objets qui ont des états clairs et identifiables ainsi qu'un comportement complexe.



États

- ✦ Un état représente une période dans la vie d'un objet pendant laquelle, ce dernier attend un événement ou accomplit une activité.
- ✦ Un état dépend des états précédents et des événements survenus.
- ✦ Un état est représenté par un rectangle dont les coins sont arrondis.
- ✦ Le nom de l'état peut être spécifié dans le rectangle et doit être unique dans le DET ou dans l'état enveloppant.
- ✦ Certains états, dits **composites**, peuvent contenir des **sous états**.

États

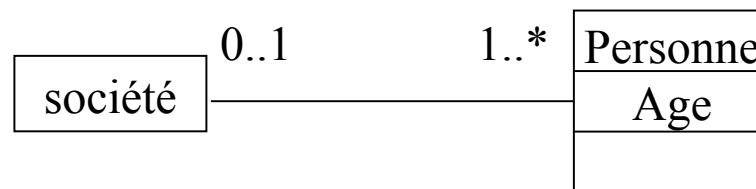
- ✦ Un état peut être partitionné en plusieurs compartiments séparés par une ligne horizontale :
 - le premier compartiment contient le nom de l'état ;
 - les autres peuvent recevoir des transitions internes ;
 - ou des sous états
- ✦ **État initial**: indique l'état de départ, par défaut. Lorsqu'un objet est créé, il entre dans cet état. Dans un DET, on ne doit avoir qu'un seul état initial
- ✦ **État final**: indique que le DET est terminé. Dans un DET, on peut avoir plusieurs états finaux comme on peut ne pas avoir d'état final (cas des systèmes qui ne s'arrêtent pas)

● État initial

⦿ État final

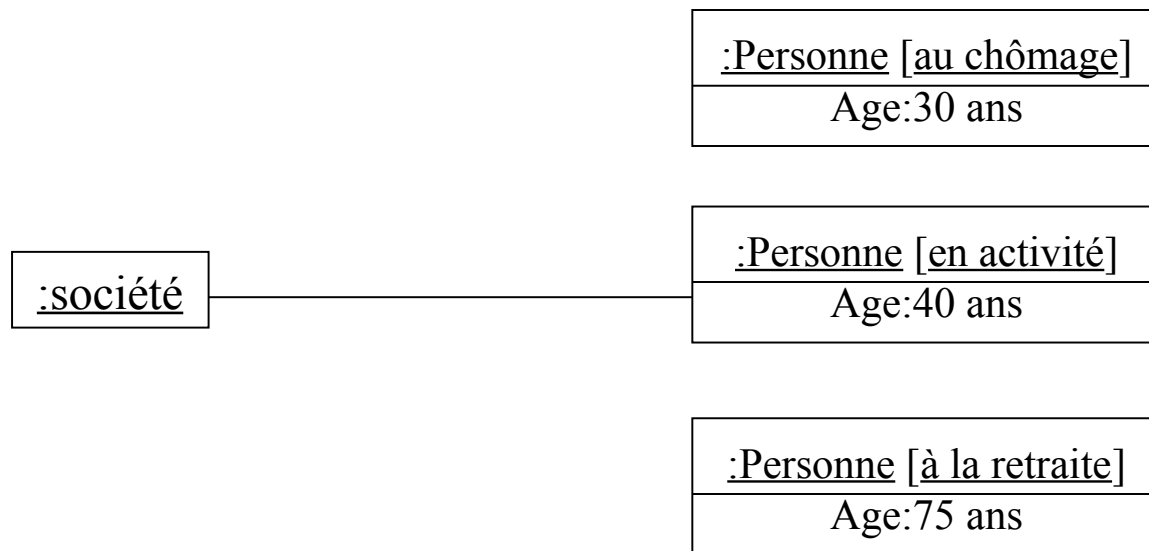
État

- ✦ L'état d'un objet est toujours l'image de la conjonction instantanée des valeurs de ses attributs et de la présence ou non des liens entre l'objet considéré et les autres objets.
- ✦ Exemple: Pour connaître les états de Personne, on doit étudier son âge et la présence éventuelle d'un lien avec une société



État

✦ Les états possibles d'un objet personne sont "*en activité*", "*au chômage*" et "*à la retraite*".



Événements

- ✦ Un événement correspond à un stimulus auquel l'objet doit répondre. Il est par nature instantané (dépourvu de durée) et doit être traité sans délai.
- ✦ Quand un événement est reçu, une transition peut être déclenchée et faire basculer l'objet dans un nouvel état.
- ✦ Types d'événements :
 - signal,
 - appel,
 - Changement,
 - et temporel

Événements

✦ Événement de type signal

- Un signal véhicule une communication asynchrone à sens unique entre deux objets.
- L'objet expéditeur n'attend pas que le destinataire traite le signal pour poursuivre son déroulement.
- La réception d'un signal est un événement pour l'objet destinataire.
- Un signal est déclaré dans un classeur stéréotypé «signal», ne fournissant pas d'opération et dont les attributs sont interprétés comme des arguments.
- Syntaxe :
`<nom_événement> ([<paramètre> : <type> [;<paramètre> : <type> ...]])`

Événements

✦ Événement de type appel

- Un événement de type appel représente la réception de l'appel d'une opération par un objet.
- La syntaxe d'un événement de type appel est la même que celle d'un signal.
- Les événements de type appel sont des méthodes déclarés au niveau des classes.

✦ Événement de type changement

- Un événement de type changement est généré par la satisfaction (*i.e.* passage de faux à vrai) d'une expression booléenne sur des valeurs d'attributs.
- Il s'agit d'une manière déclarative d'attendre qu'une condition soit satisfaite
- Syntaxe :
 - when (<condition_booléenne>)

Événements

✧ Événement temporel

- Les événements temporels sont générés par le passage du temps.
- Ils sont spécifiés
 - soit de **manière absolue** (date précise), en utilisant un événement de changement :
when (date = <date>)
 - soit de **manière relative** (temps écoulé) :
after (<durée>)
- Par défaut, le temps commence à s'écouler dès l'entrée dans l'état courant.

Transition: définition et syntaxe

- ✦ Une transition définit la réponse d'un objet à l'occurrence d'un événement.
- ✦ Elle lie généralement, deux états E1 et E2, et indique que :
 - si un événement déclencheur se produit (<événement>),
 - et que la condition de garde est vérifiée (<garde>),
 - alors l'objet peut quitter l'état E1 et entrer dans l'état E2
 - une fois exécuté certaines activités (<activité>).
- ✦ Syntaxe :
 - [<événement>]['[' <garde> ']']['/' <activité>]



Transition: condition de garde

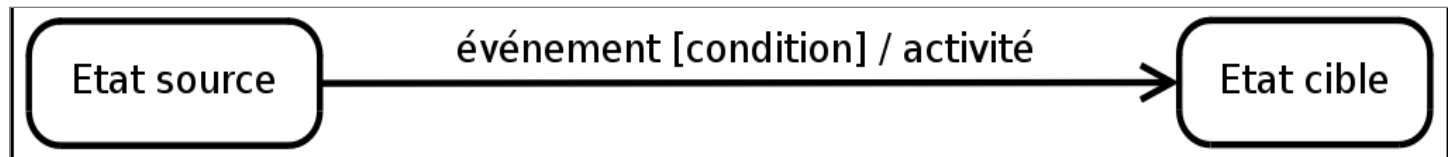
- ✦ C'est une expression logique sur les attributs de l'objet, ainsi que sur les paramètres de l'événement déclencheur conditionnant le tir de la transition.
- ✦ La condition de garde est évaluée uniquement lorsque l'événement déclencheur se produit.
 - Si l'expression est fausse à ce moment là, la transition ne se déclenche pas,
 - si elle est vraie, la transition se déclenche et ses effets se produisent.
- ✦ La condition de garde est différente de la condition dans un événement de type changement qui est évaluée continuellement jusqu'à ce qu'elle devienne vraie.

Effet d'une transition

- ✦ Lorsqu'une transition se déclenche, son effet (spécifié par '/' <activité> dans la syntaxe) s'exécute.
- ✦ Il s'agit généralement d'une activité qui peut être
 - une opération primitive comme une instruction d'assignation ;
 - l'envoi d'un signal ;
 - l'appel d'une opération ;
 - une liste d'activités, etc.
- ✦ L'activité à réaliser peut être spécifiée en langage naturel ou pseudo code).
- ✦ Lorsque l'exécution de l'effet est terminée, l'état cible de la transition devient actif.

Transition externe

- ✦ C'est une transition qui modifie l'état actif.
- ✦ Elle est représentée par une flèche allant de l'état source vers l'état cible



✦ **Transition d'achèvement:**

- transition externe dépourvue d'événement déclencheur;
- se déclenche à la fin de l'activité contenue dans l'état source.
- Elle peut contenir une condition de garde qui est évaluée au moment où l'activité contenue dans l'état s'achève, et non pas ensuite.

Transition interne

- ✦ Une transition interne ne possède pas d'état cible, **l'état actif reste le même à la suite de son déclenchement.**
- ✦ Les règles de déclenchement et la syntaxe d'une transition interne sont les mêmes que pour une transition externe.
- ✦ Les transitions internes ne sont pas représentées par des arcs mais sont spécifiées dans le compartiment de l'état.
- ✦ Les transitions internes possèdent des noms d'événements prédéfinis correspondant à des déclencheurs particuliers.
 - **entry** : activité qui s'accomplit quand on entre dans l'état.
 - **exit** : activité qui s'accomplit quand on sort de l'état.
 - **do** : commence dès que l'activité **entry** est terminée. Si une transition se déclenche, l'activité **do** est interrompue et l'activité **exit** de l'état s'exécute.
 - **include** : invoquer un sous diagramme d'états

Transition interne

✦ Exemple:

saisie mot de passe
entry/set echo invisible
exit/set echo normal
character/traiter caractère
help/afficher aide
clear/remise à zéro mot de passe et chronomètre
after(20s)/exit

Points de choix

✧ But : Regrouper ou diviser des transitions mutuellement exclusives

✧ Point de jonction

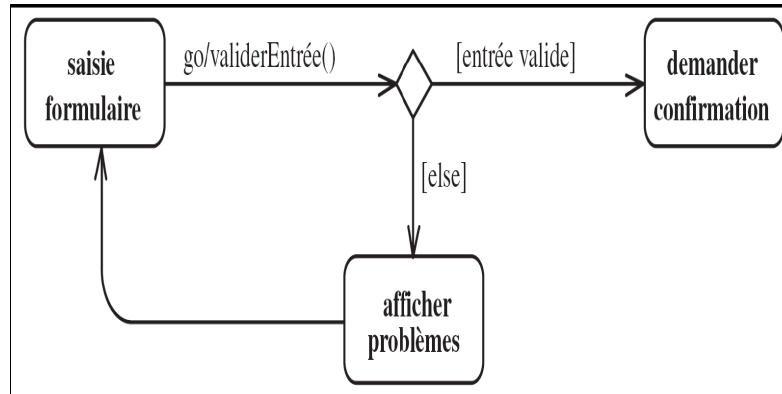
- **statique:** les gardes sont évaluées **avant** le franchissement de la transition
- **dynamique:** les gardes sont évaluées **lorsque le point de jonction est atteint**

✧ Point de décision

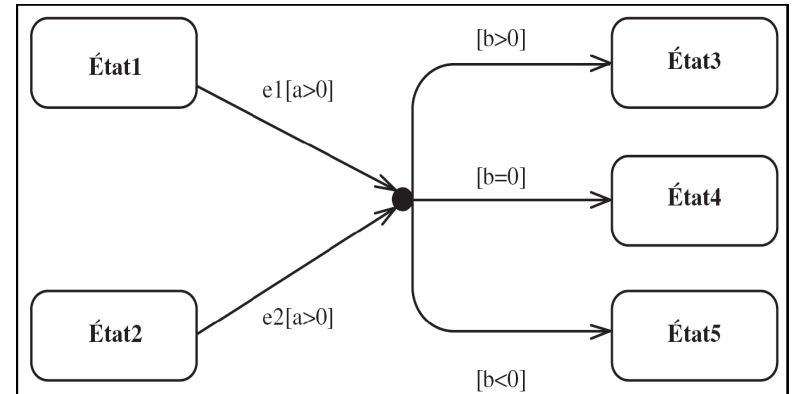
- possède une entrée et au moins deux sorties.
- Les gardes situées après le point de décision sont évaluées au moment où il est atteint.
- Une fois le point de décision atteint, au moins un chemin doit être franchissable.

Points de choix: Exemple

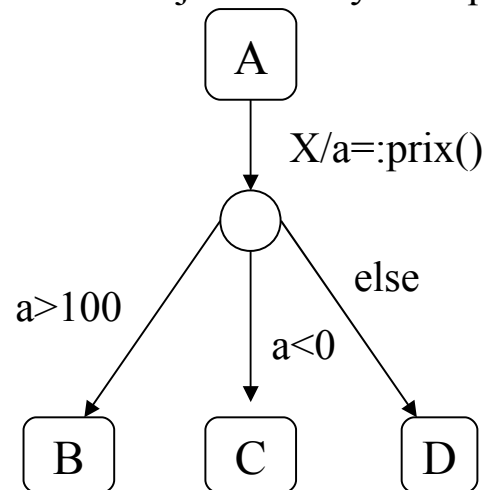
Point de décision



Point de jonction statique



Point de jonction dynamique



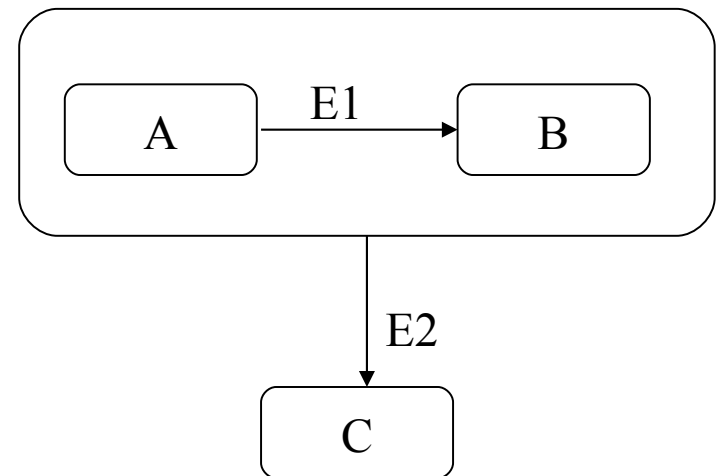
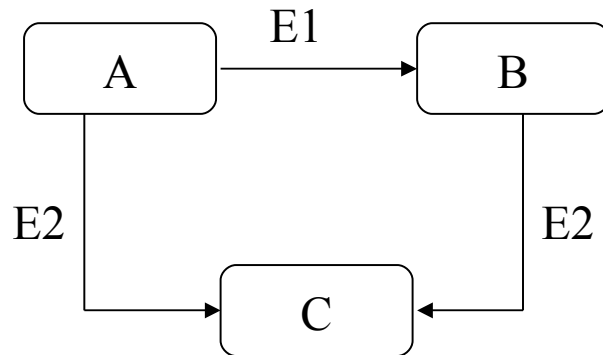
États composites

✧ Un état composite est un état décomposé en régions contenant chacune un ou plusieurs sous états.

✧ Les sous états sont soit concurrents soit disjoints

✧ **États disjoints:**

- Les sous états sont mutuellement exclusifs, l'objet doit se trouver dans un seul état à la fois.

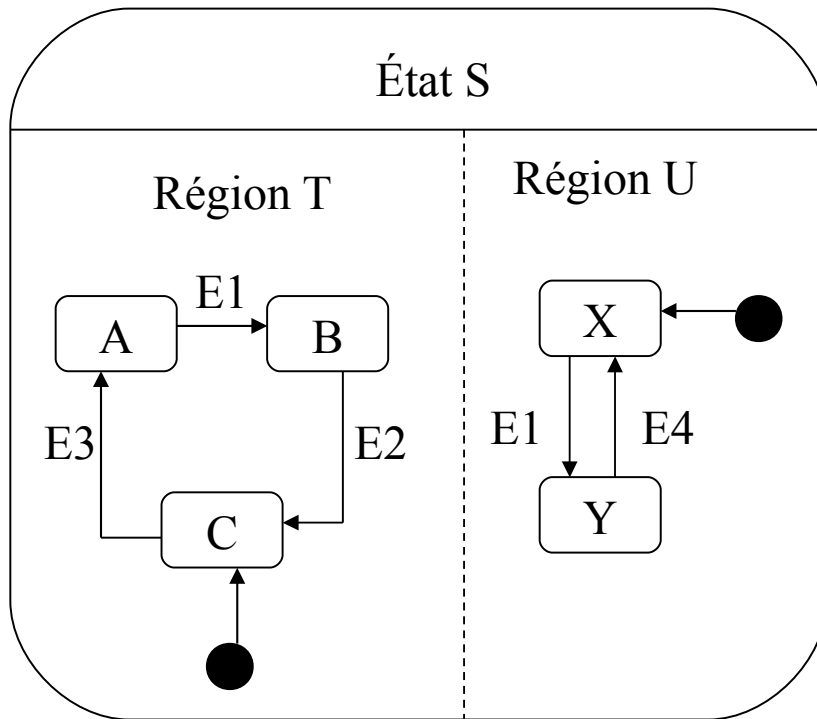


États composites

✦ États concurrents:

- Un état peut être composé de plusieurs sous états concurrents. Ces sous états sont alors appelés **régions**
- Dans ce type de composition, l'objet doit être simultanément dans tous les sous états.
- Les régions dans un état composite sont représentées dans le symbole de l'état et elles sont séparées par des lignes en pointillées.
- Il est possible d'attribuer un nom à une région donnée. Ce nom reste, toutefois, optionnel
- Le déclenchement d'une transition vers l'état composite entraîne l'activation de tous les états initiaux des diverses régions.

États composites



- ✧ Si un événement arrive à S, T et U seront simultanément activées. L'objet est alors placé dans l'état composite (C,X)
- ✧ Si l'événement E3 arrive, T et U évoluent indépendamment. L'objet passe alors à l'état composite (A,X)
- ✧ T et U évoluent simultanément lorsque l'objet reçoit l'événement E1 alors qu'il est à l'état (A,X) auquel cas il passe à (B,Y)

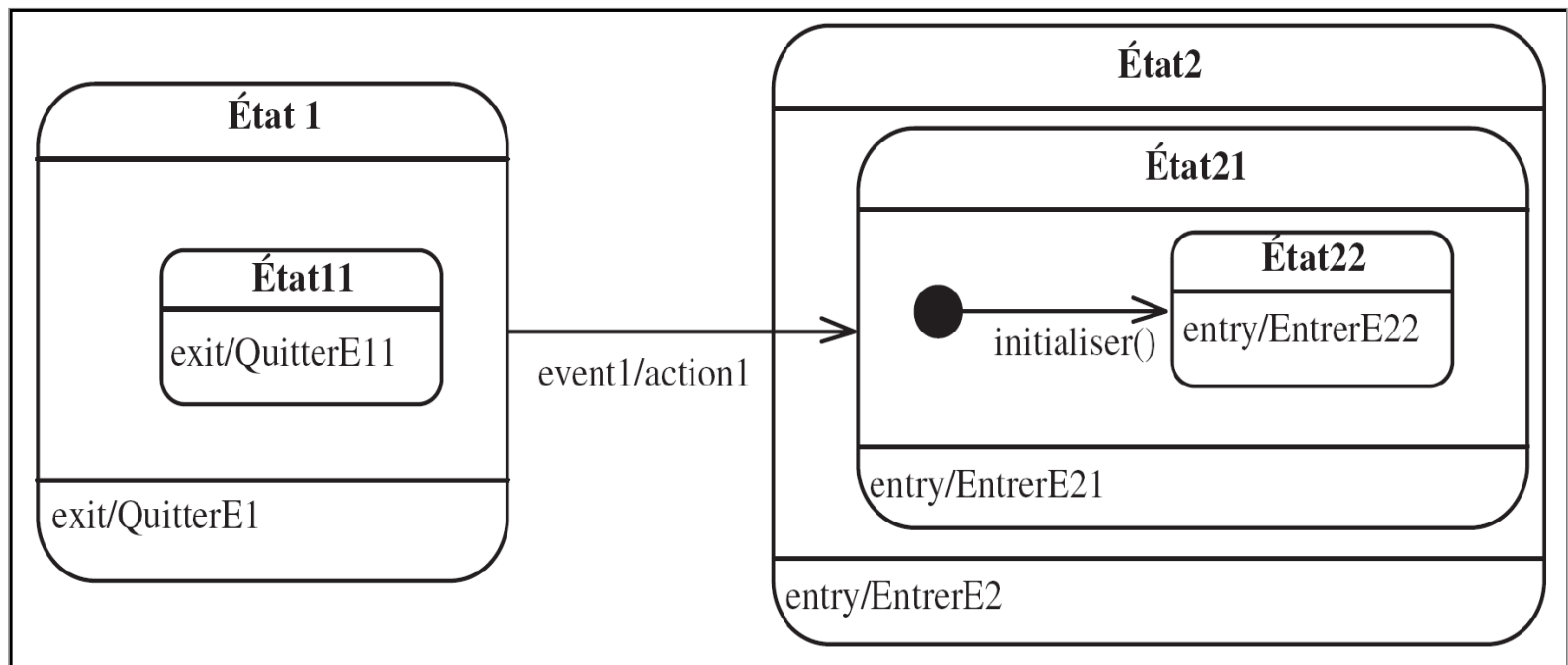
États composites

✦ Transitions entre états composites

- Les actions d'entrée des états sont toujours effectuées de manière séquentielle dans l'ordre d'accès (du niveau hiérarchique le plus élevé au plus bas).
- Une transition provenant d'un état composite s'applique à tous ses sous états. Quelque soit le niveau d'emboîtement, les sous états héritent de cette transition de sortie.
- Le déclenchement de la transition entraîne la sortie de tous les sous états, la réalisation des différentes actions de sortie (du niveau hiérarchique le plus bas jusqu'au niveau courant), puis la réalisation de l'action de la transition sortant de l'état composite avant le passage au nouvel état.

Transitions entre états composites

✦ **Exemple:** Depuis l'état État 1, la réception de l'événement event1 produit la séquence d'activités QuitterE11, QuitterE1, action1, EntrerE2, EntrerE21, initialiser(), EntrerE22, et place le système dans l'état État22.



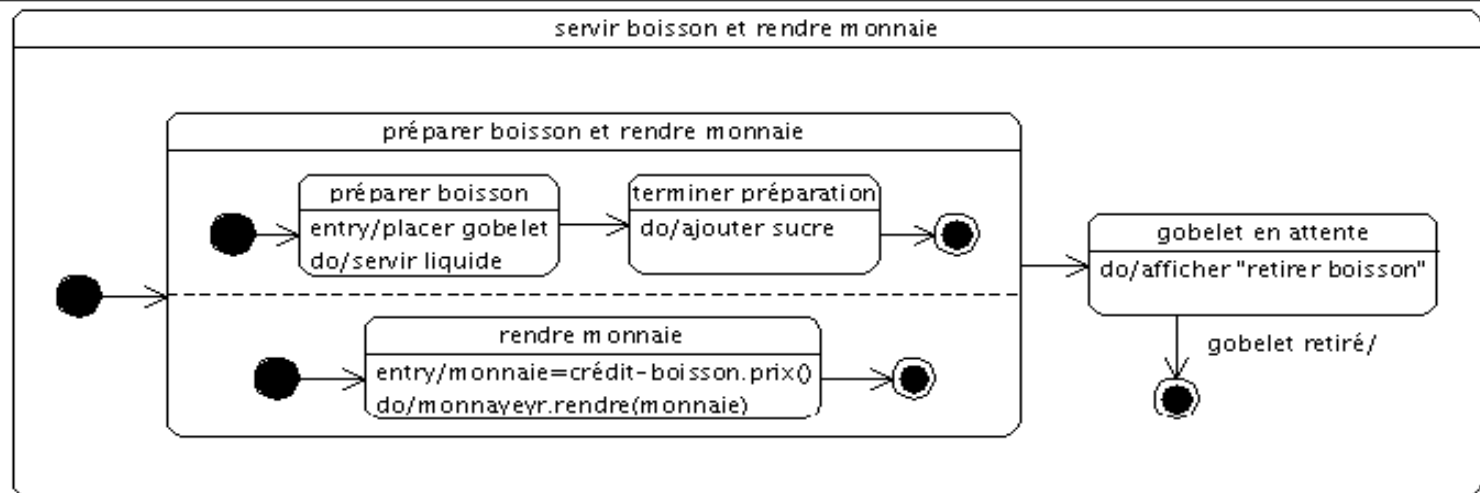
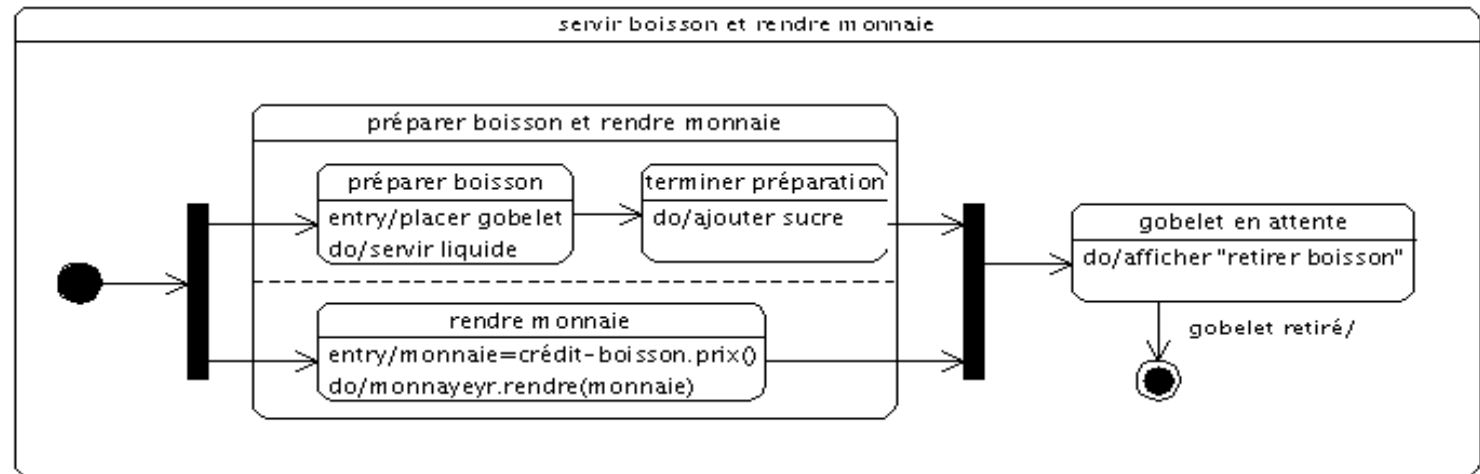
États composites



Transitions entre états concurrents

- Une transition concurrente peut avoir plusieurs états sources et destinations
- Il est possible de définir des synchronisations ou des répartitions du flot de contrôle en utilisant respectivement des **transitions de synchronisation** ou de **débranchement**
- La représentation d'une transition concurrente se fait par le biais d'une barre de synchronisation symbolisée par un trait épais
- Plusieurs flèches peuvent arriver sur cette barre ou en partir selon qu'une synchronisation, un débranchement ou les deux sont définis
- Un nom des transitions peut être ajouté à proximité de la barre
- La transition concurrente est déclenchée lorsque tous les états sources sont atteints et entraîne ainsi l'activation de tous les états cibles

Transition concurrente: exemples



État historique

- ✦ UML permet de mémoriser le dernier sous état visité par le biais de l'indicateur spécial H représenté par un cercle contenant la lettre H
- ✦ Le mécanisme d'historique s'applique au niveau dans lequel l'indicateur H est déclaré
- ✦ L'indicateur H peut être placé n'importe où dans l'état composite

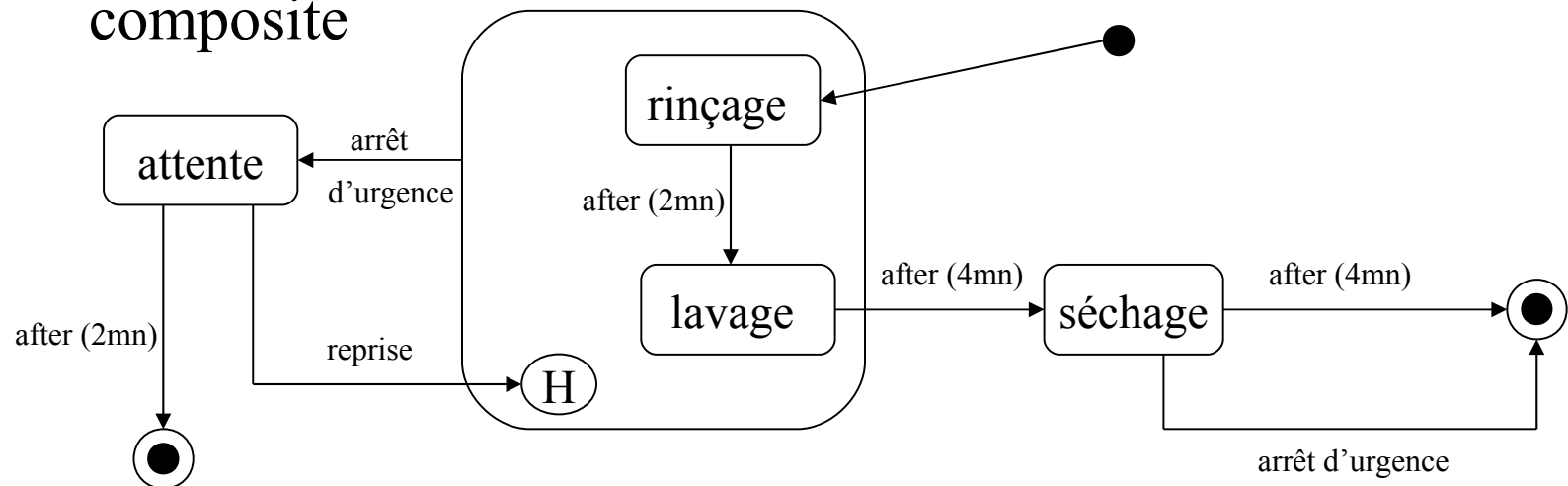




Diagramme d'activités (DAC)

Dame Samb
UCAD/FST/DMI

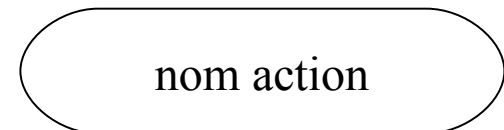


Diagramme d'activités (DAC): objectifs

- ✦ Les DACs sont relativement proches des DETs dans leur présentation, mais leur interprétation est sensiblement différente.
- ✦ Les DACs mettent l'accent sur les activités (traitements) et les transitions.
- ✦ Un DAC peut être utilisé pour modéliser le comportement interne d'une méthode (réalisation d'une opération) ou le déroulement d'un cas d'utilisation.
- ✦ Les DACs offrent une vision proche des langages de programmation impératifs comme C++ ou Java.

Action

- ✦ Une action est le plus petit traitement qui puisse être exprimé en UML. Elle a une incidence sur l'état du système ou en extrait une information.
- ✦ Les actions sont des étapes discrètes à partir desquelles se construisent les comportements.
- ✦ Une action peut être, par exemple :
 - une affectation de valeur à des attributs ;
 - un accès à la valeur d'une propriété structurelle ;
 - la création d'un nouvel objet ou lien ;
 - un calcul arithmétique simple ;
 - l'émission d'un signal ;
 - la réception d'un signal ;
 - ...



Le nom (placé à l'intérieur du symbole) n'est pas nécessairement unique dans un DAC donnée

Action

✦ Une action peut être définie en langage naturel, en pseudo code ou avec un langage de programmation

✦ Exemples

Ouvrir porte

Dossier.ouvrir(fichier)

$i:=i+1$

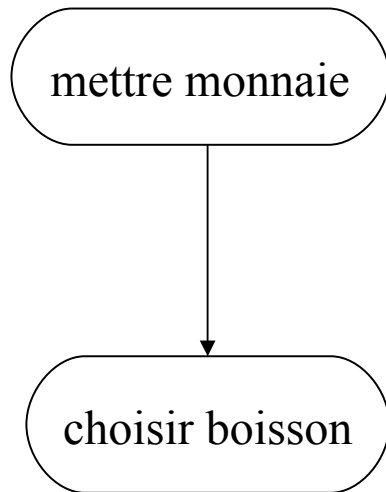
activités

- ✧ 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 noeuds reliés par des arcs (transitions).
- ✧ Les étapes le long du flot d'une activité sont représentées par des **noeuds d'activités**
- ✧ Il existe trois familles de noeuds d'activités
 - les noeuds exécutables;
 - les noeuds objets ;
 - et les noeuds de contrôle

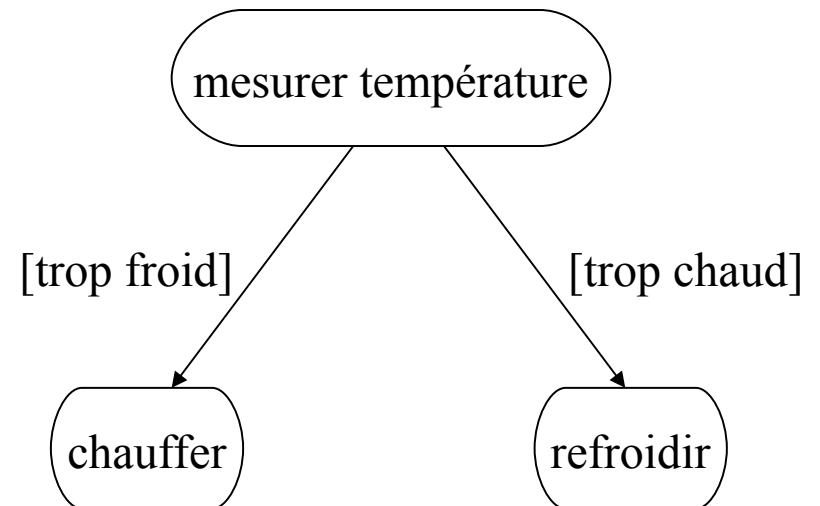
Transition

- ✦ Les transitions matérialisent le passage d'une activité vers une autre. Elles sont représentées par des flèches en traits pleins qui connectent les activités entre elles.
- ✦ Les transitions peuvent être **automatiques**
 - 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).
 - Elles ne possèdent pas de signature d'événement
- ✦ Les transitions peuvent inclure des actions telles que définies dans les automates d'états (DET)
- ✦ Les transitions peuvent prendre des conditions de garde booléennes, mutuellement exclusives appelées décisions

Transition : Exemples



Transition automatique



Transition conditionnelle

Nœuds exécutables

✦ Un noeud exécutable est un noeud d'activité qu'on peut exécuter (i.e. une activité).

✦ **Noeud d'action**

- Unité fondamentale de fonctionnalités exécutables 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 noeud d'action doit avoir au moins un arc entrant.

Saisir code

Nœuds exécutables

✦ Nœud d'activité structurée

- Portion structurée d'une activité donnée qui n'est partagée avec aucun autre nœud structuré.
- Graphiquement, le contour d'un nœud d'activité structurée est en pointillé.
- Les transitions d'une activité structurée doivent avoir leurs nœuds sources et cibles dans le même nœud d'activité structurée.
- Les nœuds et les arcs contenus dans un nœud d'activité structurée ne peuvent pas être contenus dans un autre nœud d'activité structurée.
- Un nœud structuré est dénoté par le stéréotype « *structured* » et identifié par un nom unique.

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 ;
 - nœud de fin d'activité ;
 - nœud de fin de flot ;
 - nœud de décision ;
 - nœud de fusion ;
 - nœud de bifurcation ou de débranchement ;
 - nœud d'union ou de jointure.

Nœuds de contrôle



Noeud initial

- noeud de contrôle à partir duquel le flot débute lorsque l'activité enveloppante est invoquée. Il est représenté par un petit cercle plein.



Noeud de fin d'activité

- Lorsqu'un flot d'exécution atteint un noeud de fin d'activité, l'exécution de l'activité enveloppante s'achève et tout noeud ou flot actif au sein de l'activité enveloppante est abandonné.
- Si l'activité a été invoquée par un appel synchrone, un message contenant les valeurs sortantes est transmis en retour à l'appelant.



Noeud de fin de flot

- Lorsqu'un flot d'exécution atteint un noeud de fin de flot, il se termine, mais cette fin de flot n'a aucune incidence sur les autres flots actifs de l'activité enveloppante.

Nœuds de contrôle

✧ **Noeud de décision**

- permet de faire un choix entre plusieurs flots sortants. Il possède un arc entrant et plusieurs arcs sortants. Il est représenté par un losange

✧ **Noeud de fusion**

- sert à accepter un seul flot (sortant) parmi plusieurs flots (entrants)
- Il est représenté par un losange et il est possible de fusionner un noeud de fusion et un noeud de décision.

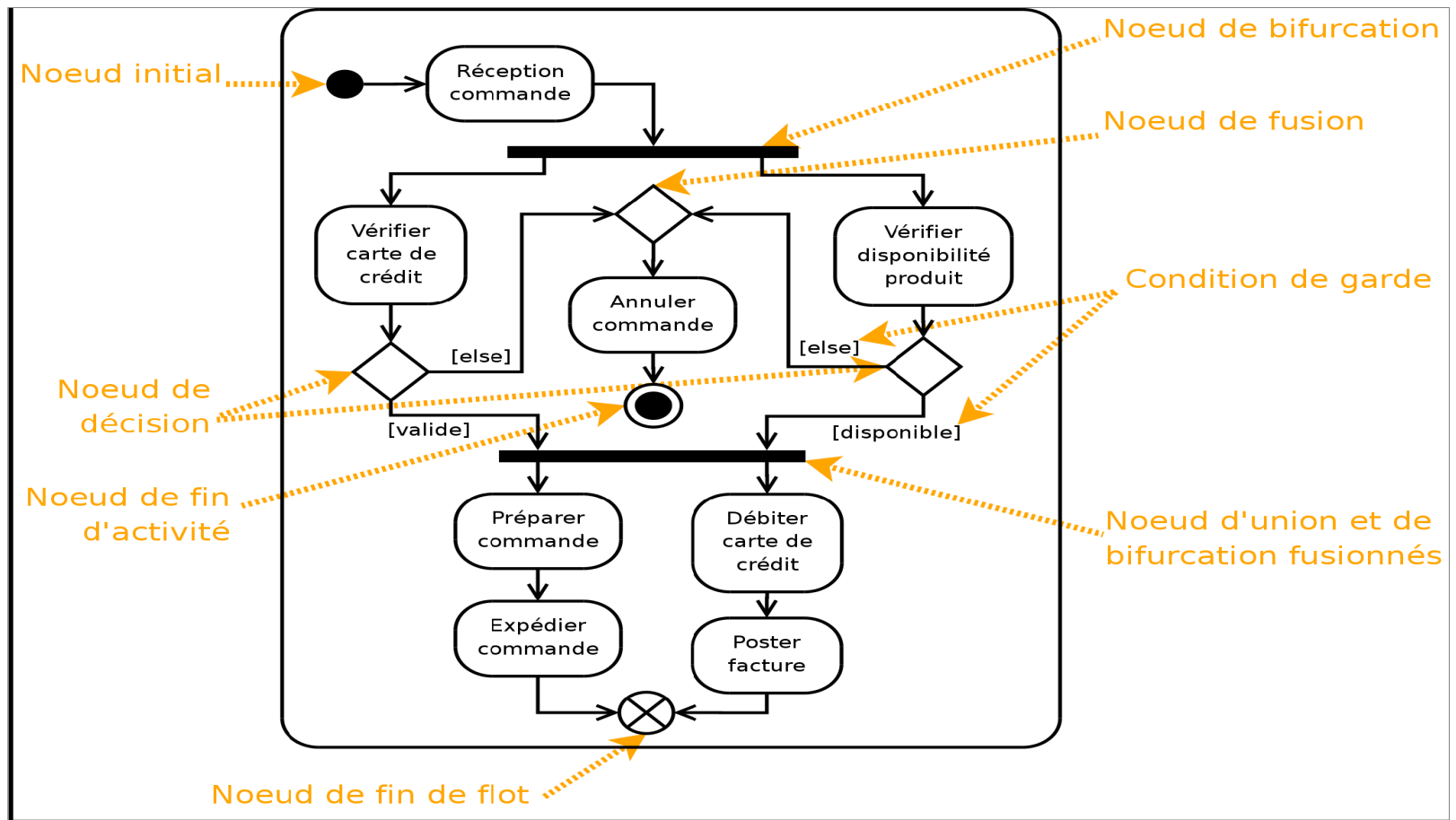
✧ **Noeud de bifurcation ou débranchement**

- sépare un flot en plusieurs flots concurrents. Il possède donc un arc entrant et plusieurs arcs sortants. Il est représenté par un trait plein

✧ **Noeud d'union ou de jointure**

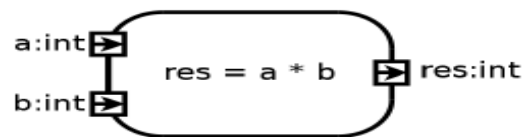
- synchronise des flots multiples. Il possède donc plusieurs arcs entrants et un seul arc sortant. Il est représenté par un trait plein
- Il est possible de fusionner un noeud de bifurcation et un noeud de union

Nœuds de contrôle: illustration



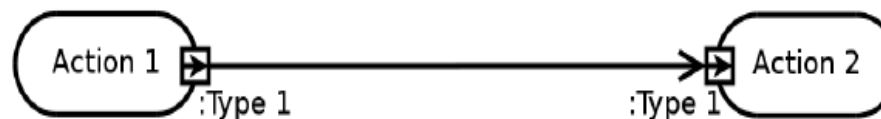
Nœuds d'objets

- ✦ Un nœud d'objet permet de définir un flot de données dans un diagramme d'activités.
- ✦ Il représente l'existence d'un objet généré par une action dans une activité et utilisé par d'autres actions.
- ✦ **Pin d'entrée ou de sortie:**
 - Permet de spécifier les valeurs passées en argument (pin d'entrée) à une activité et les valeurs de retour (pin de sortie).



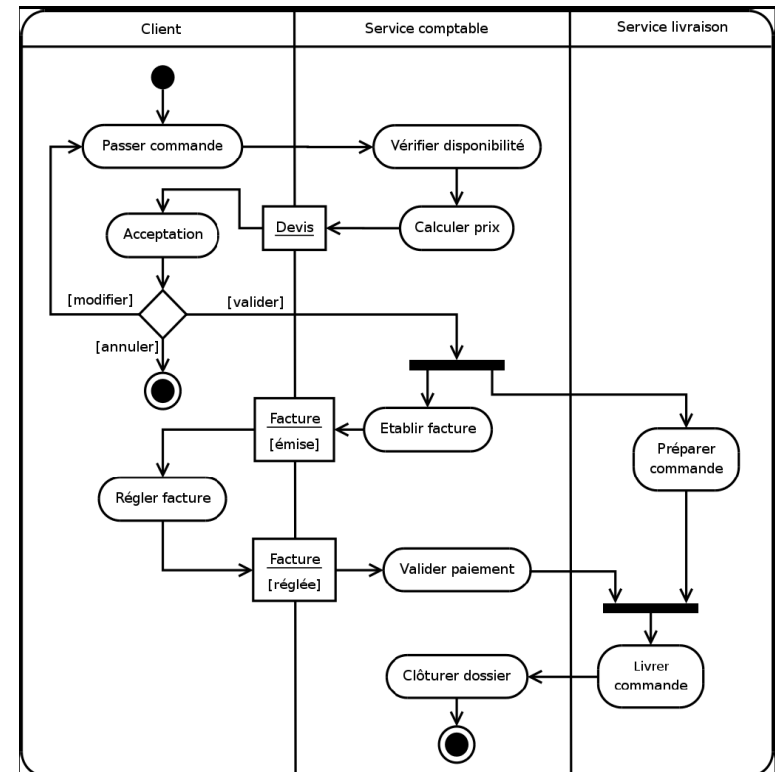
✦ Flot de données:

- permet de passer des données d'une activité à une autre



Partitions

✦ Les DACs peuvent être découpés en plusieurs partitions pour montrer les responsabilités au sein d'un mécanisme ou d'une organisation



Outils de modélisation UML

✦ **Logiciels propriétaires**

- Rational Rose: <http://www.rational.com>
- Together: <http://www.togethersoft.com>
- Objectteering: <http://www.objectteering.com>
- Visio: <http://www.microsoft.com/office/visio>

✦ **Logiciels libres**

- Umbrello: <http://uml.sourceforge.net>
- AgroUML: <http://argouml.tigris.org>
- BoUML: <http://www.bouml.free>