

DIAGRAMME D'ACTIVITÉ

Diagramme comportemental

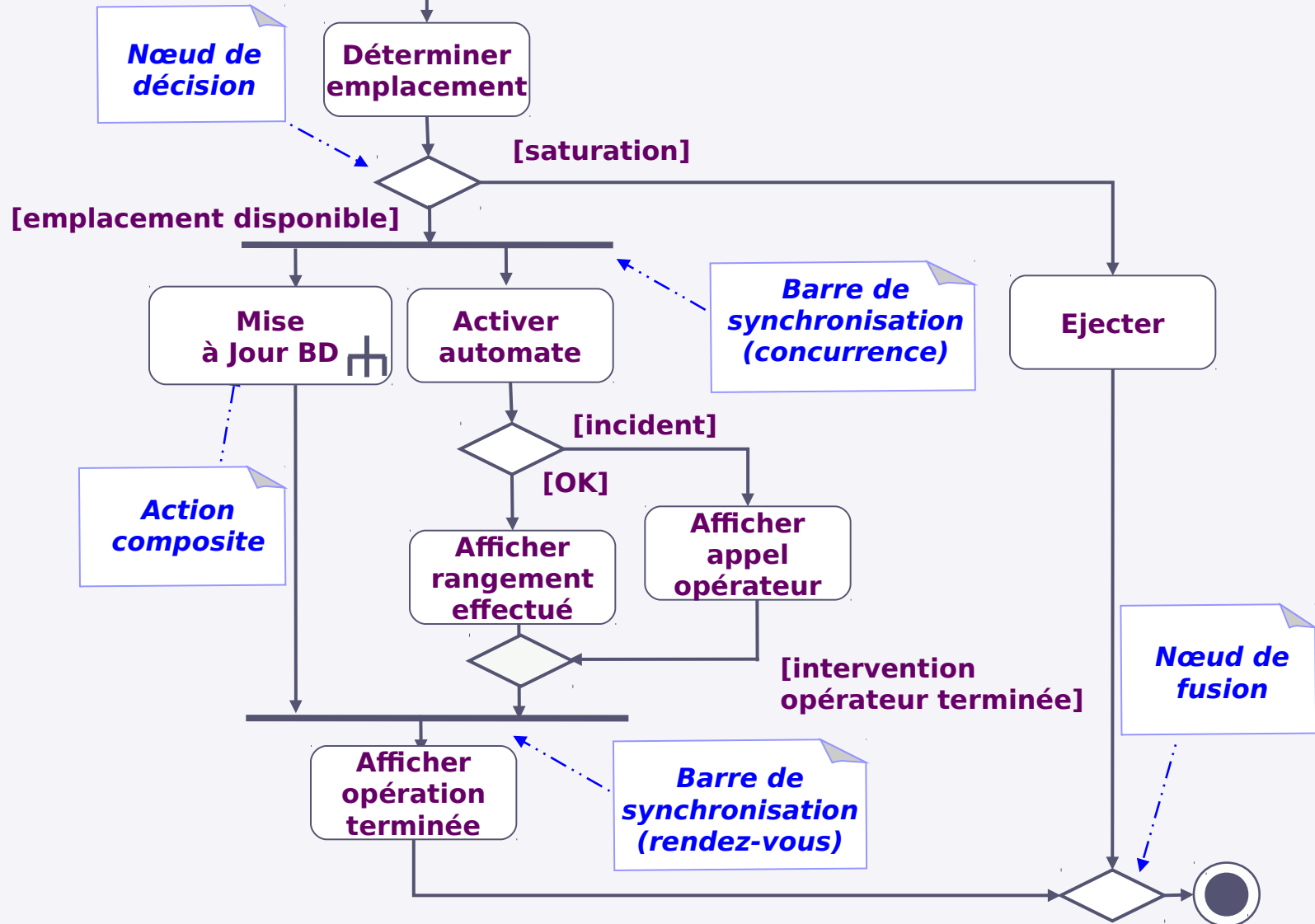
*Enchaînement des actions au
sein d'une activité à l'aide de
flots de contrôle et d'objets*



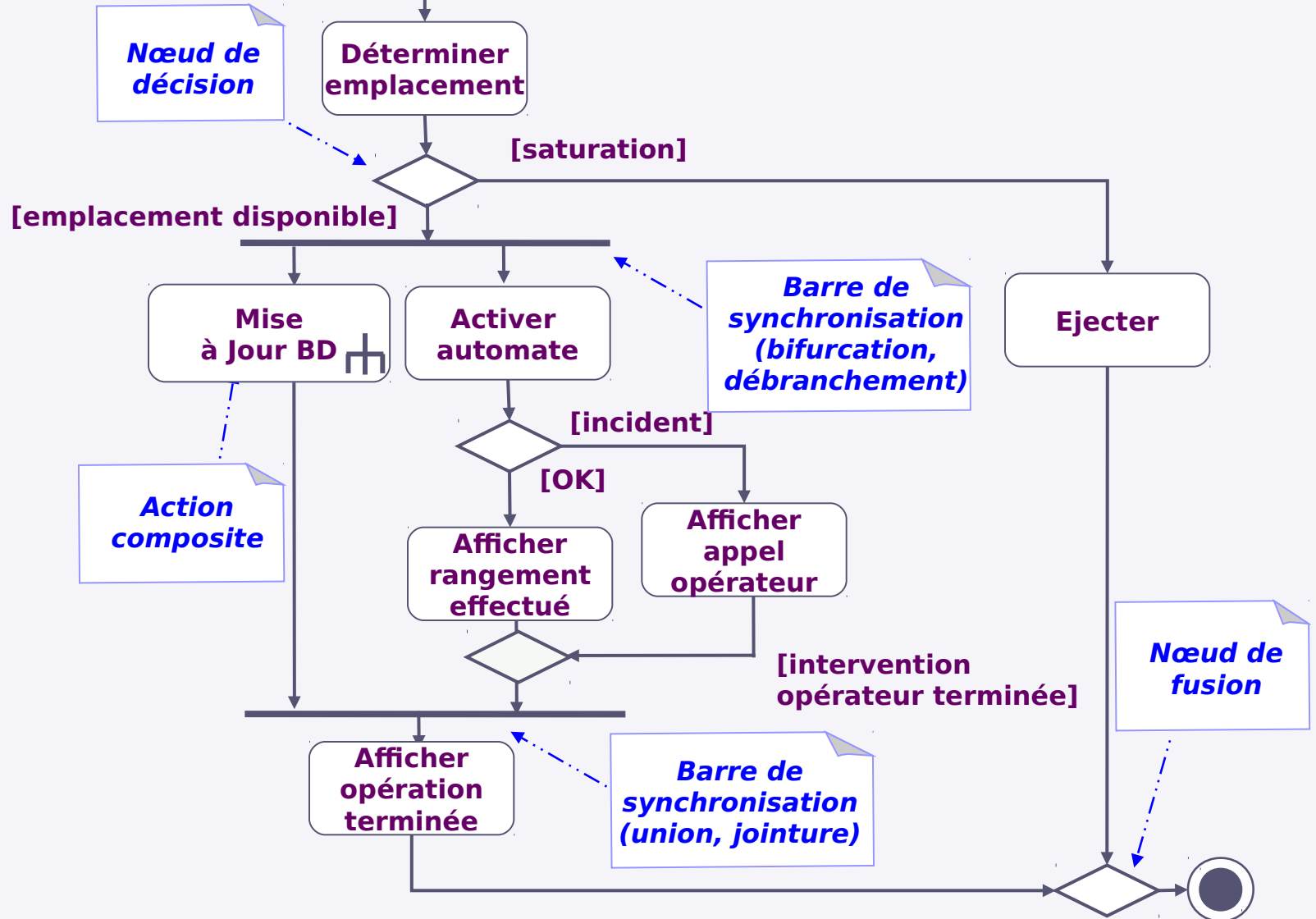
INTRODUCTION

- **Diagramme comportemental** d'UML.
- Permet de modéliser tout système ayant un comportement.
- Représentation **proche de l'organigramme**
- UML permet de représenter graphiquement :
 - ✓ le **comportement d'une méthode** ou
 - ✓ le **déroulement d'un cas d'utilisation**,
à l'aide de diagrammes d'activité
(*une variante des diagrammes d'états-transitions*).
- Réservé à des **opérations complexes** ou **sensibles**.

RANGEMENT D'UNE PALETTE



RANGEMENT D'UNE PALETTE



INTRODUCTION

- La **description d'un cas d'utilisation** par un diagramme d'activité correspond à sa **traduction algorithmique**.
- On se concentre ici sur les **activités telles que les voient les acteurs** qui collaborent avec le système dans le cadre d'un processus métier.
- La différence principale entre les **diagrammes d'interaction** et les **diagrammes d'activité** est que les premiers mettent l'accent sur le **flot de contrôle d'un objet à l'autre**, tandis que les seconds insistent sur le **flot de contrôle d'une action à l'autre**.

DIAGRAMME D'ACTIVITÉ

- Un **diagramme d'activité de base** contient un nombre restreint d'éléments :
 - ✓ des **actions**
 - ✓ des **transitions** entre actions, pouvant porter des **conditions**
 - ✓ des **branchements conditionnels** ou **décisions**
 - ✓ un **début** et une ou plusieurs **terminaisons** possibles.

EXEMPLE

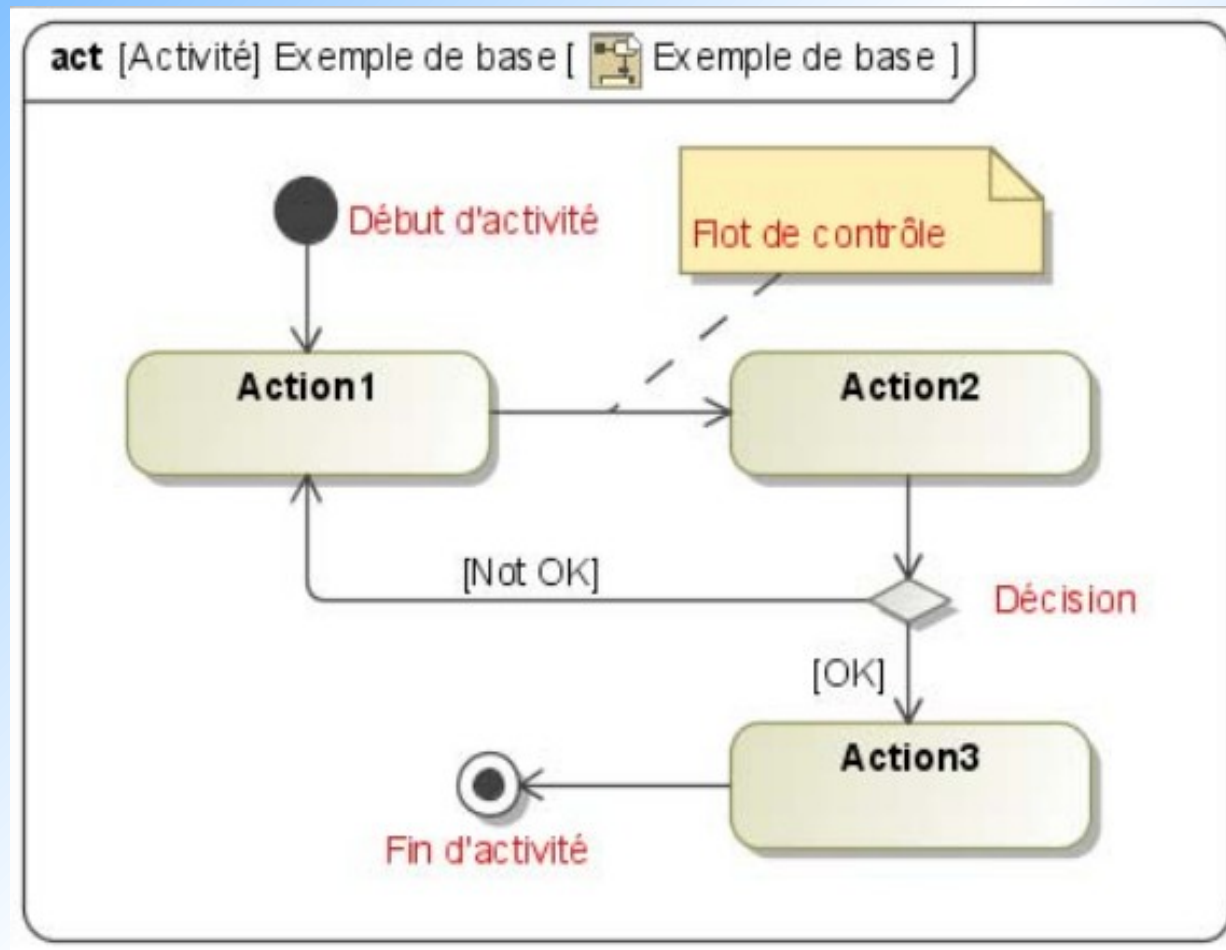


DIAGRAMME D'ACTIVITÉ

- Dans un diagramme, **une seule action** est en cours (*ou active*) **à la fois**.
- Si plusieurs transitions sortent d'une même action, cela signifie qu'il y a un choix disponible pour l'utilisateur.
Une seule des transitions sera déclenchée lors d'une navigation particulière.

action

ACTIVITÉ / ACTION

activité

râteau en bas à droite

- Une **activité** représente l'exécution d'un calcul complexe, alors qu'une action est un comportement élémentaire.
- Une **activité** définit un comportement décrit par un séquençement organisé d'unités dont les éléments simples sont les actions.
- Une **activité** peut être décomposée en **plusieurs actions**.

ACTIVITÉ / ACTION

- Une **action** est le plus petit traitement qui puisse être exprimé en UML. Cela représente une **étape simple de l'activité**.
- Une étape simple signifie que l'**action** n'est **pas décomposée** en éléments plus simples. Ne veut pas dire que l'action elle-même est simple.
- **Exemples d'actions** :
 - fonctions mathématiques
 - appels à d'autres comportements (*activités*)
 - traitements de données

ACTIVITÉ / ACTION

- **Exemples plus concrets d'actions :**

- calcul de TVA
- envoi d'une commande à un fournisseur
- constitution d'une liste d'articles à commander par le fournisseur car son stock s'épuise.

- Les **activités** sont destinées à être **réutilisées** au sein d'une application alors que les **actions** sont typiquement **spécifiques** et utilisées seulement dans le cadre **d'une activité** donnée.

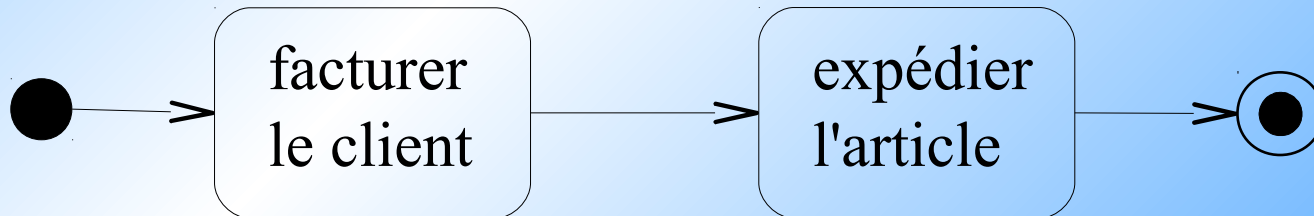
ACTIVITÉ / ACTION

paramètre
impliqué dans
l'activité

traiter une vente
articleAcheté : Article

activité simple

traiter une vente
articleAcheté : Article

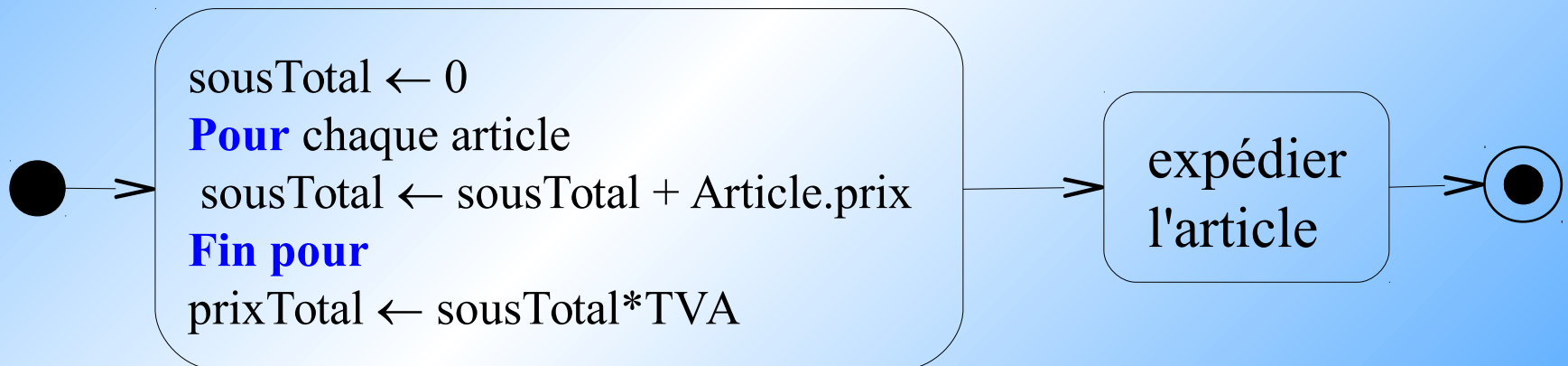


activité détaillée

ACTIVITÉ / ACTION

- On peut **intégrer du pseudo-code** à l'intérieur d'une action.

traiter une vente
articleAcheté : Article

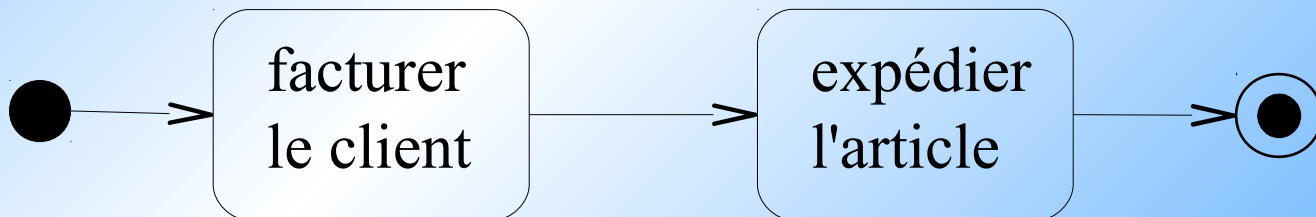


ACTIVITÉ / ACTION

- Une activité peut comporter des **conditions a priori** et **a posteriori** qui s'appliquent en totalité à celle-ci.

traiter une vente
articleAcheté : Article

<<precondition>> vente terminée
<<postcondition>> articles envoyés



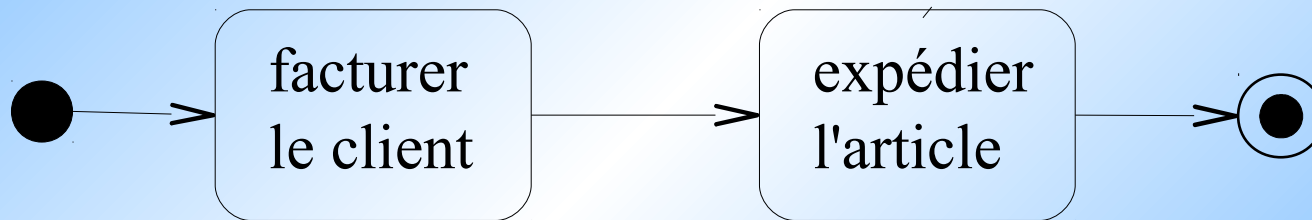
ACTIVITÉ / ACTION

- Une **action** peut être assortie de **conditions locales a priori** et **a posteriori** qui doivent être satisfaites respectivement avant l'invocation de l'action et après la fin de celle-ci.

ACTIVITÉ / ACTION

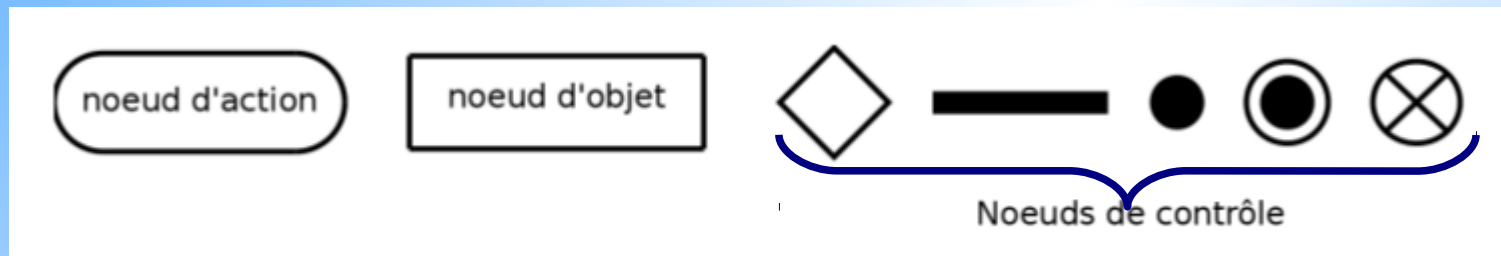
traiter une vente
articleAcheté : Article

<<localPrecondition>>
l'adresse de l'acheteur
est valide.



<<localPostcondition>>
une facture a été
envoyée à l'acheteur.

NŒUDS D'ACTIVITÉ

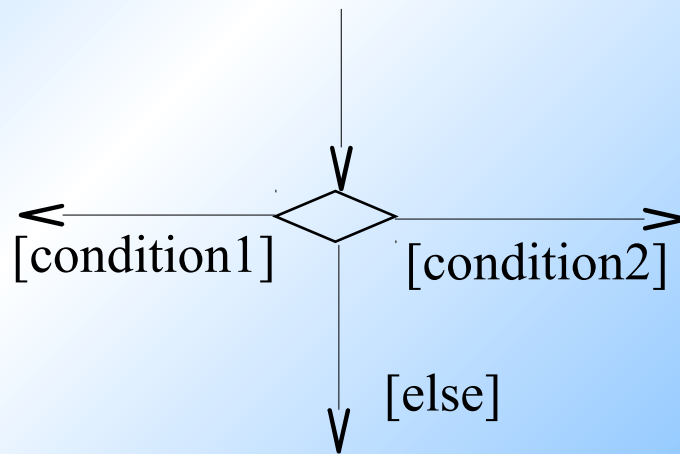


Noeuds de contrôle :

- noeud de décision ou de fusion
- noeud de bifurcation ou d'union
- noeud initial
- noeud final
- noeud final de flot

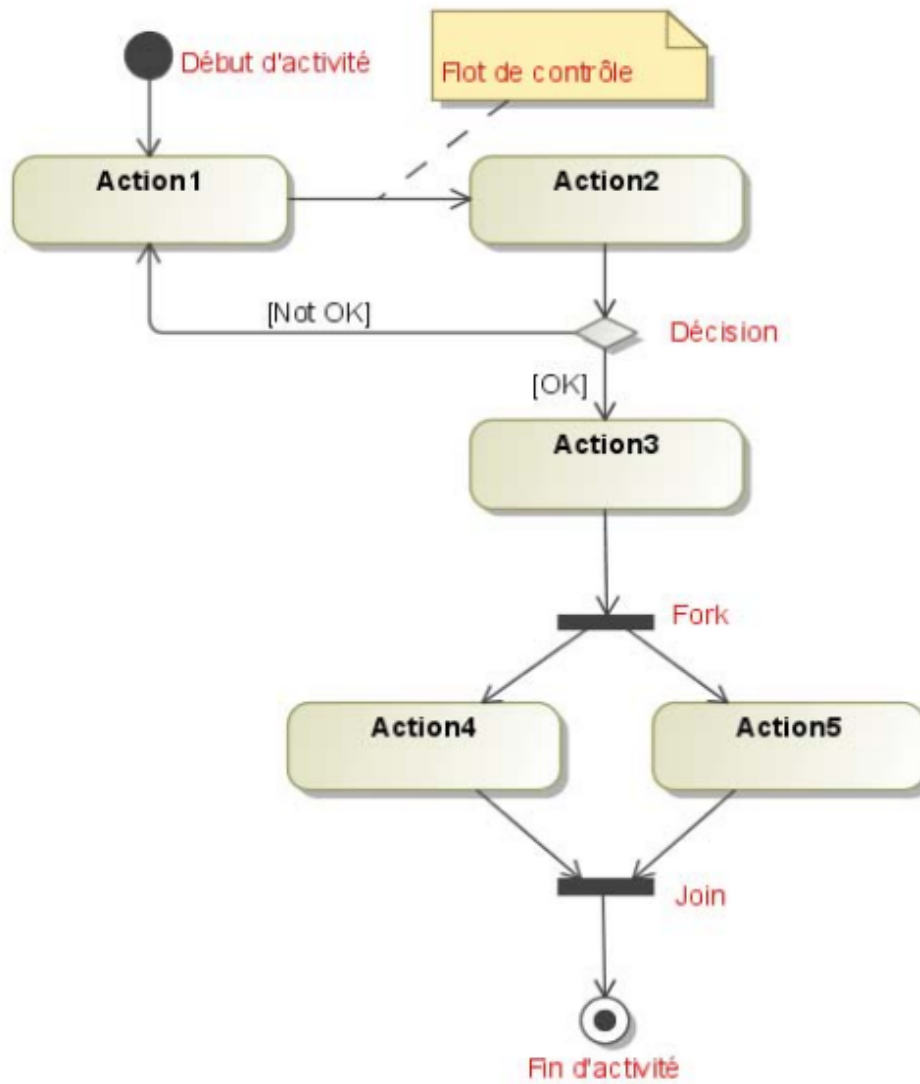
NŒUD DE DÉCISION

- 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.



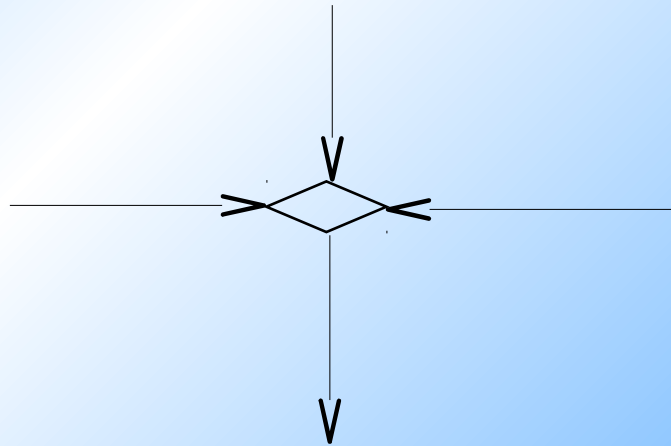
NŒUD DE DÉCISION

- Si, quand le nœud de décision est atteint, aucun arc en aval n'est franchissable (*i.e. aucune condition de garde n'est vraie*), c'est que le **modèle est mal formé**.
- L'utilisation d'une garde **[else]** est recommandée après un nœud de décision, car elle garantit un modèle bien formé. En effet, la condition de garde **[else]** est validée si et seulement si toutes les autres gardes des transitions ayant la même source sont fausses.
- Dans le cas où plusieurs arcs sont franchissables (*i.e. plusieurs conditions de garde sont vraies*), seul l'un d'entre eux est retenu et ce choix est non déterministe.

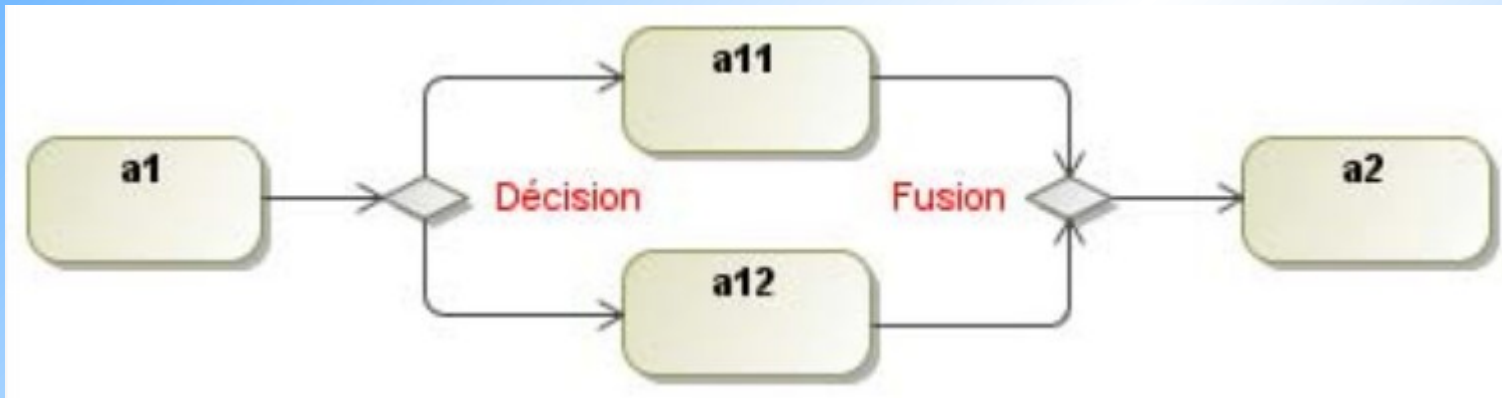


NŒUD DE FUSION

- 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 **accepter un flot parmi plusieurs**.



NŒUD DE FUSION



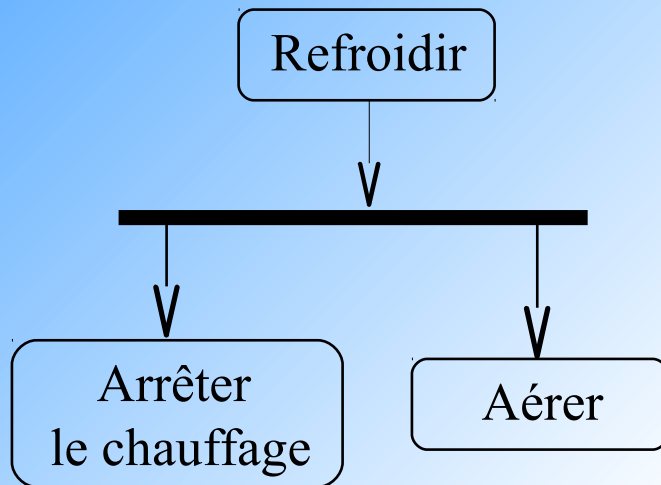
NŒUD DE BIFURCATION

- 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.
- Les **transitions** au départ d'une **barre de synchronisation** sont déclenchées **simultanément**.



NŒUD DE BIFURCATION

Exemple

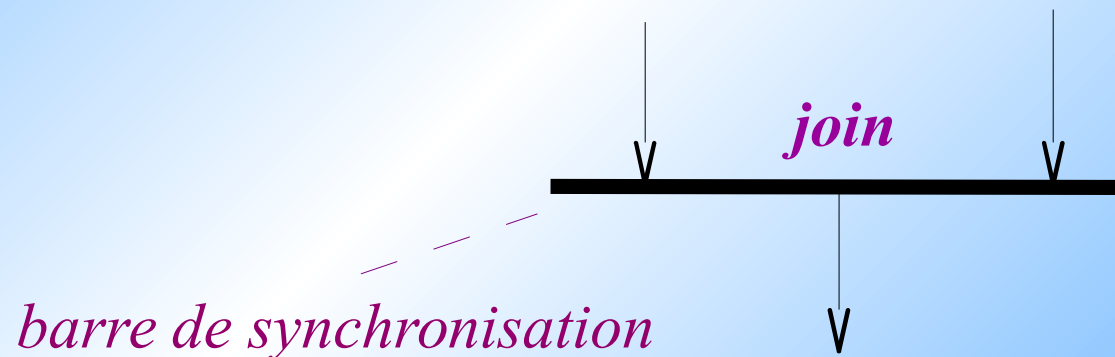


- Pour refroidir, il faut **simultanément**
 - ✓ arrêter le chauffage et
 - ✓ ouvrir les fenêtres

- Cela signifie également que l'**ordre d'exécution des actions n'a pas d'importance**.
- Le diagramme se borne à énoncer les règles de séquencement minimales qui doivent être appliquées.

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.
- Lorsque tous les arcs entrants sont activés, l'arc sortant l'est également.

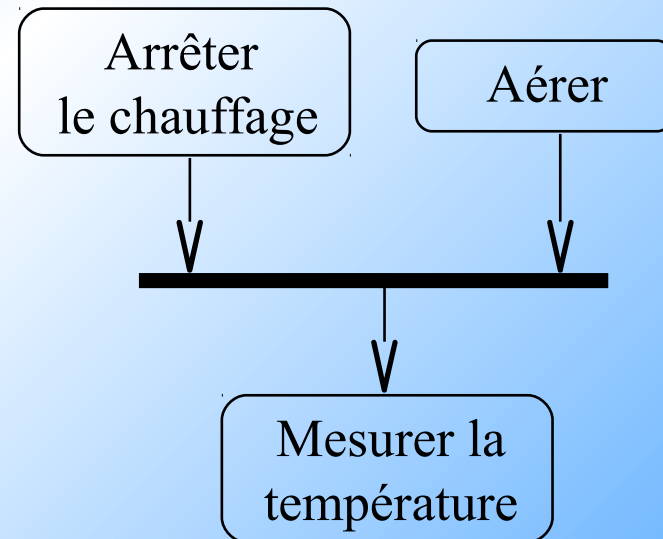


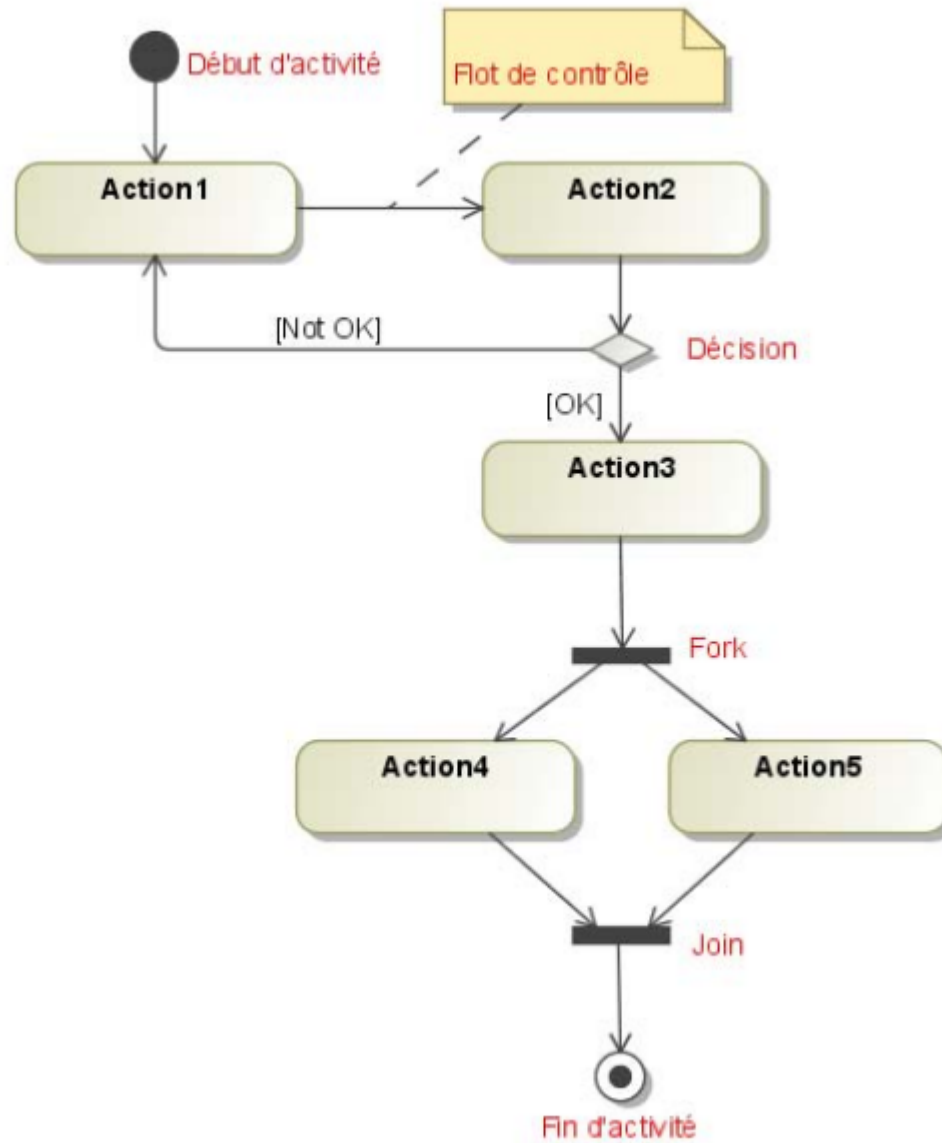
NŒUD D'UNION

- Inversement, un nœud d'union ne peut être franchie que lorsque toutes les transitions en entrée sur la barre ont été déclenchées.

- La mesure de température est effectuée une fois que :

- ✓ le chauffage est arrêté **et**
- ✓ la pièce aérée.

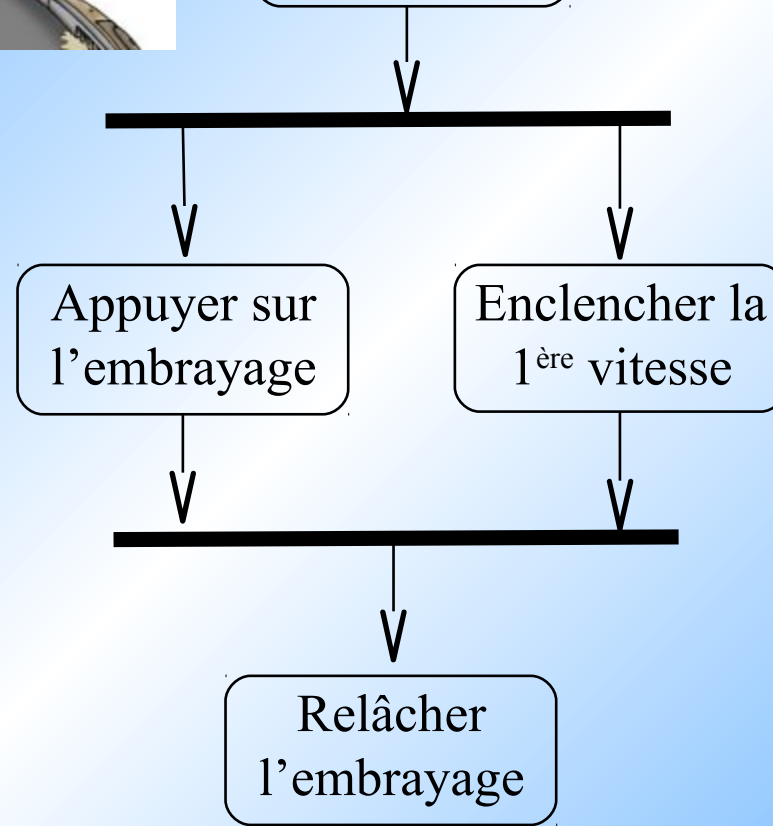




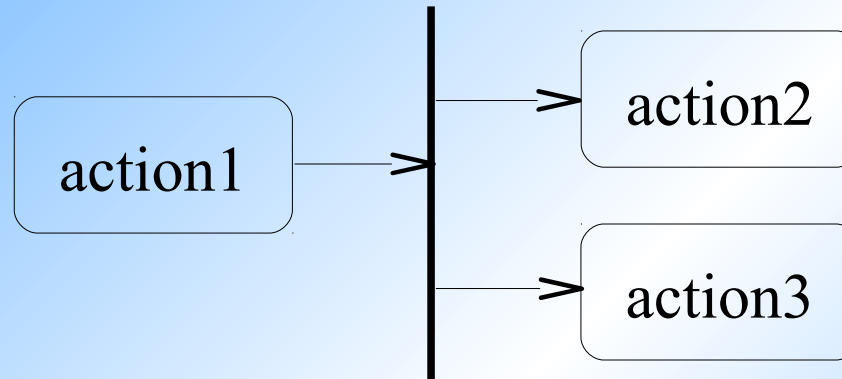


EXEMPLE

Desserrer le
frein à main



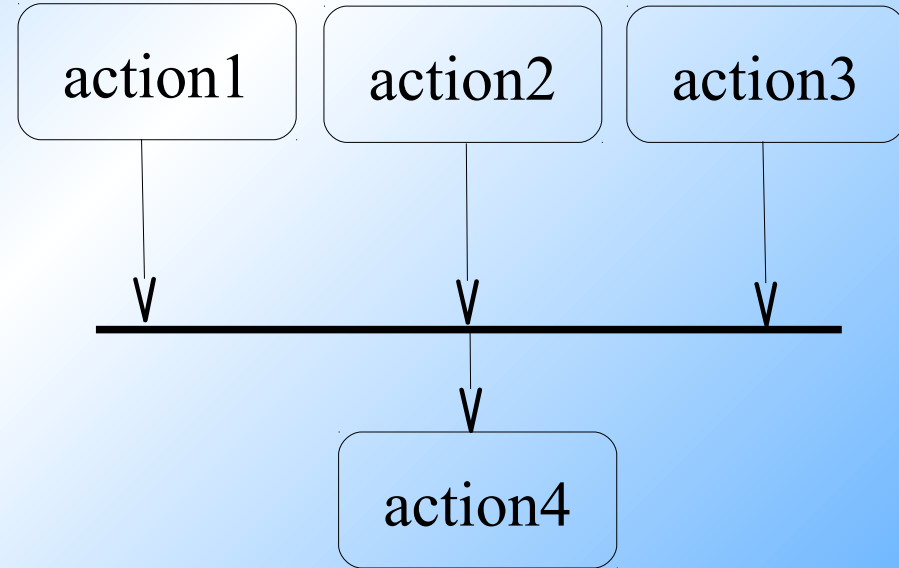
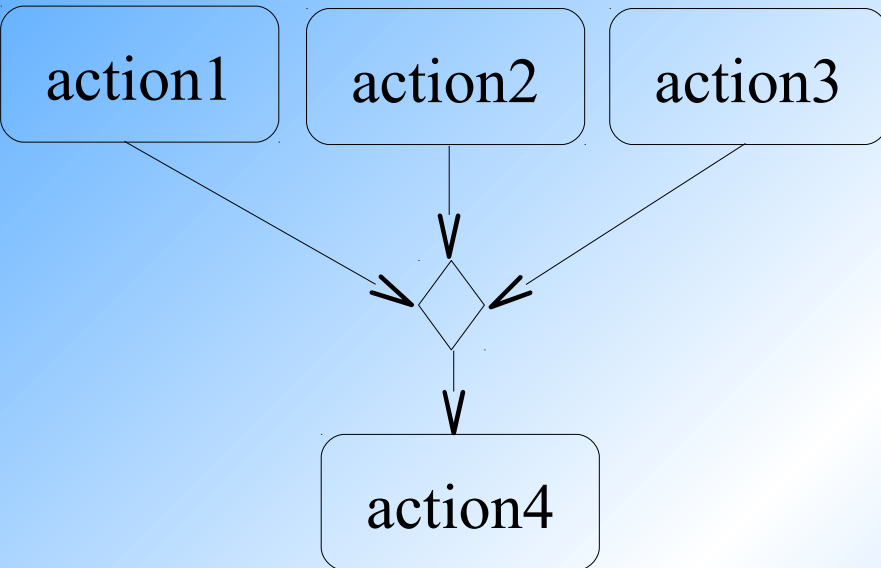
EXERCICE



L'action1 se termine. Dire ce qui est vérifié :

- ☐ L'action2 et l'action3 s'exécutent simultanément.
- ☐ L'action2 uniquement s'exécute.
- ☐ L'action2 s'exécute puis l'action3 s'exécute sans pour autant que l'action2 soit terminée.
- ☐ L'action3 uniquement s'exécute.
- ☐ L'action3 s'exécute puis l'action2 s'exécute sans pour autant que l'action3 soit terminée.

EXERCICE

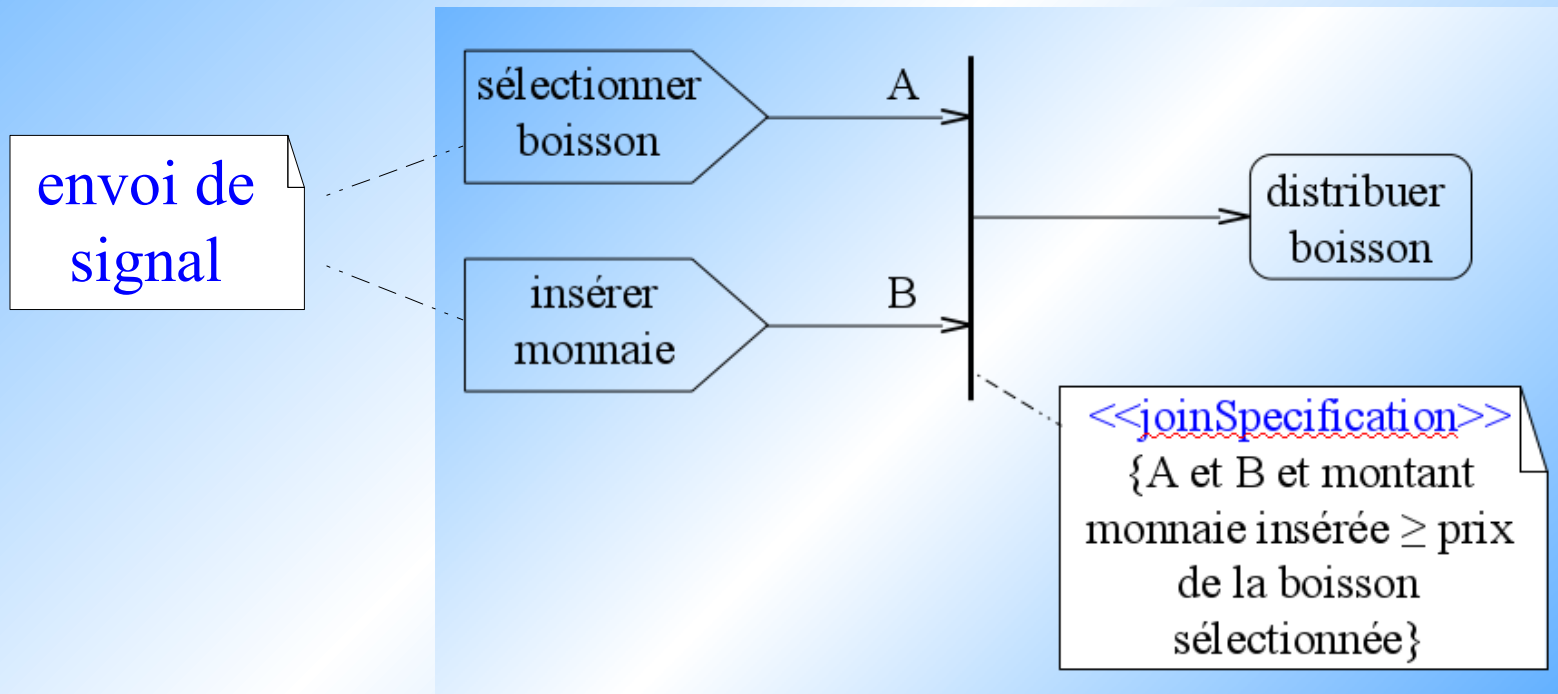


Que se passe-t-il lorsque les actions suivantes se terminent ?

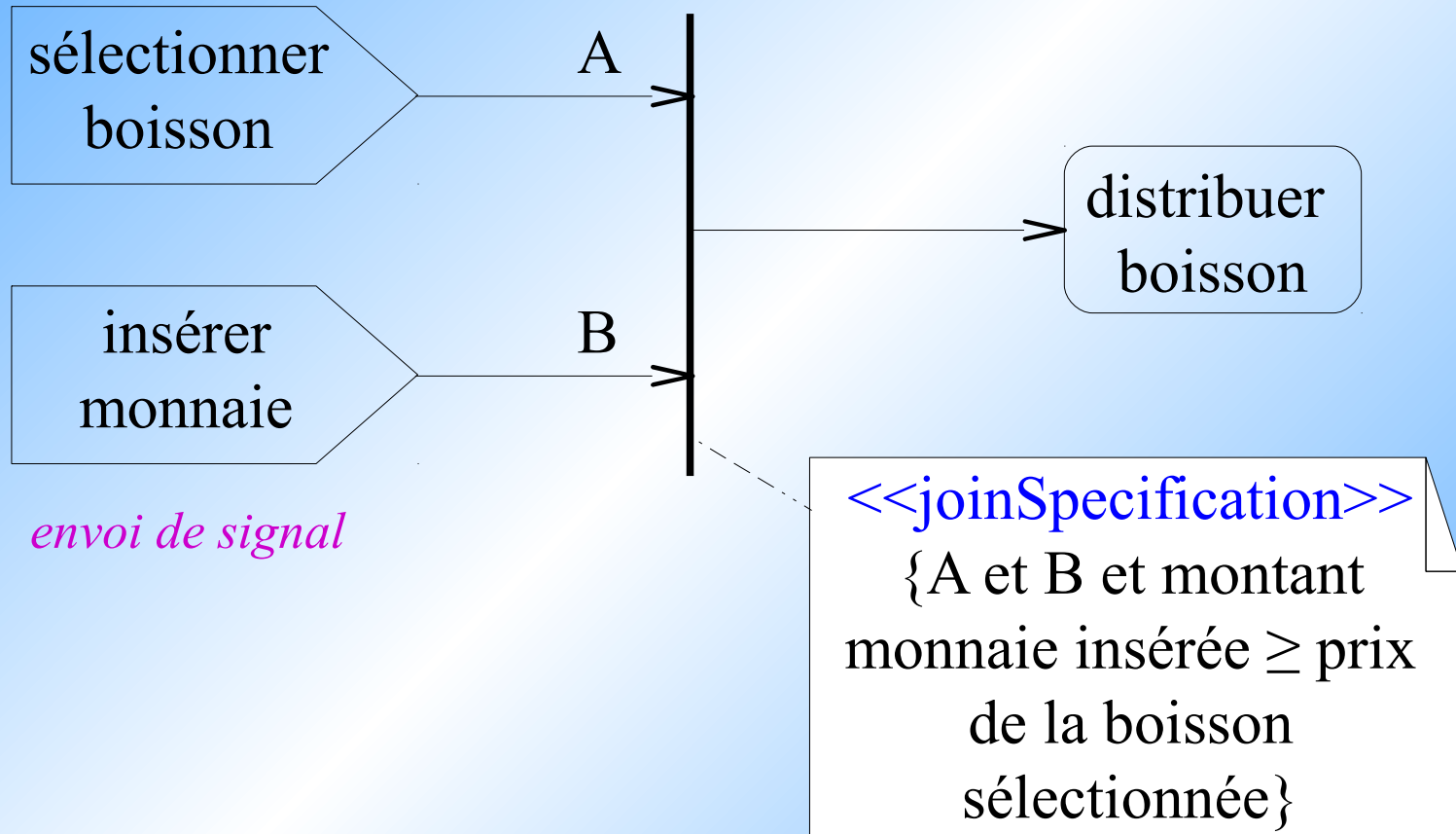
(action1), (action2), (action3),
(action1 et action2),
(action1 et action2 et action3)

NŒUD D'UNION

- Le comportement par défaut du *join* peut être redéfini par une spécification : *join specification*.
- Il s'agit d'une expression logique utilisant les noms des flots entrants, qui doivent donc être nommés pour l'occasion.

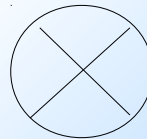


NŒUD D'UNION

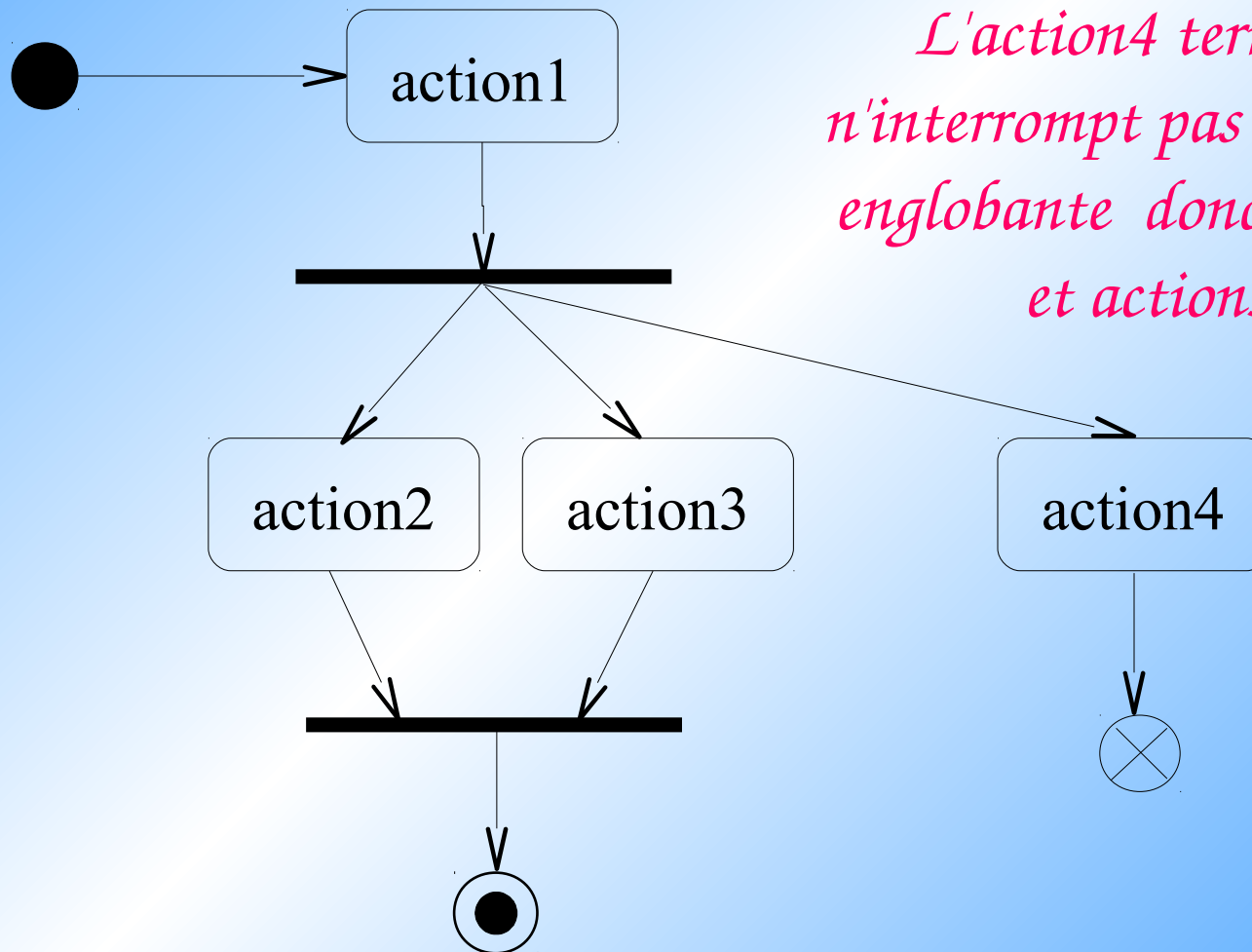


NŒUD DE FIN DE FLOT

- 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.

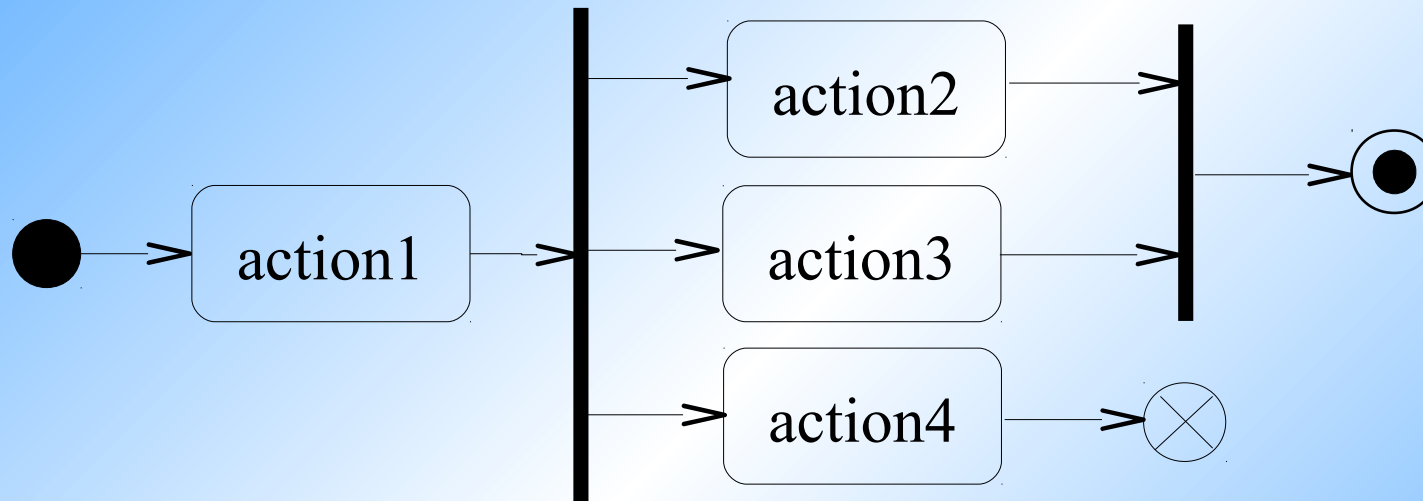


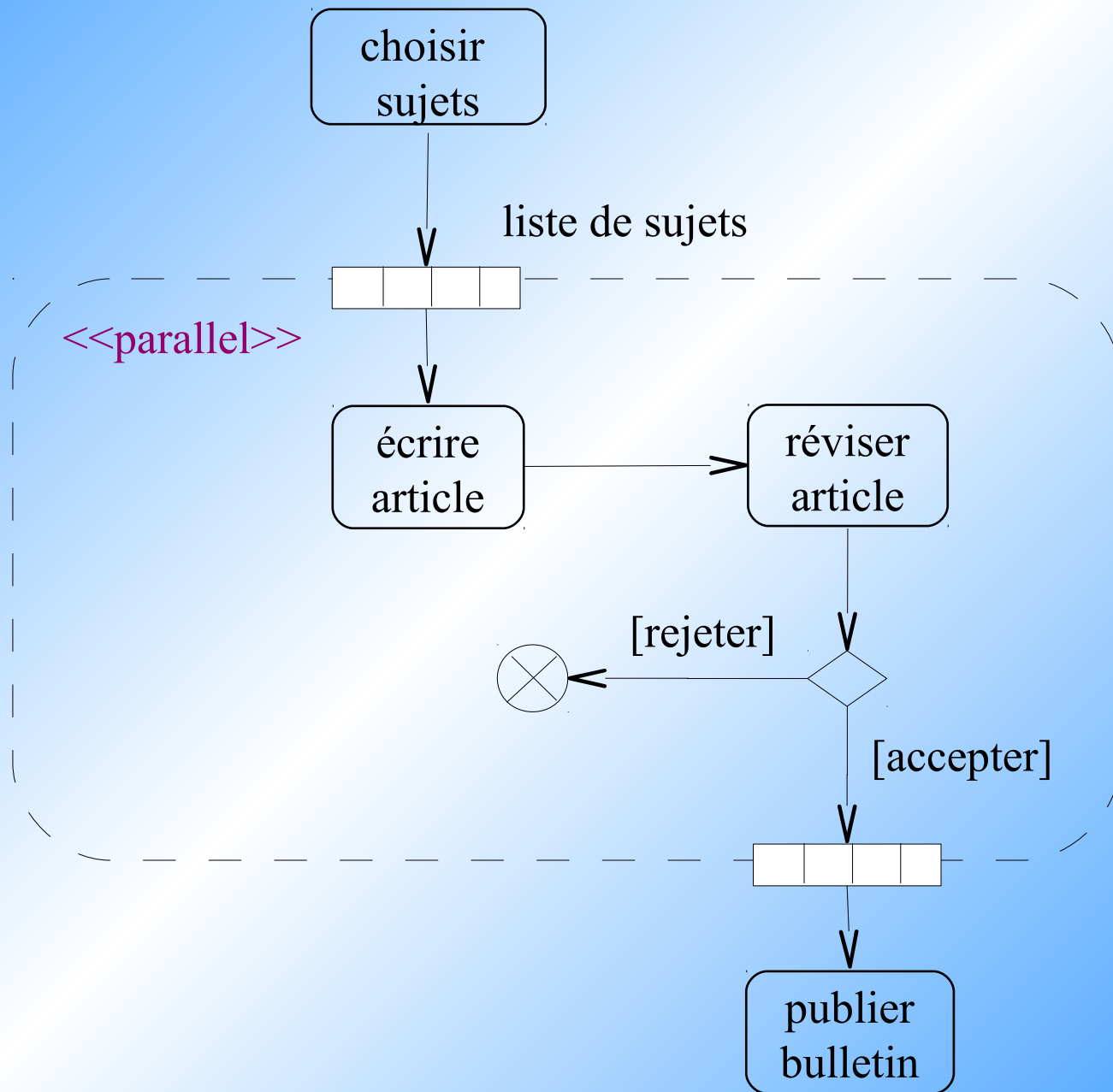
NŒUD DE FIN DE FLOT



*L'action4 terminée
n'interrompt pas l'activité
englobante donc action2
et action3*

NŒUD DE FIN DE FLOT



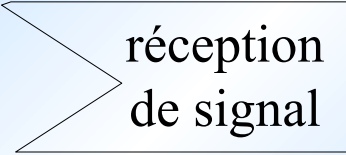


SIGNAUX

- Les actions peuvent répondre à des signaux et en émettre.

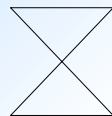


envoi de
signal



réception
de signal

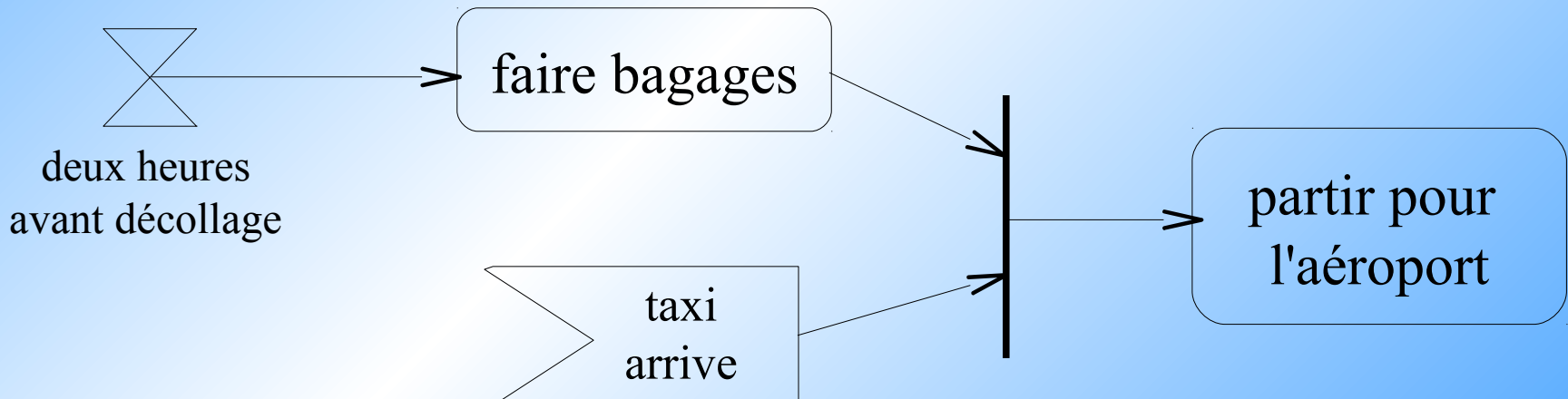
- On peut également avoir des signaux temporels.



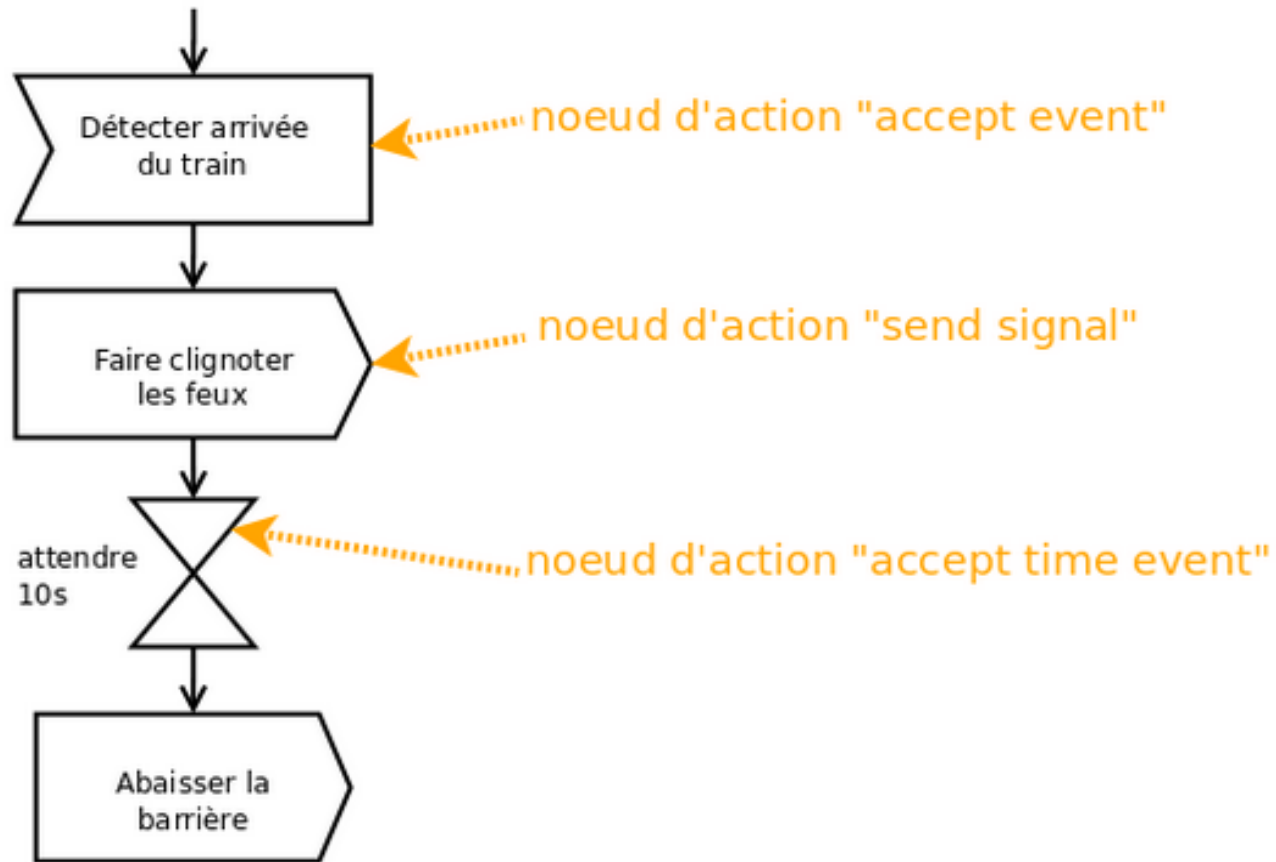
at ()

SIGNAUX

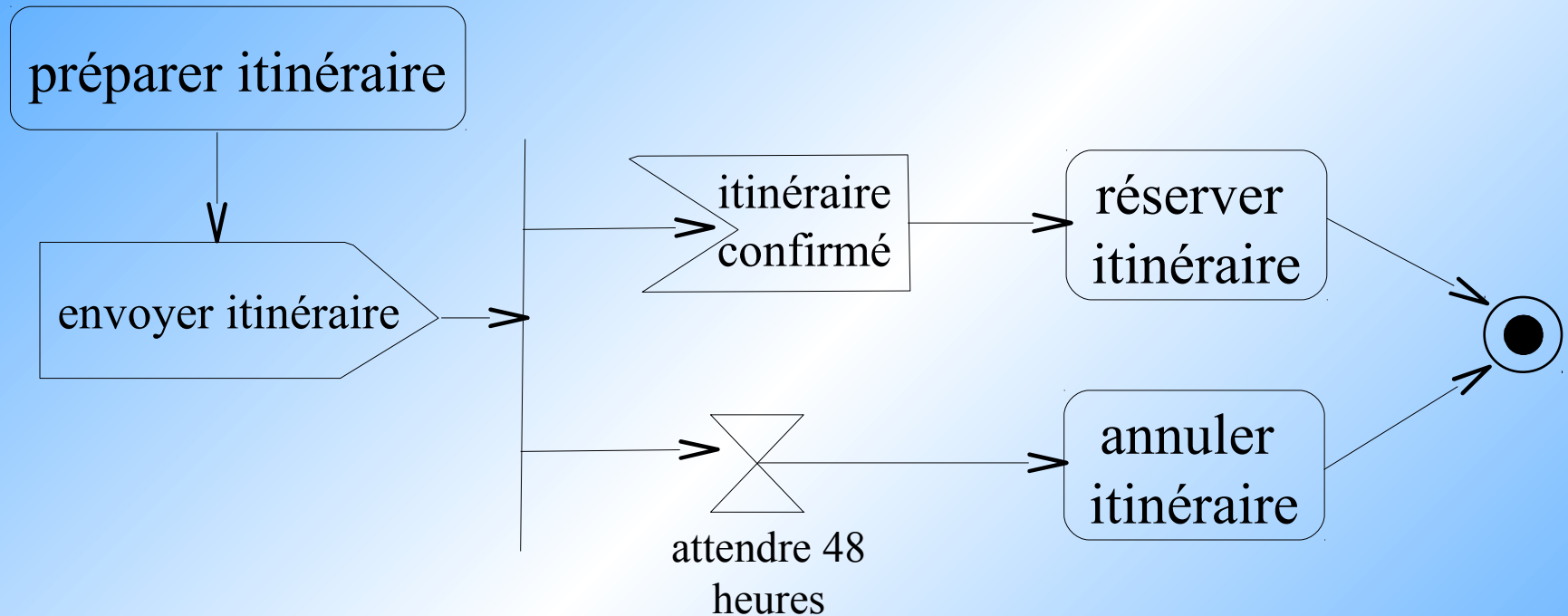
- Les **signaux temporels** sont liés au **passage du temps**.
- Ils peuvent par exemple indiquer des fins de mois dans une année fiscale ou des microsecondes dans un contrôleur temps réel.



SIGNAUX

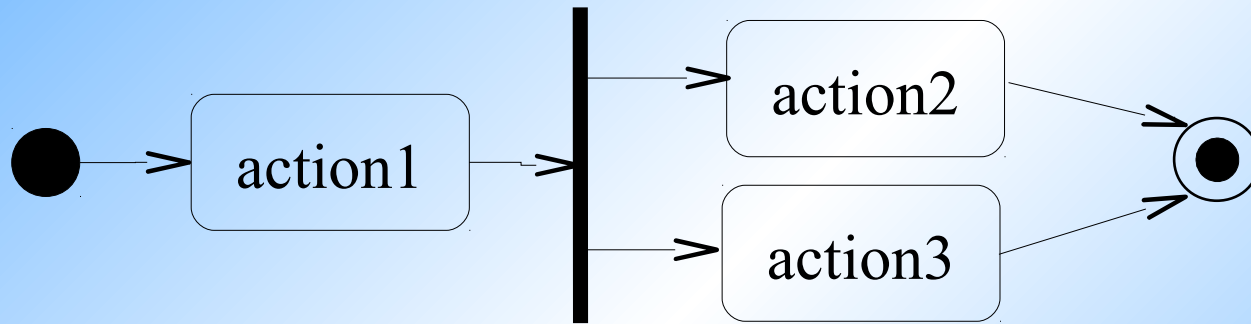


SIGNAUX



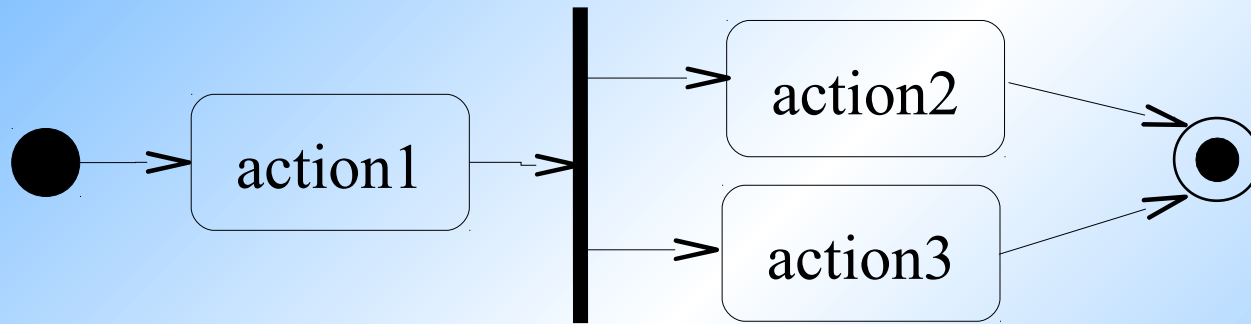
Les deux flots sont en compétition : le premier qui atteindra l'état final gagnera et mettra fin à l'autre.

EXERCICE



L'action2 se termine. Que se passe-t-il ?

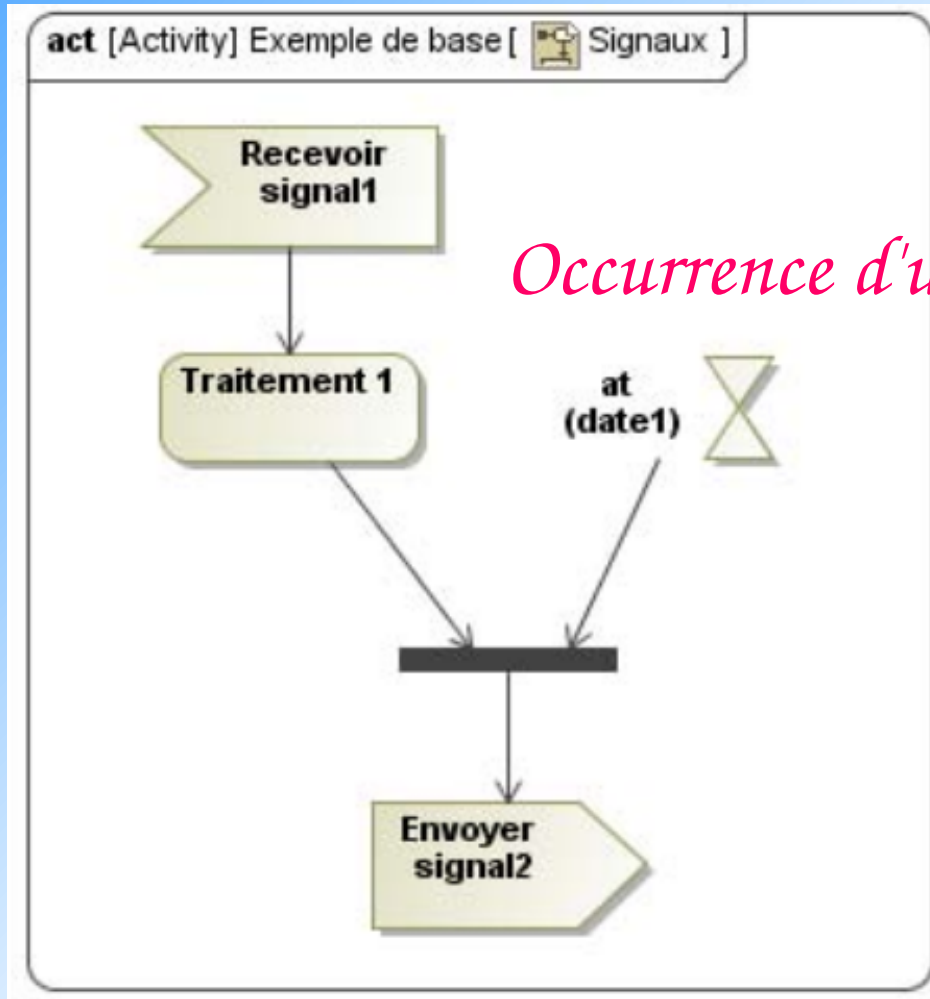
SOLUTION



L'action2 se termine. Que se passe-t-il ?

L'activité se termine.

SIGNAUX



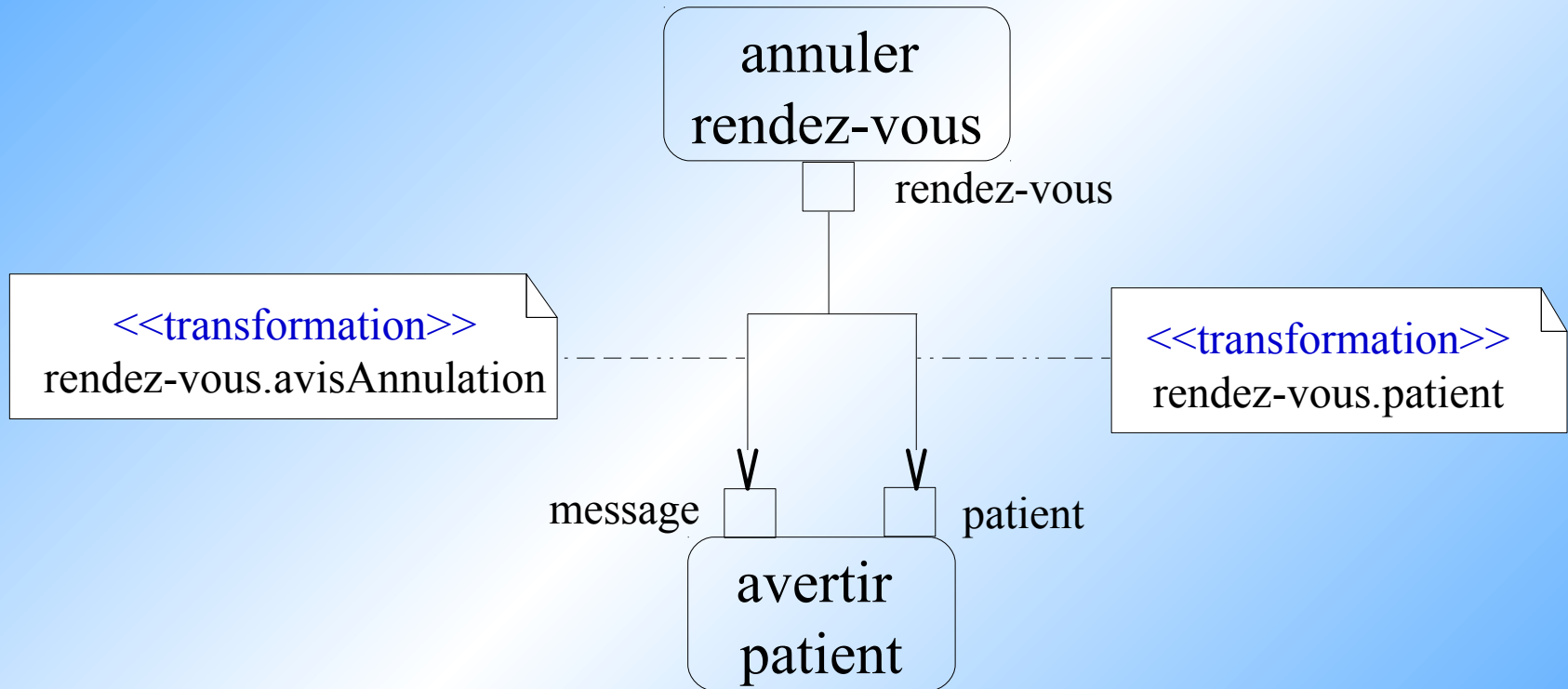
Occurrence d'une date

*On trouve aussi after
expiration d'un timer*

CONNECTEURS

- Les actions (*tout comme les méthodes*) peuvent avoir des **paramètres**. Pour cela, on utilise des **connecteurs**.
- On doit veiller à ce que, pour une action sortante, **les paramètres de sortie correspondent aux paramètres d'entrée** d'une autre action.
- **Si pas de correspondance**, on doit indiquer une **transformation**.
- **Pas d'obligation de représenter les connecteurs**. Uniquement si on veut visualiser les données dont les différentes actions ont besoin et ce qu'elles produisent.

CONNECTEURS



NŒUD D'OBJET

*Ce diagramme sert non seulement à préciser la séquence d'actions
mais aussi*

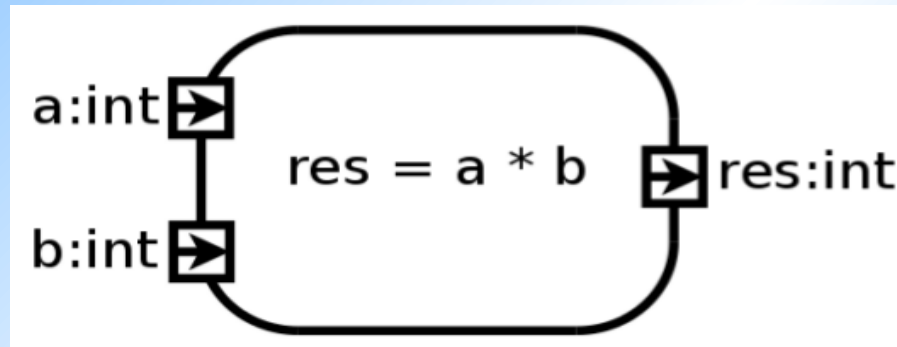
*ce qui est produit,
consommé ou
transformé*

au cours de l'exécution de cette activité.

NŒUD D'OBJET

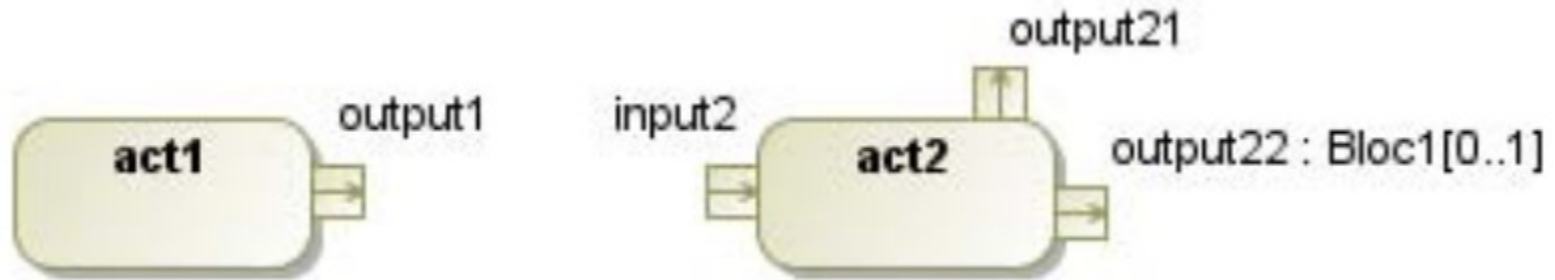
- Jusqu'ici, nous avons montré comment modéliser le comportement du flot de contrôle dans un diagramme d'activité.
- Or, les flots de données n'apparaissent pas et sont pourtant un élément essentiel des traitements (*arguments des opérations, valeurs de retour...*).
- Un nœud d'objet permet de définir un flot d'objets (*i.e. un flot de données*) dans un diagramme d'activité.
- Ce nœud représente l'existence d'un objet généré par une action dans une activité et utilisé par d'autres actions.

BROCHES OU PUNAISES OU PINS D'ENTRÉE / SORTIE

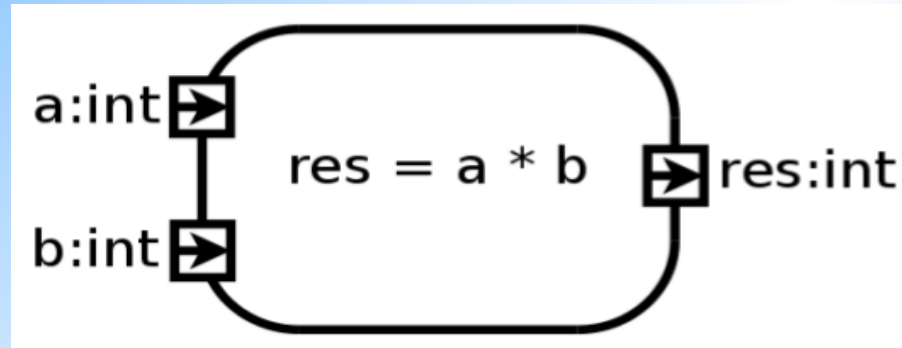


- L'action débute uniquement si affectation d'une valeur à chacune de ses punaises ou pins d'entrée ou broches d'entrée.
- Quand l'action est terminée, une valeur doit être affectée à chacune de ses punaises de sortie.
- Flèche uniquement si c'est nécessaire (*ambiguïté*).

BROCHES OU PUNAISES OU PINS D'ENTRÉE / SORTIE

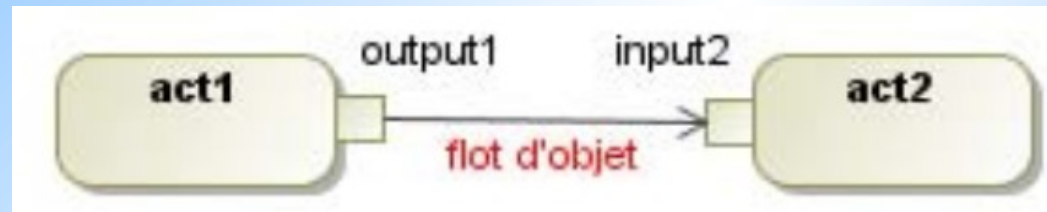
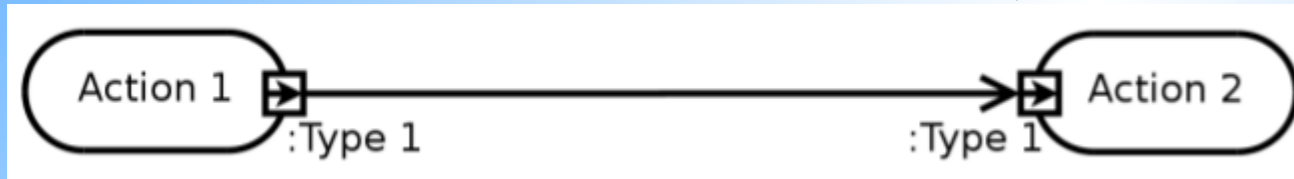


PIN DE VALEUR



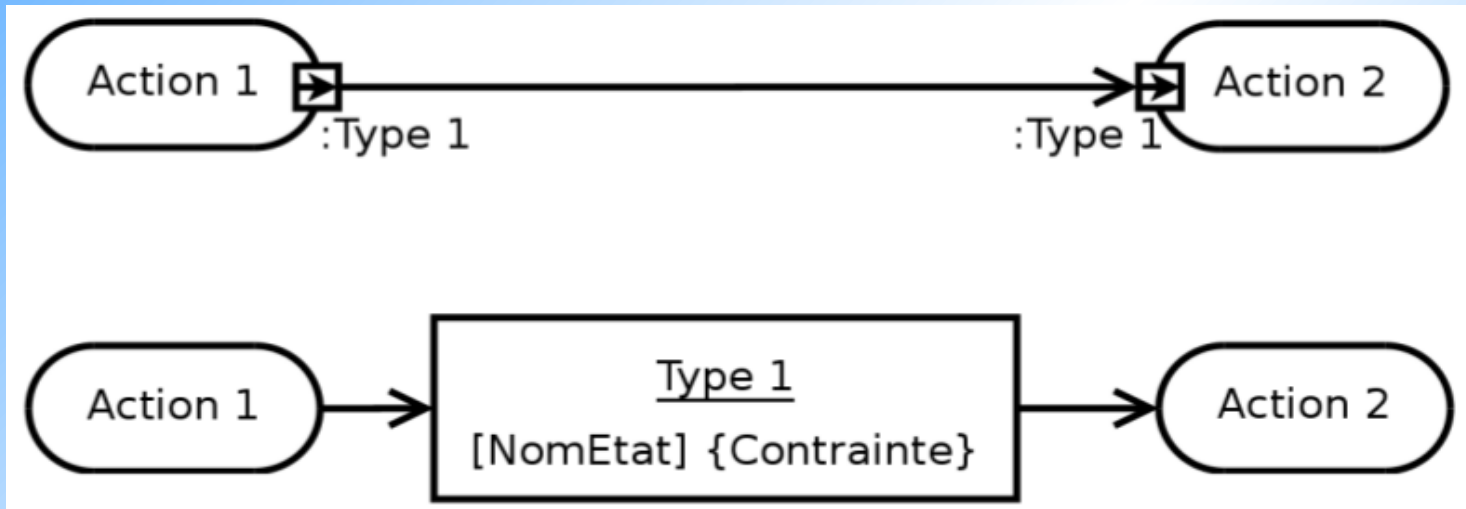
- **Pin de valeur** : pin d'entrée qui fournit une valeur à une action sans que cette valeur ne provienne d'un arc de flot d'objet.
- Pin de valeur toujours associée à une valeur spécifique.
- **Représentation** : pin d'entrée avec la valeur associée écrite à proximité.

FLOT D'OBJET



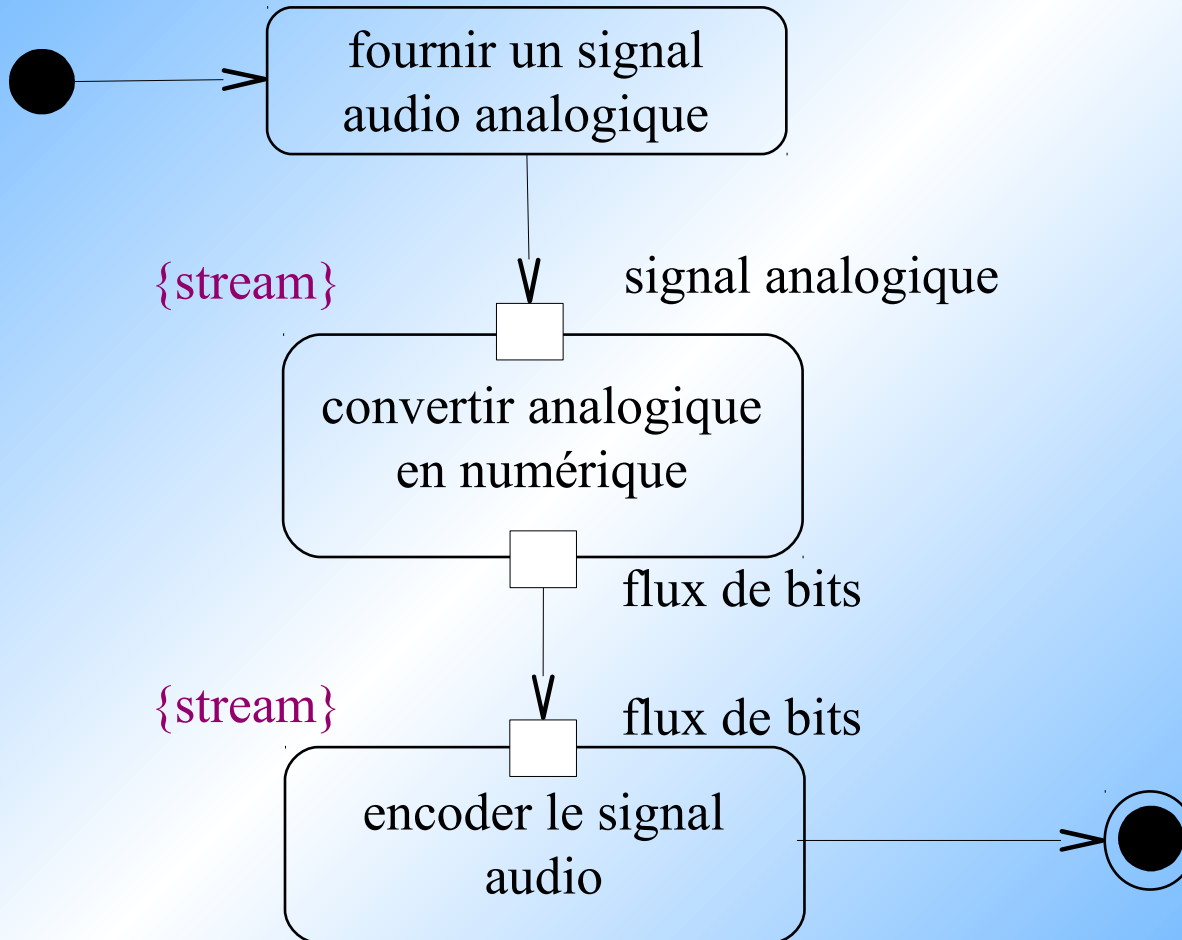
- Un flot d'objet permet de passer des données d'une action à une autre.
- Un arc reliant une punaise de sortie à une punaise d'entrée est un flot d'objets.
- Dans cette configuration, le type de la punaise réceptrice doit être identique ou parent (*au sens de la relation de généralisation*) du type de la punaise émettrice.

FLOT D'OBJET

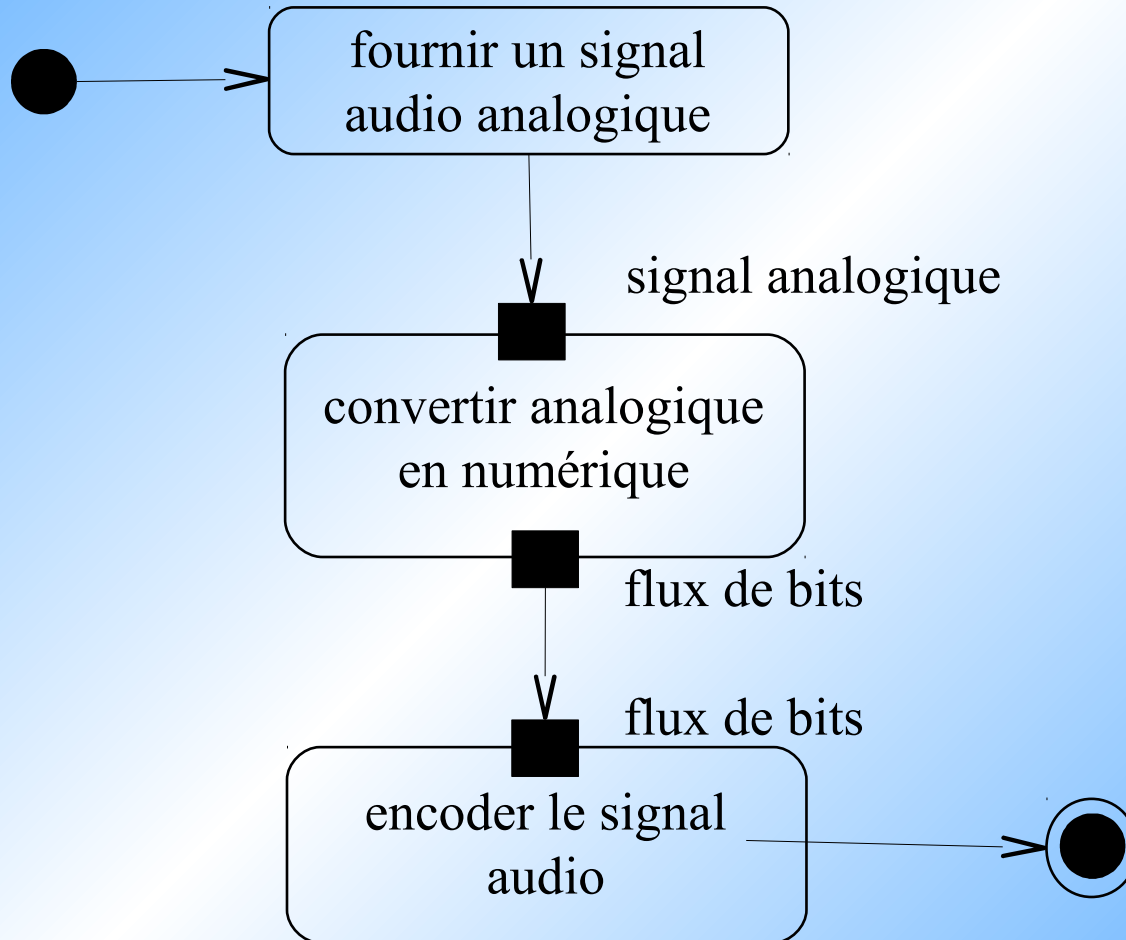


- Il existe une autre représentation possible d'un flot d'objet, plus axée sur les données car elle fait intervenir un nœud d'objet.

ad encoder un signal audio

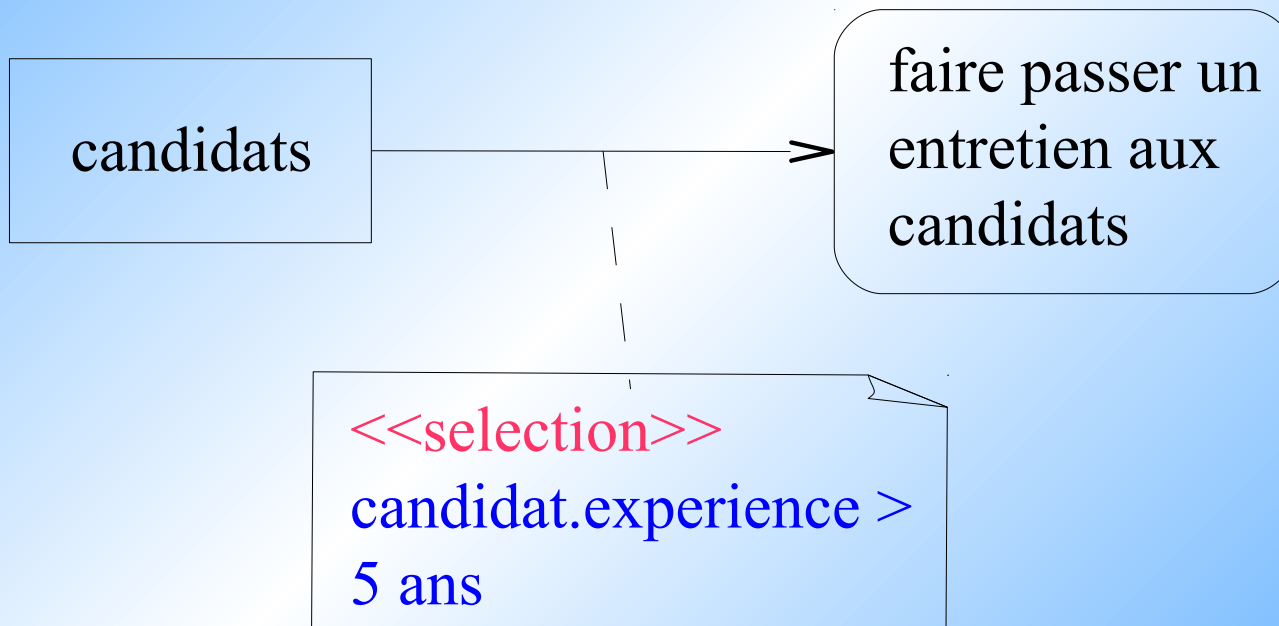


ad encoder un signal audio



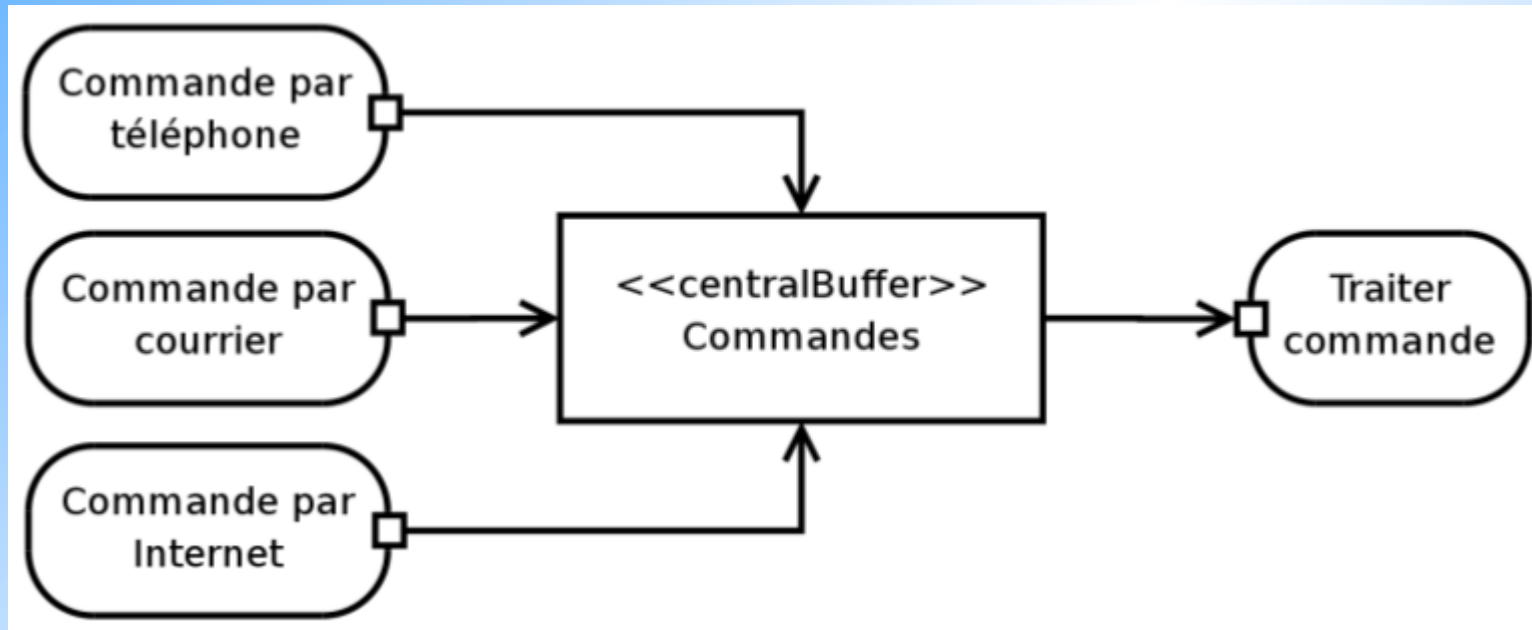
FLUX D'OBJETS : SÉLECTION

faire passer un entretien aux candidats



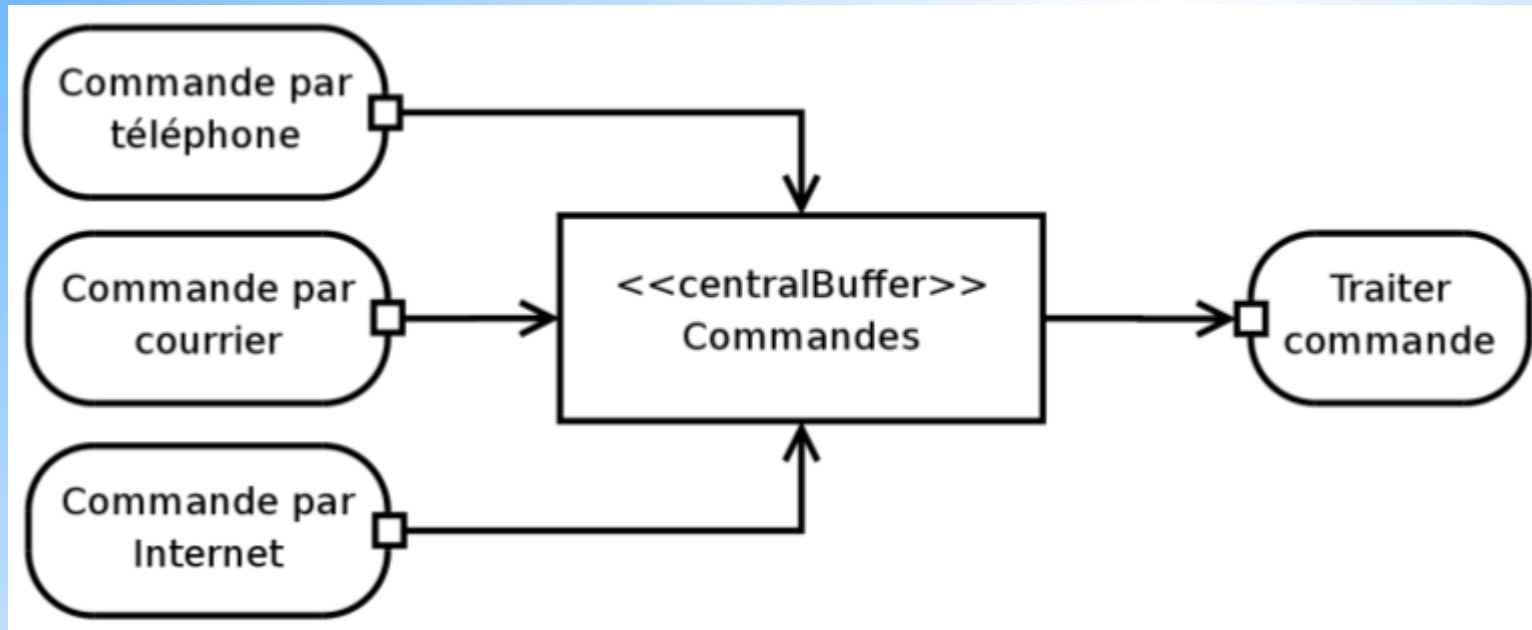
Sélection

NŒUD TAMPON CENTRAL



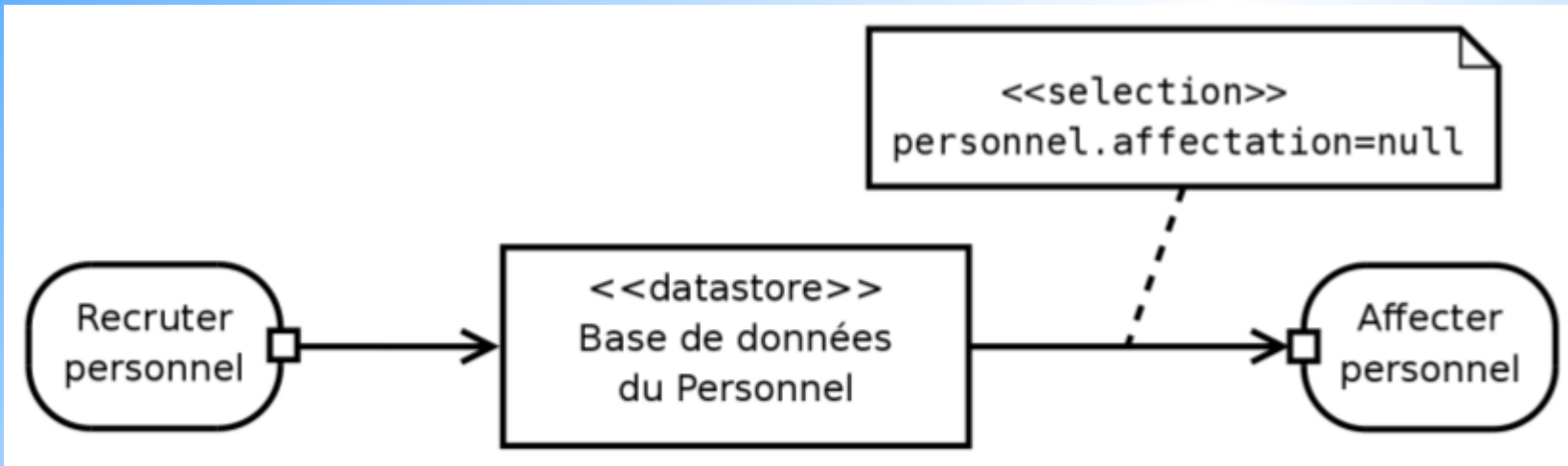
Centralisation de toutes les commandes prises par différents procédés, avant qu'elles ne soient traitées.

NŒUD TAMPON CENTRAL



- Un nœud tampon central est un nœud d'objet qui accepte :
 - les entrées de plusieurs nœuds d'objets ou
 - produit des sorties vers plusieurs nœuds d'objets.

NŒUD DE STOCKAGE DES DONNÉES



Le personnel, après avoir été recruté par l'activité **Recruter personnel**, est stocké de manière persistante dans le nœud de stockage **Base de donnée du Personnel**.

Chaque employé qui n'a pas encore reçu d'affectation (étiquette stéréotypée **<<selection>> : personnel.affectation=null**) est disponible pour être utilisé par l'activité **Affecter personnel**.

NŒUD DE STOCKAGE DES DONNÉES

- **Nœud de stockage des données** = nœud tampon central particulier qui assure la persistance des données.
- Lorsqu'une information est sélectionnée par un flux sortant, l'information est dupliquée et ne disparaît pas du nœud de stockage des données comme ce serait le cas dans un nœud tampon central.
- Lorsqu'un flux entrant véhicule une donnée déjà stockée par le nœud de stockage des données, cette dernière est écrasée par la nouvelle.

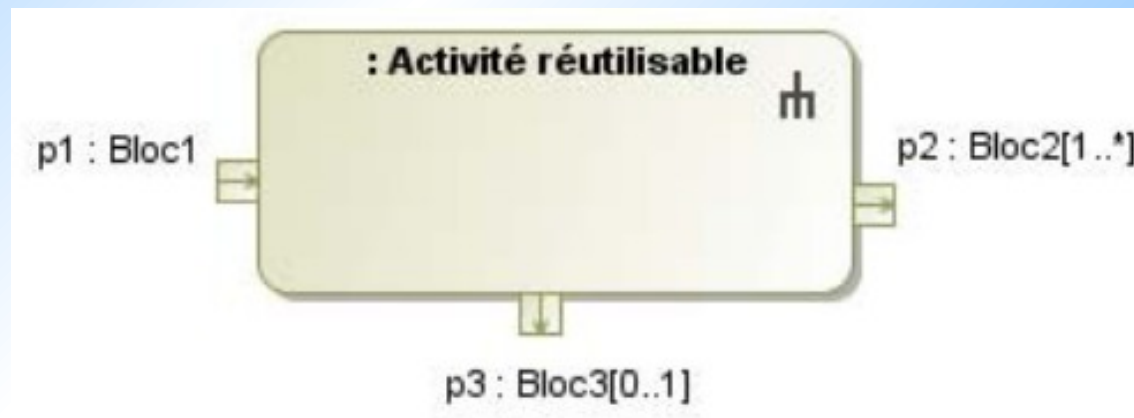
NŒUD D'ACTIVITÉ STRUCTURÉE

- Les activités fournissent le contexte dans lequel les actions s'exécutent.
- Mais les activités elles-mêmes peuvent être utilisées et surtout réutilisées à travers des actions d'appel (*call BehaviorAction*).
Il est ainsi possible de se constituer des bibliothèques d'activités réutilisables, décrivant des algorithmes classiques dans le contexte métier.
- Il est également possible de décrire des traitements très complexes de façon descendante avec un nombre de niveaux de décomposition arbitraire.

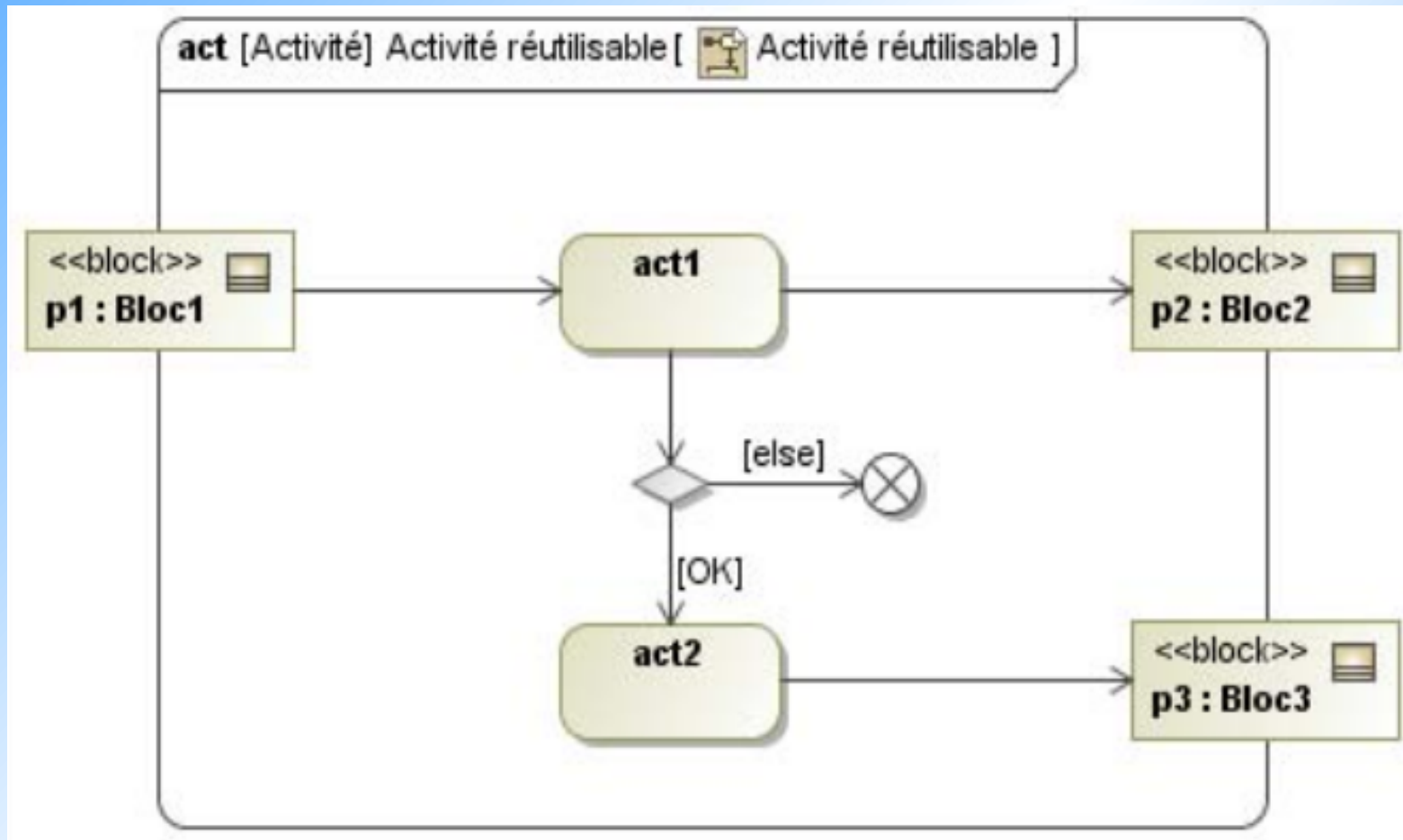
NŒUD D'ACTIVITÉ STRUCTURÉE

- L'action d'appel est représentée graphiquement par une fourche à droite et en bas de la boîte d'action, ainsi que par la chaîne :

nom d'action : nom d'activité



NŒUD D'ACTIVITÉ STRUCTURÉE



Décomposition de l'activité précédente

ZONE D'EXPANSION

- Une **zone** ou région d'**expansion** est utilisée lorsque la sortie d'une action déclenche plusieurs invocations d'une autre action.
- Une région d'expansion marque une zone dans laquelle des actions se déroulent une fois pour chaque élément d'une collection.

ZONE D'EXPANSION

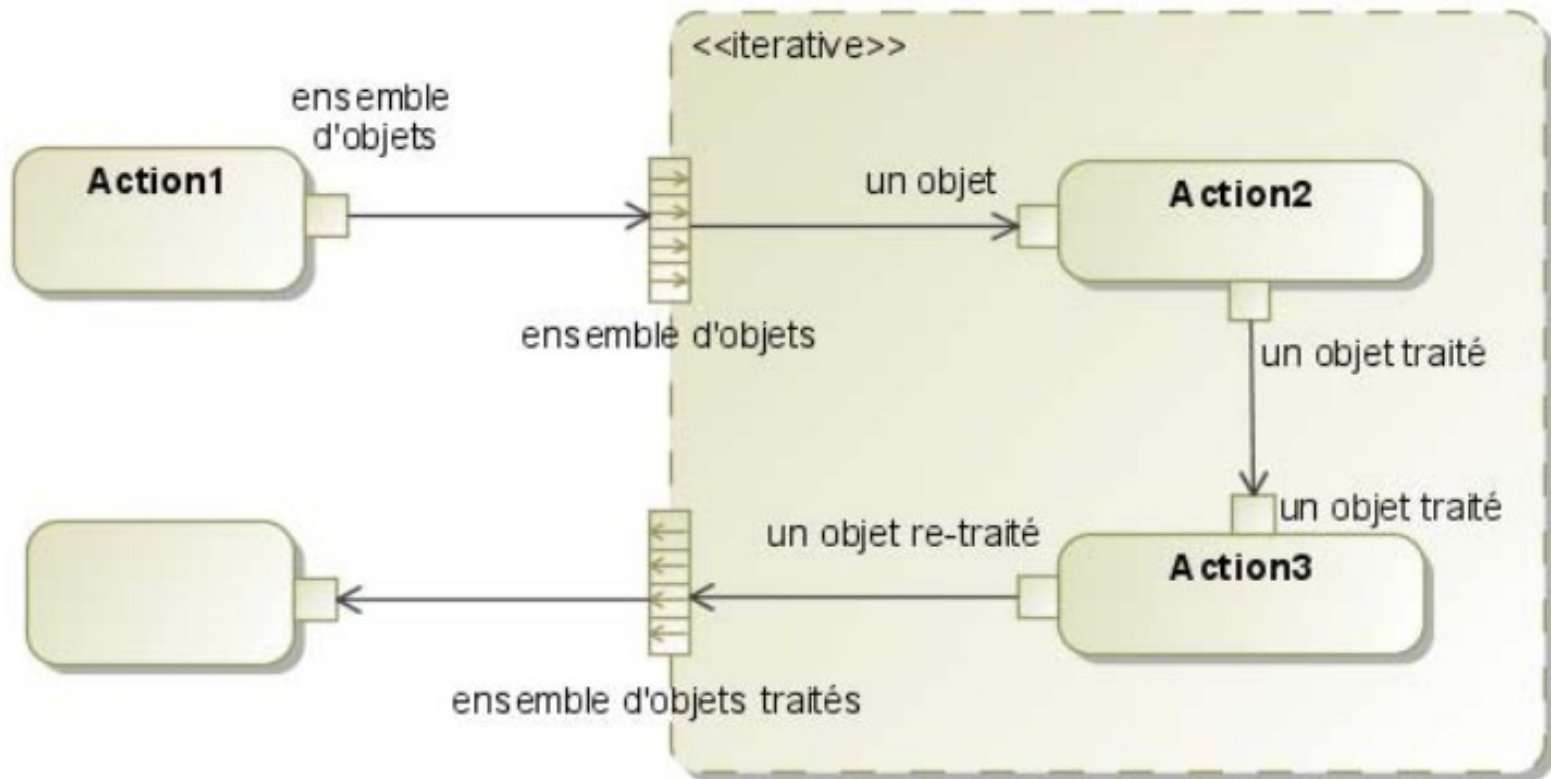
- Une **zone** ou région d'**expansion** est un nœud dans une activité qui accepte un ensemble d'objets en entrée, les traite individuellement et finalement les retourne en sortie tous traités.
- Cela permet de représenter finement le traitement de chaque objet d'un ensemble, dans le cas d'un algorithme commun.
- La région d'expansion se représente comme la région interruptible, mais avec un port multiple (*quatre carrés au lieu d'un*).

ZONE D'EXPANSION

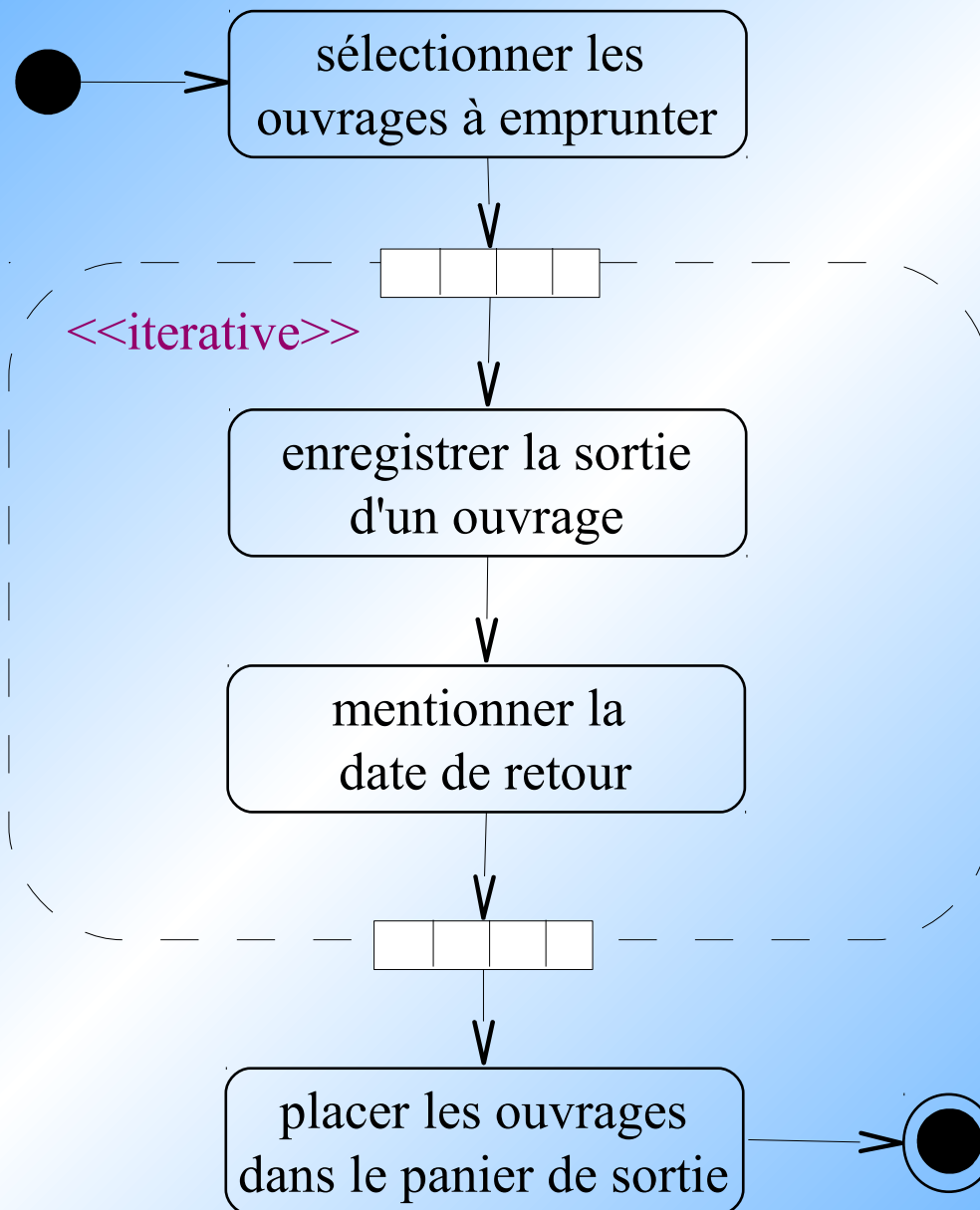
- **Zone d'expansion** = nœud d'activité structurée qui s'**exécute une fois pour chacun des éléments d'une collection**.
- La **collection** est **passée** à la zone d'expansion **à travers une punaise** d'expansion représenté graphiquement par une juxtaposition de punaises pour évoquer la notion de collection.
- Chaque exécution de l'activité interne **insère une valeur de retour dans la collection de sortie**.
- La zone d'expansion est stéréotypée **<<parallel>>**, **<<iterative>>** ou encore **<<stream>>** en fonction du comportement souhaité pour l'ordonnancement des exécutions de l'activité interne.

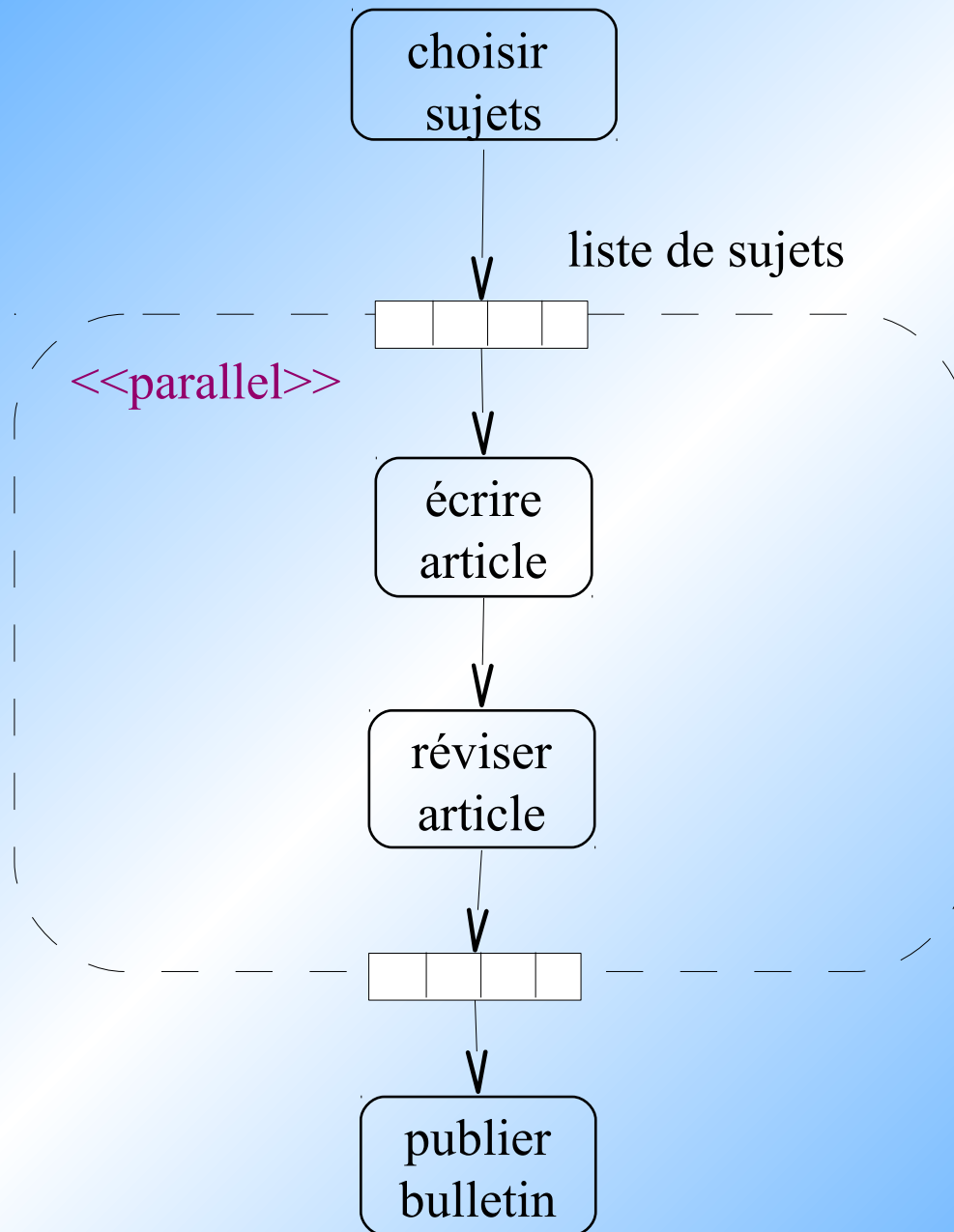
ZONE D'EXPANSION

act [Activité] Exemple de base [ Région d'expansion]



ad enregistrer la sortie des ouvrages





ZONE D'EXPANSION

- **<<parallel>>** (*parallèle*) : les objets de l'ensemble sont traités en parallèle.

adaptée quand les exécutions de l'activité interne sont indépendantes et peuvent être réalisées dans un **ordre indifférent, voire simultanément**.

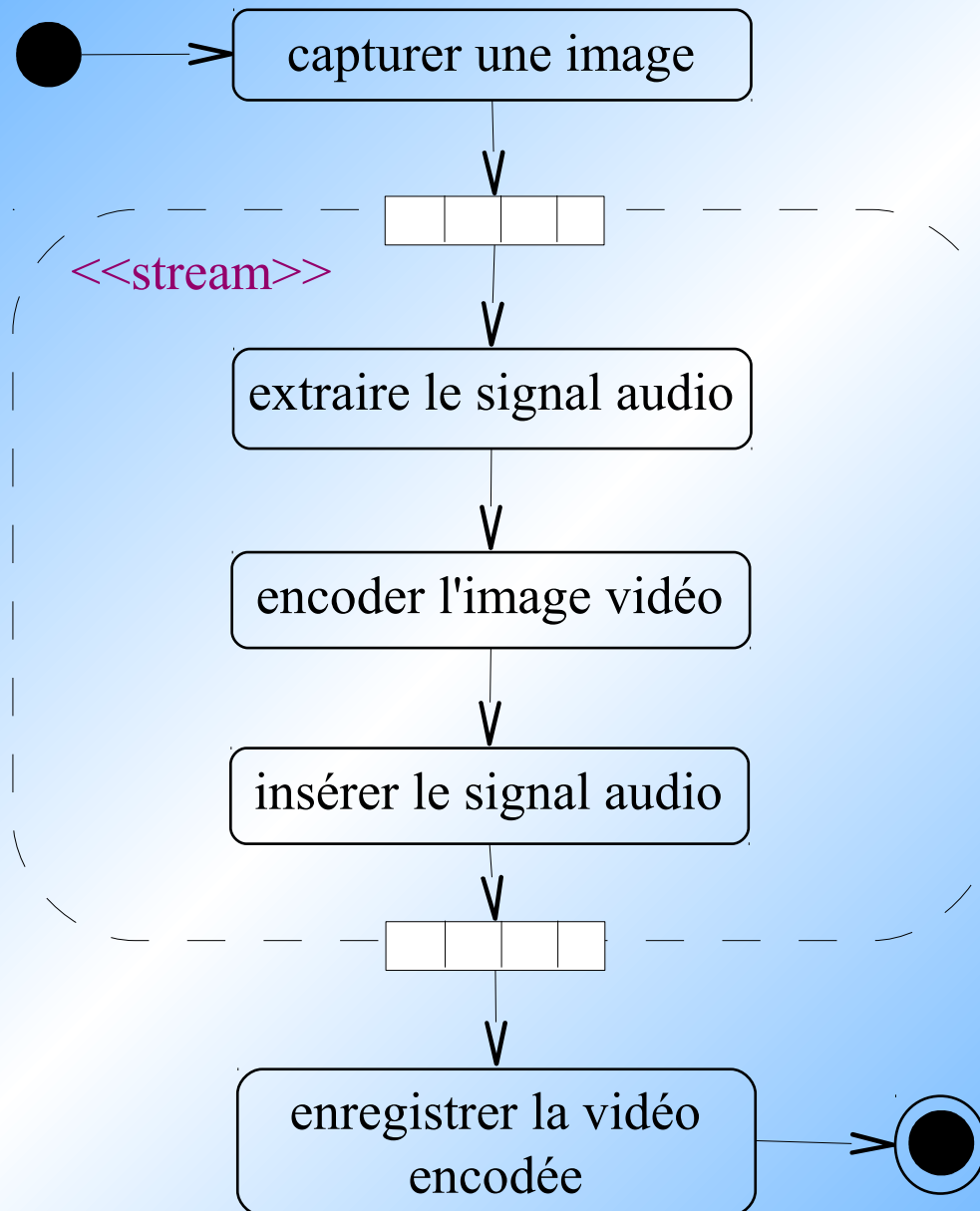
- **<<iterative>>** (*séquentielle*) : les objets de l'ensemble sont traités séquentiellement.

impose que les différentes exécutions de l'activité interne soient réalisées séquentiellement dans l'**ordre des éléments de la collection**.

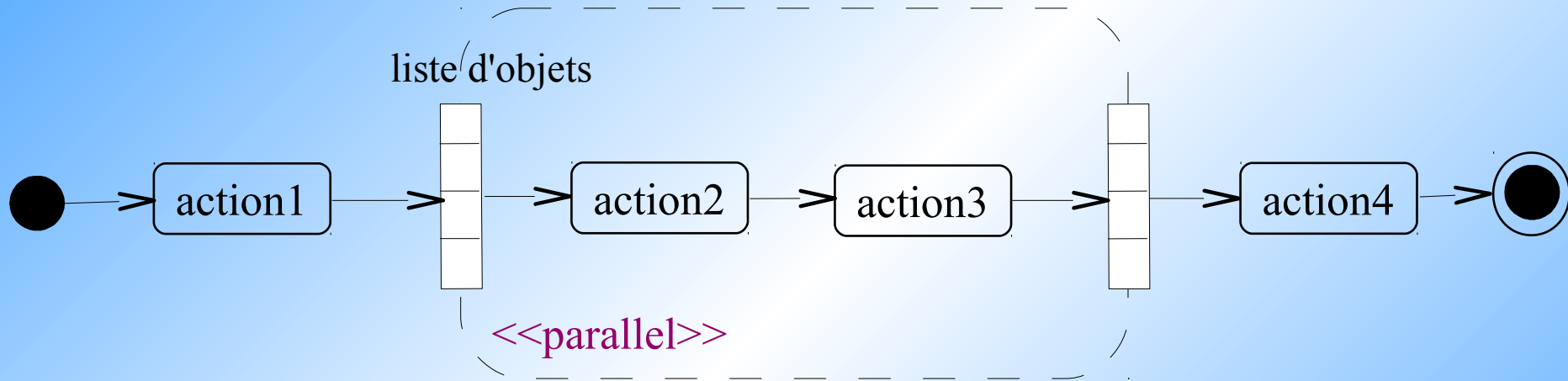
ZONE D'EXPANSION

- **<<stream>>** (*continu*) : les objets de l'ensemble sont traités comme un seul flot, un peu comme sur une chaîne de montage. Chaque objet n'a pas à attendre que le traitement de l'objet précédent soit complètement terminé.
utilisé quand la **collection est un flux de données**.

ad encoder des images vidéo



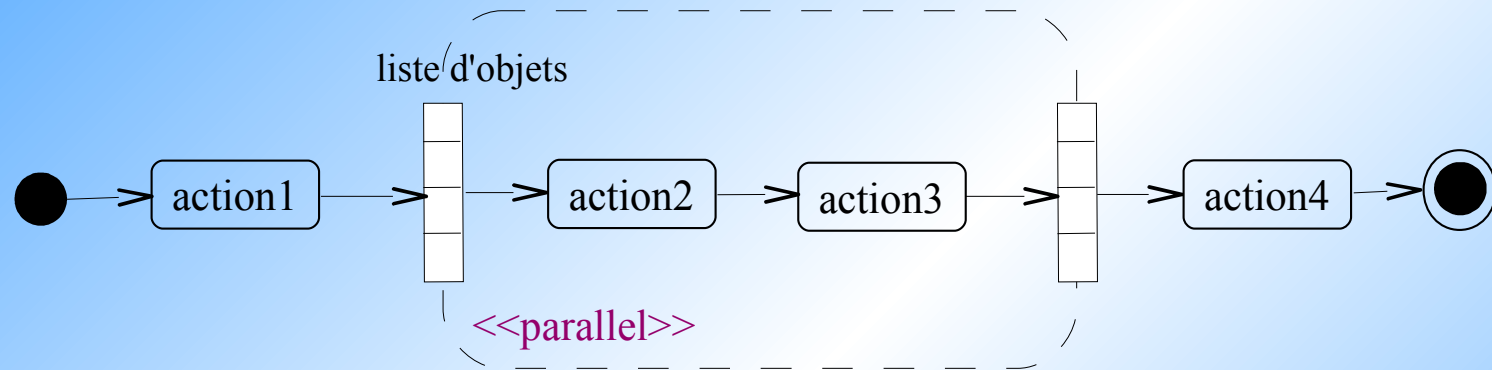
EXERCICE



On suppose que la liste des objets se compose des objets o1 et o2.

Donnez l'ordre d'exécution des actions.

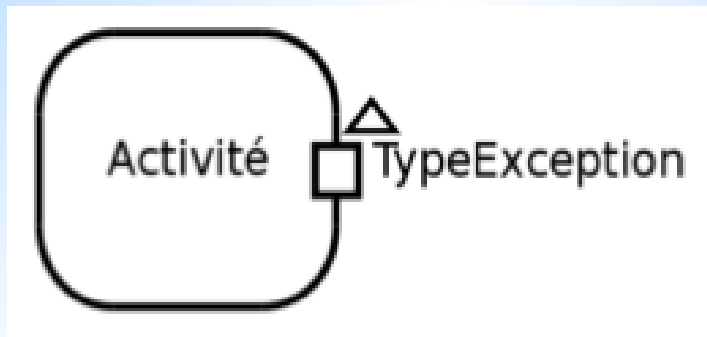
SOLUTION



- ☛ L'action2 est exécutée sur o1, puis l'action3 est exécutée sur o1, puis l'action2 est exécutée sur o2 et pour finir, l'action3 est exécutée sur o2.
- ☛ L'action2 est exécutée sur o1, puis l'action2 est exécutée sur o2, puis l'action3 est exécutée sur o1 et pour finir, l'action3 est exécutée sur o2.
- ☛ L'action2 est exécutée sur o2, puis l'action2 est exécutée sur o1, puis l'action3 est exécutée sur o2 et pour finir, l'action3 est exécutée sur o1.
- ☛ L'action2 est exécutée sur o1, puis l'action2 est exécutée sur o2, puis l'action3 est exécutée sur o2 et pour finir, l'action3 est exécutée sur o1.
- ☛ L'action2 est exécutée sur o2, puis l'action2 est exécutée sur o1, puis l'action3 est exécutée sur o1 et pour finir, l'action3 est exécutée sur o2.

EXCEPTIONS

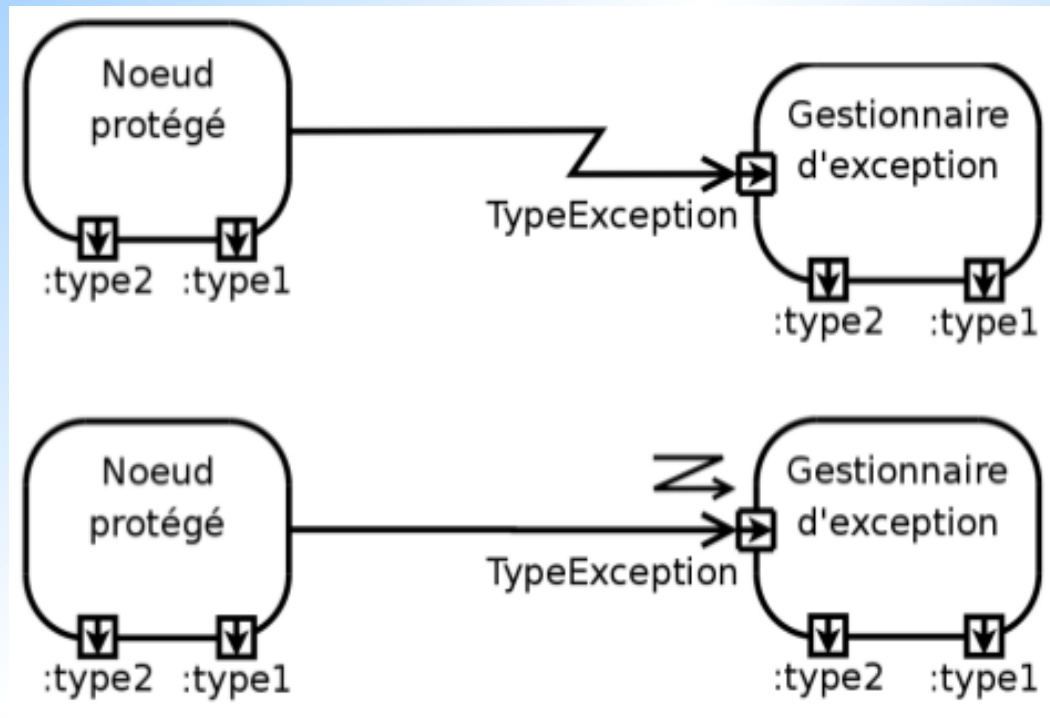
- Une **exception** est générée quand une **situation anormale entrave le déroulement nominal** d'une tâche.
- Elle peut être **générée automatiquement** pour signaler une erreur d'exécution (*débordement d'indice de tableau, division par zéro, . . .*), ou être **soulevée explicitement par une action** (*RaiseException*) pour signaler une situation problématique qui n'est pas prise en charge par la séquence de traitement normale.



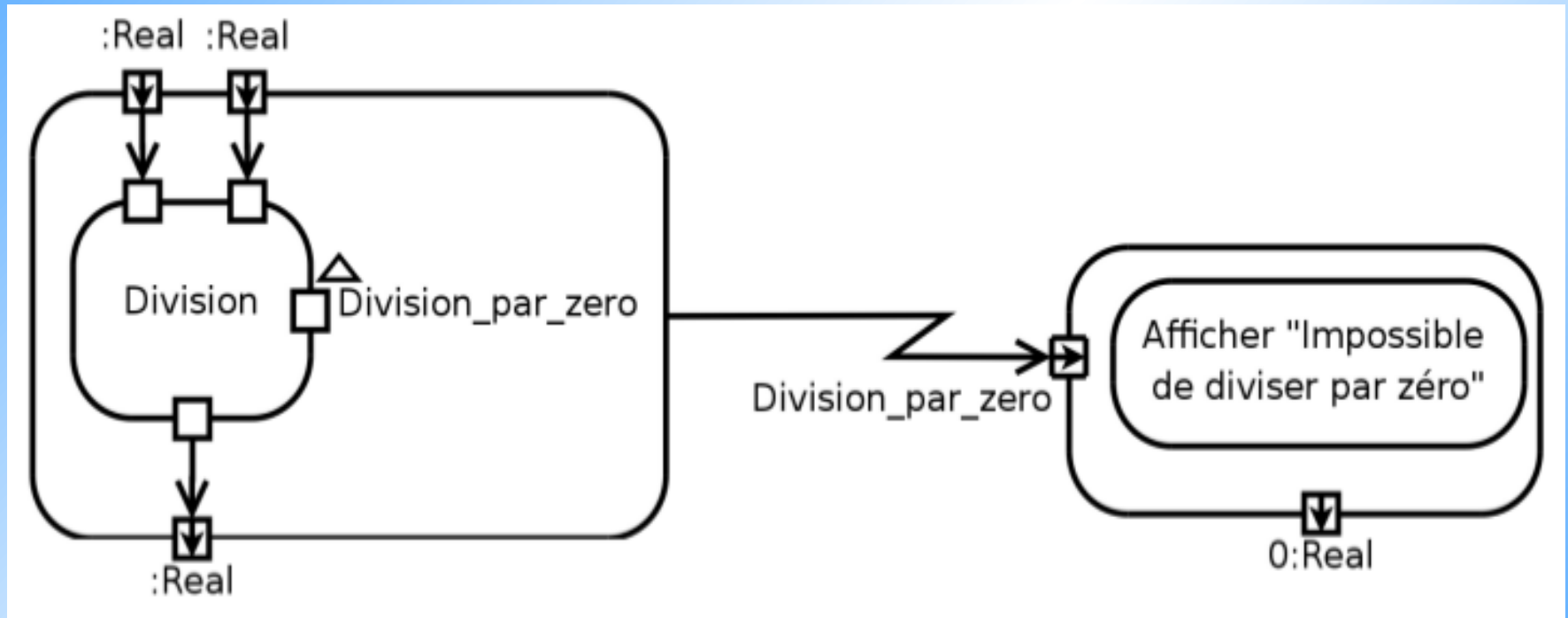
Exemple : division par zéro

GESTIONNAIRE D'EXCEPTION

- Un **gestionnaire d'exception** est une **activité possédant une punaise d'entrée du type de l'exception** qu'il gère et lié à l'activité qu'il protège par un arc en zigzag ou un arc classique orné d'une petite flèche en zigzag.



GESTIONNAIRE D'EXCEPTION



RÉGION INTERRUPTIBLE

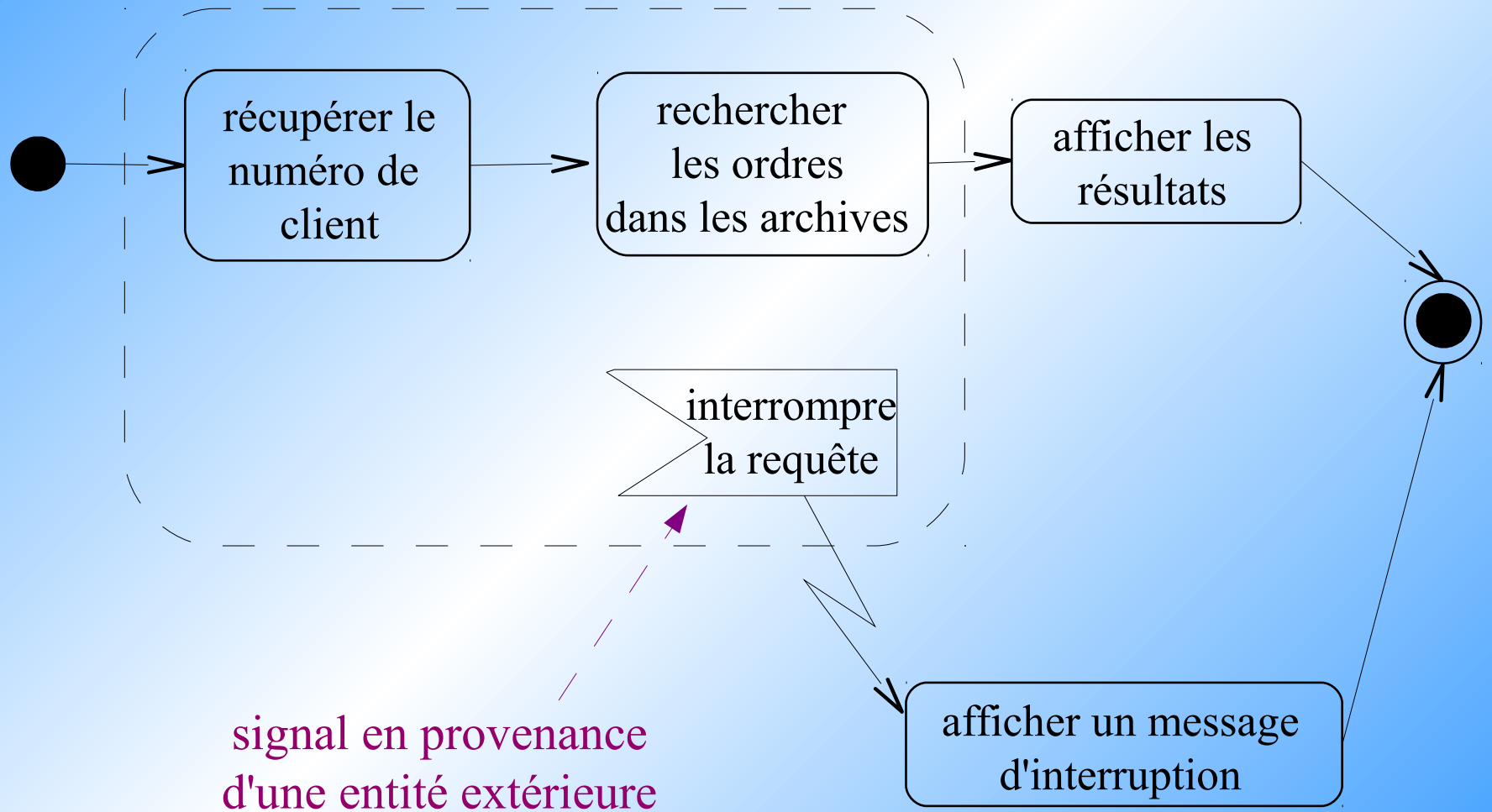
- Il peut arriver qu'on veuille **terminer** seulement **un sous-ensemble des actions** pour une raison particulière.
- UML propose le concept de **région interruptible** pour modéliser cette situation.
- Elle est représentée graphiquement par un rectangle aux bords arrondis en pointillés qui englobe un sous-ensemble des actions de l'activité.
- Il doit exister également un mécanisme d'interruption, typiquement une réception d'événement (*accept event*) suivie d'un flot d'interruption, représenté par un éclair.

RÉGION INTERRUPTIBLE



permet de spécifier un ensemble de nœuds d'action au sein duquel l'activité se termine si un certain événement d'interruption se produit.

ad récupérer les ordres



GESTIONNAIRE D'EXCEPTION

- Lorsqu'une **exception survient**, l'exécution de l'**activité en cours** est **abandonnée sans générer de valeur de sortie**.
- Le mécanisme d'exécution **recherche** alors un **gestionnaire d'exception** susceptible de traiter l'exception levée ou une de ses classes parentes.
- Si l'**activité** qui a levé l'exception n'est **pas protégée** de cette exception, l'**exception est propagée à l'activité englobante**.
- L'exécution de cette dernière est abandonnée, ses valeurs de sortie ne sont pas générées et un **gestionnaire d'exception est recherché** à son niveau.

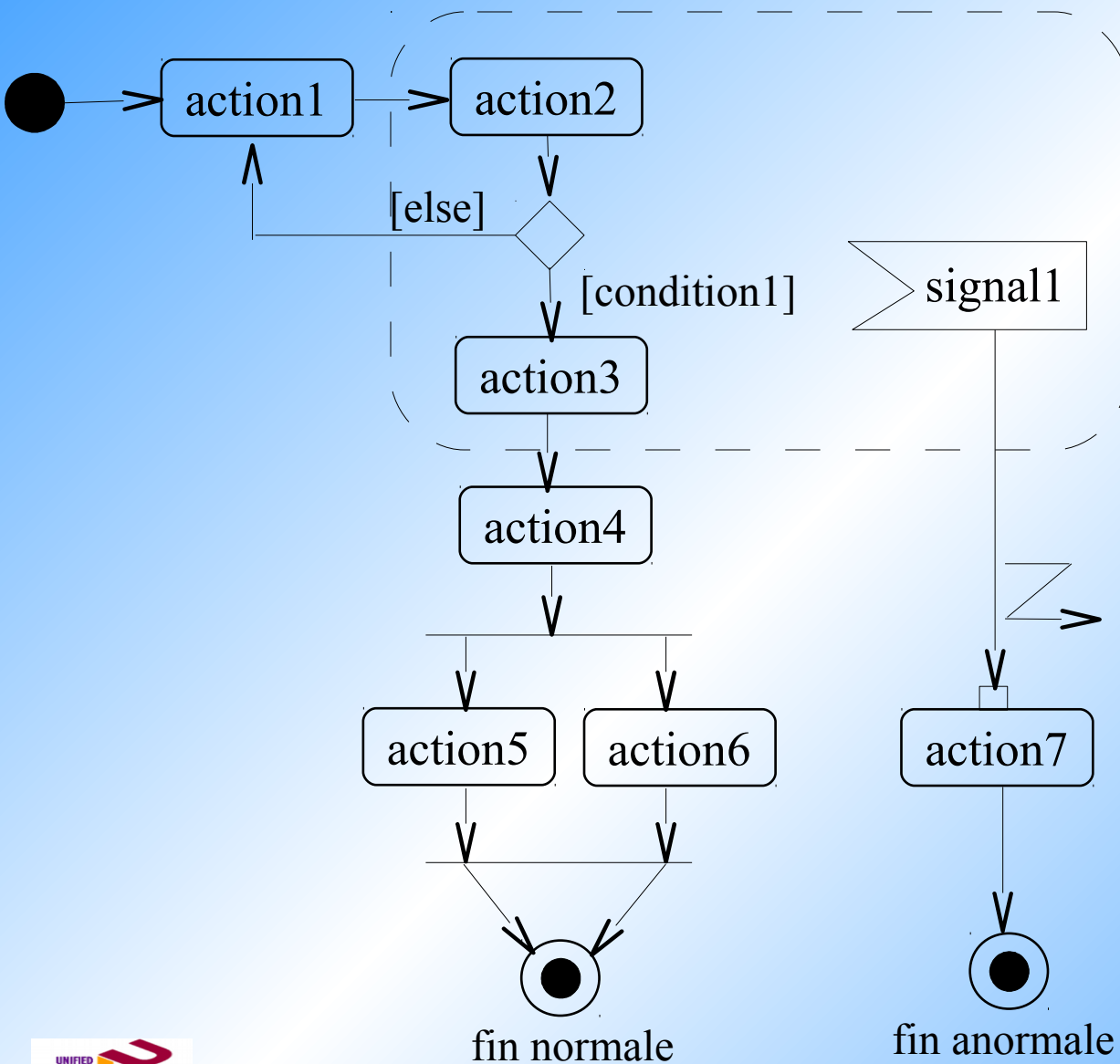
GESTIONNAIRE D'EXCEPTION

- Ce **mécanisme de propagation** se poursuit jusqu'à ce qu'un **gestionnaire adapté** soit trouvé.
- Si l'**exception se propage jusqu'au sommet d'une activité** (*i.e. il n'y a plus d'activité englobante*), trois cas de figure se présentent.
- Si l'activité a été invoquée de manière asynchrone, aucun effet ne se produit et la gestion de l'exception est terminée.
- Si l'activité a été invoquée de manière synchrone, l'exception est propagée au mécanisme d'exécution de l'appelant.
- Si l'exception s'est propagée à la racine du système, le modèle est considéré comme incomplet ou mal formé.

GESTIONNAIRE D'EXCEPTION

- Dans la plupart des langages orientés objet, une exception qui se propage jusqu'à la racine du programme implique son arrêt.
- **Quand un gestionnaire d'exception adapté a été trouvé** et que son exécution se termine, **l'exécution se poursuit comme si l'activité protégée s'était terminée normalement**, les valeurs de sortie fournies par le gestionnaire remplaçant celle que l'activité protégée aurait dû produire.

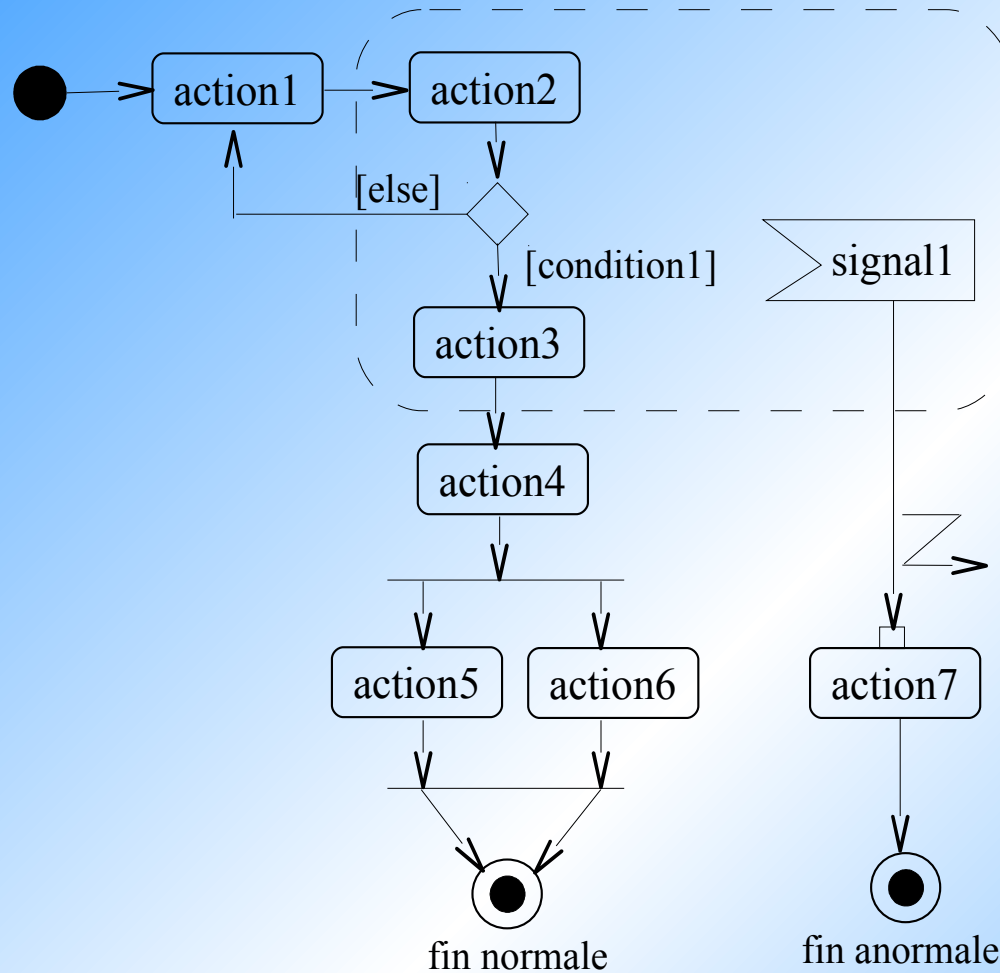
EXERCICE



Le signal signal1 survient.
Que se passe-t-il
dans chacune des actions
suivantes ?

action1
action2
action3
action4

SOLUTION



Le signal signal1 survient.

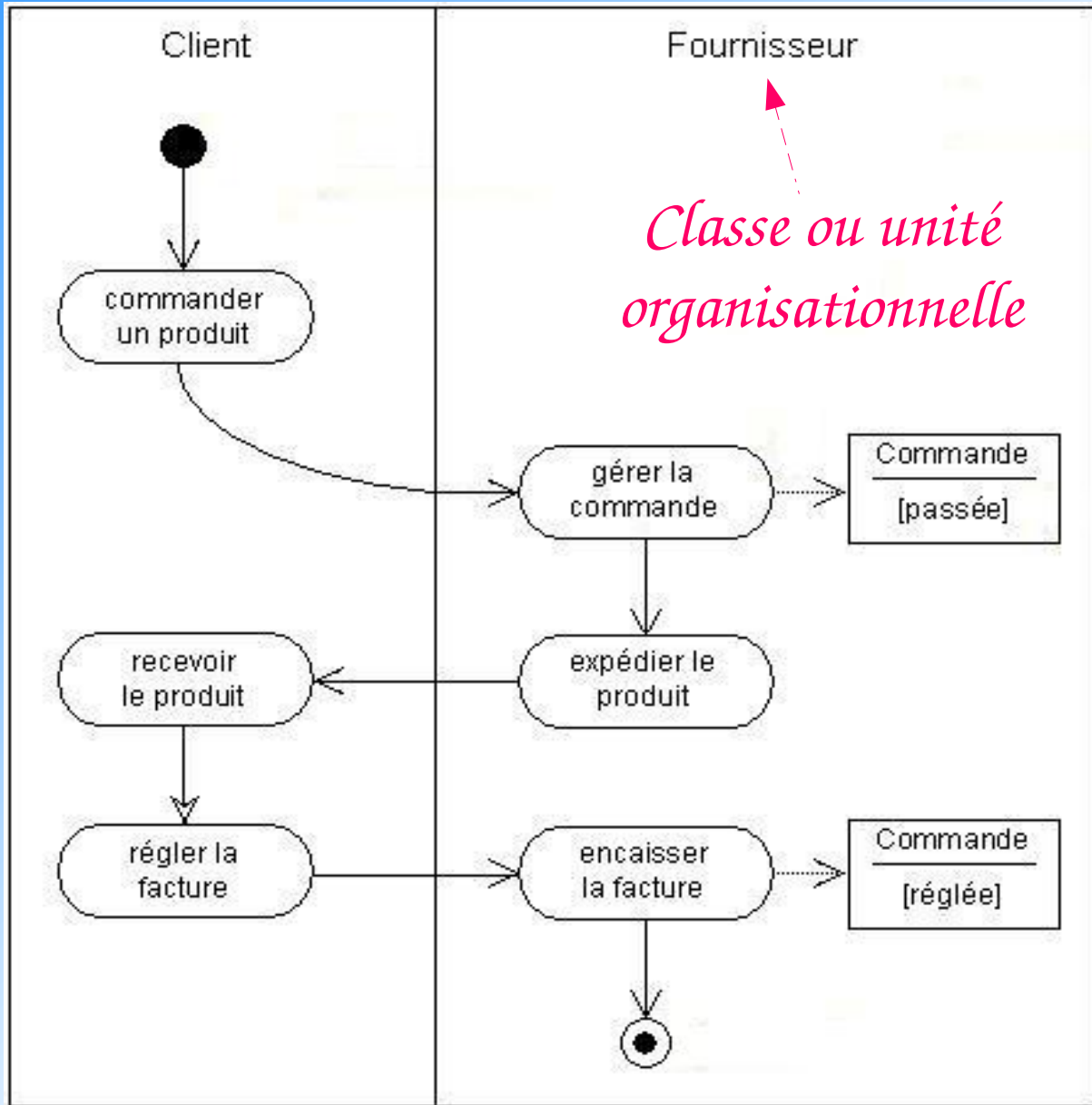
action1 : action2
action2 : action7
action3 : action7
action4 : action5, action6

action en cours

action suivante

PARTITION

- Les diagrammes d'activité **indiquent ce qui se passe mais ne disent pas qui fait quoi.**
- En termes de **programmation**, cela signifie qu'ils ne précisent pas quelle **classe** est responsable de chaque action.
- En termes de modélisation des **processus métier**, ils ne permettent pas de savoir quelle partie de l'**organisation** exécute chaque action.
- Ce n'est pas nécessairement un problème : il est souvent plus significatif de se concentrer sur ce qui est fait que sur les responsables de telle ou telle partie d'un comportement.



*Classe ou unité
organisationnelle*

Les diagrammes d'activité peuvent être découpés en partitions, pour montrer qui est responsable de quoi, au sein d'un mécanisme ou d'une organisation.

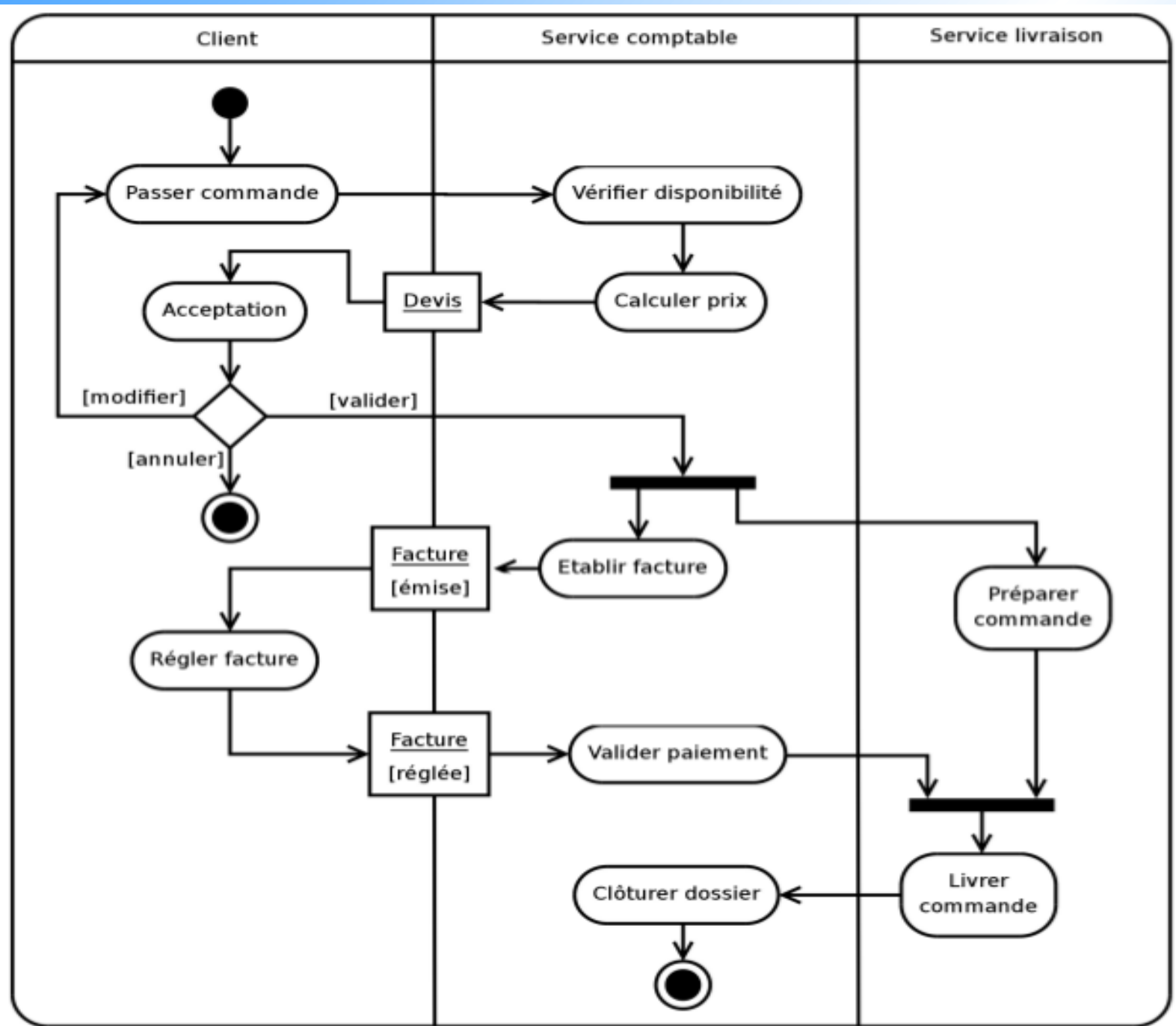
*Utile pour
modéliser des
processus
d'entreprise*

PARTITION

- Afin d'organiser un diagramme d'activité selon les différents responsables des actions représentées, il est possible de définir des "*couloirs d'activité*" ou "*lignes d'eau*".
- Les **partitions**, appelées aussi **couloirs d'activité** ou **lignes d'eau** (*swimlane*), permettent d'**organiser les nœuds d'actions** dans un diagramme d'activité en opérant des regroupements.
- Les partitions n'ont pas de signification bien arrêtée, mais correspondent souvent à des unités d'organisation du modèle.

PARTITION

- Par exemple, on peut les utiliser pour spécifier la classe responsable d'un ensemble tâche. Dans ce cas, la classe est responsable de l'implémentation du comportement des nœuds inclus dans ladite partition.
- Les nœuds d'une activité peuvent être organisés en partitions verticales ou horizontales permettant de séparer les responsabilités.
- C'est particulièrement utile pour modéliser des processus d'entreprise.



Faire le diagramme d'activité du cas d'utilisation *Retirer de l'argent*.

● Scénario nominal

- 1- Le porteur de carte introduit sa carte dans le GAB.
- 2- Le GAB vérifie que la carte introduite est valide.
- 3- Le GAB demande au porteur de carte de saisir son code d'identification.
- 4- Le porteur de carte entre son code d'identification.
- 5- Le GAB compare le code d'identification avec celui qui est codé sur la puce de la carte.
- 6- Le GAB demande une autorisation.
- 7- Le système d'autorisation donne son accord et indique le solde hebdomadaire.

- **Scénario nominal** (*suite*)

8- Le GAB demande au porteur de carte de saisir le montant désiré du retrait.

9- Le porteur de carte saisit le montant désiré du retrait.

10- Le GAB contrôle le montant demandé par rapport au solde hebdomadaire.

11- Le GAB demande au porteur de carte s'il veut un ticket.

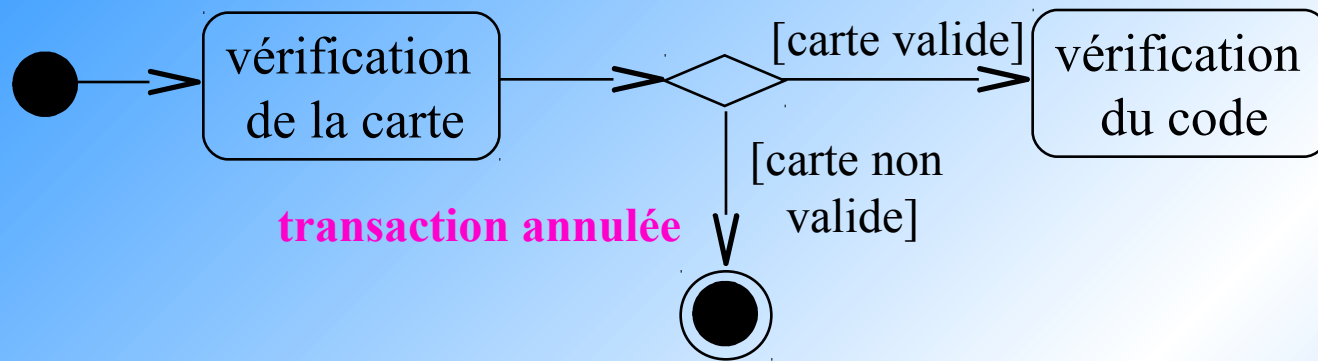
12- Le porteur de carte demande un ticket.

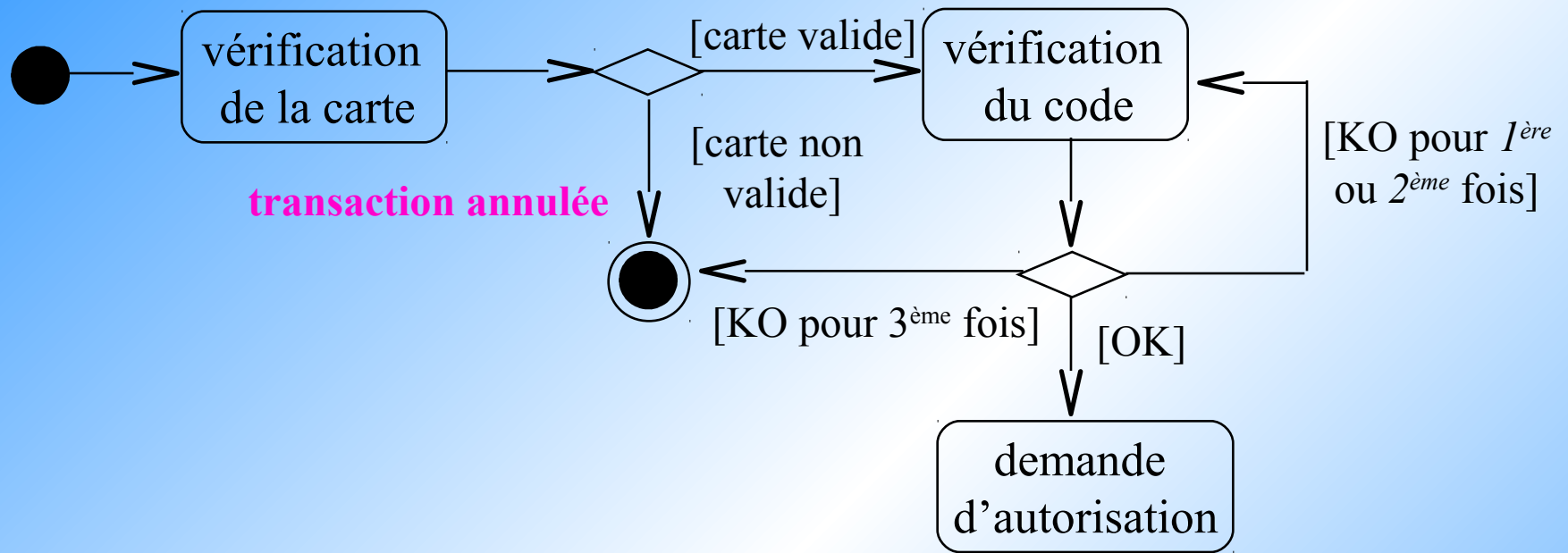
13- Le GAB rend sa carte au porteur de carte.

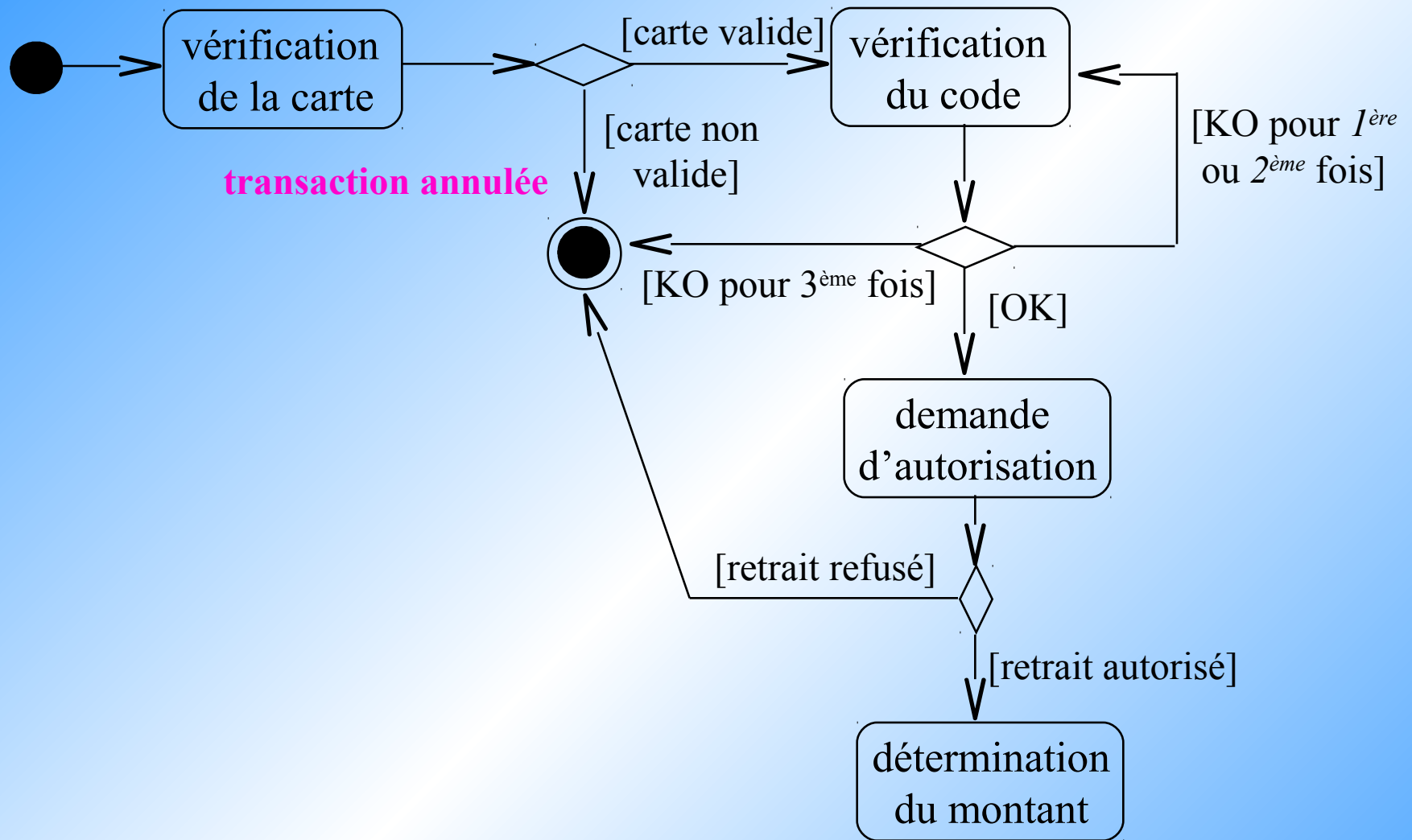
14- Le porteur de carte reprend sa carte.

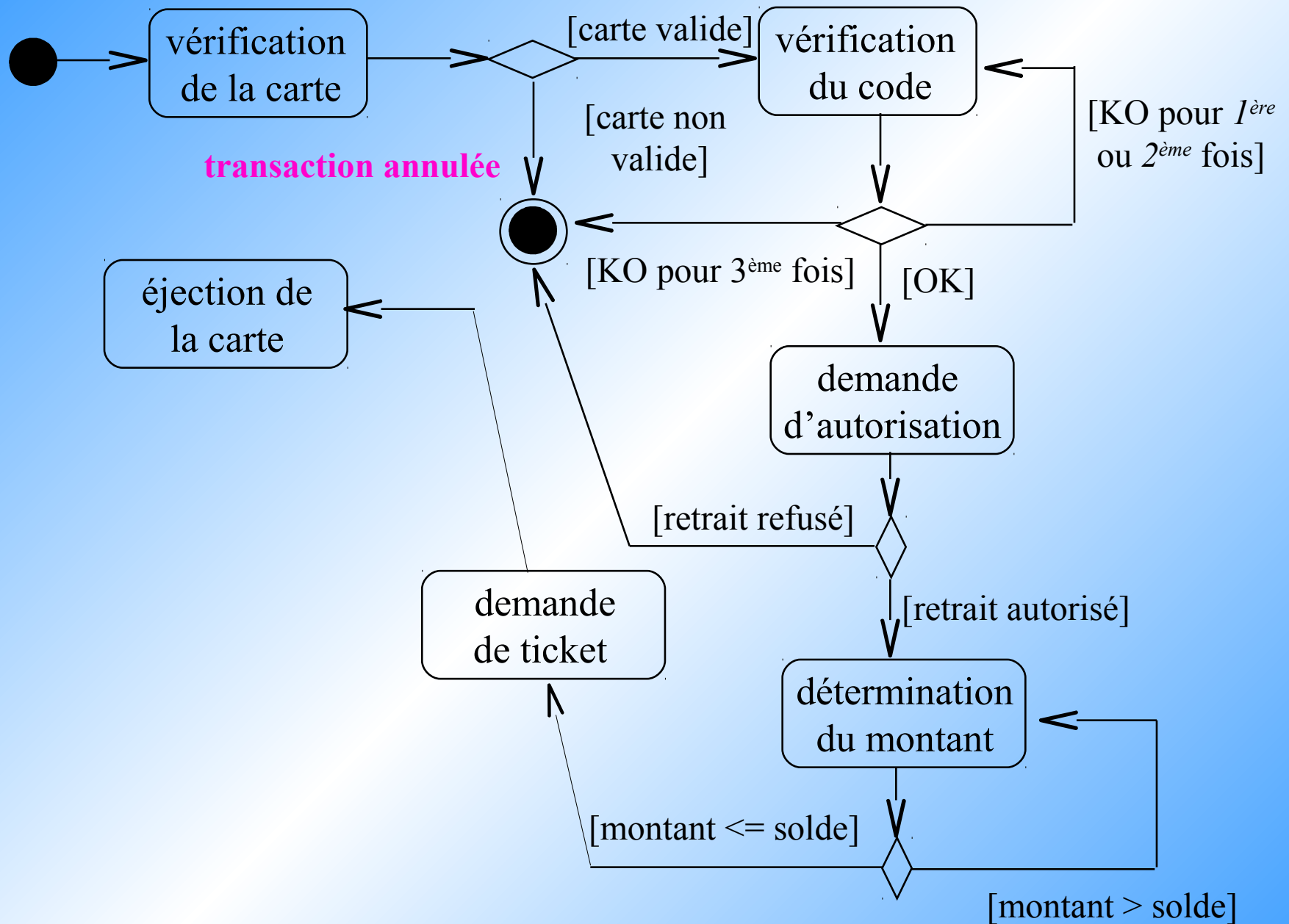
15- Le GAB délivre les billets et un ticket.

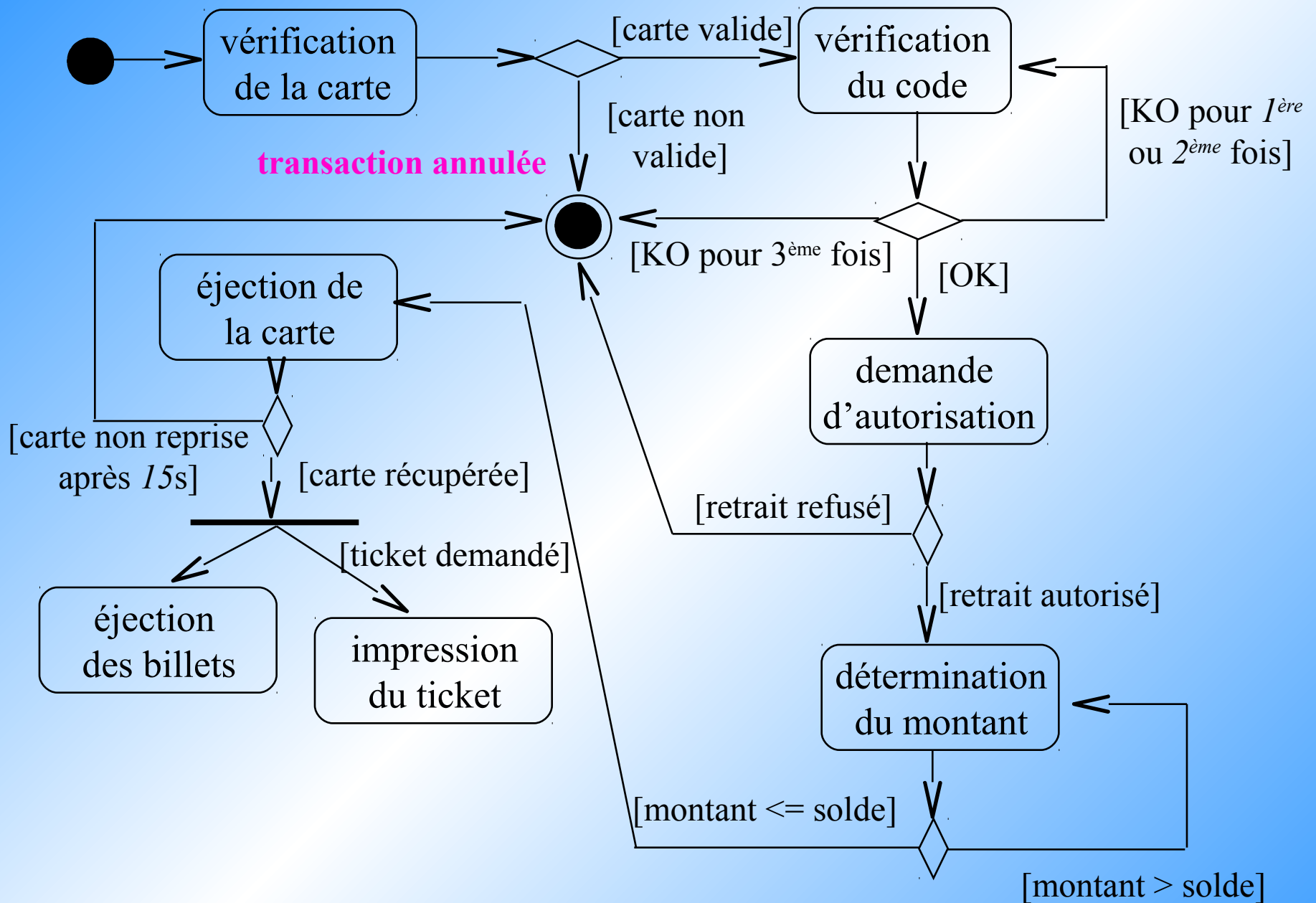
16- Le porteur de carte prend les billets et le ticket.

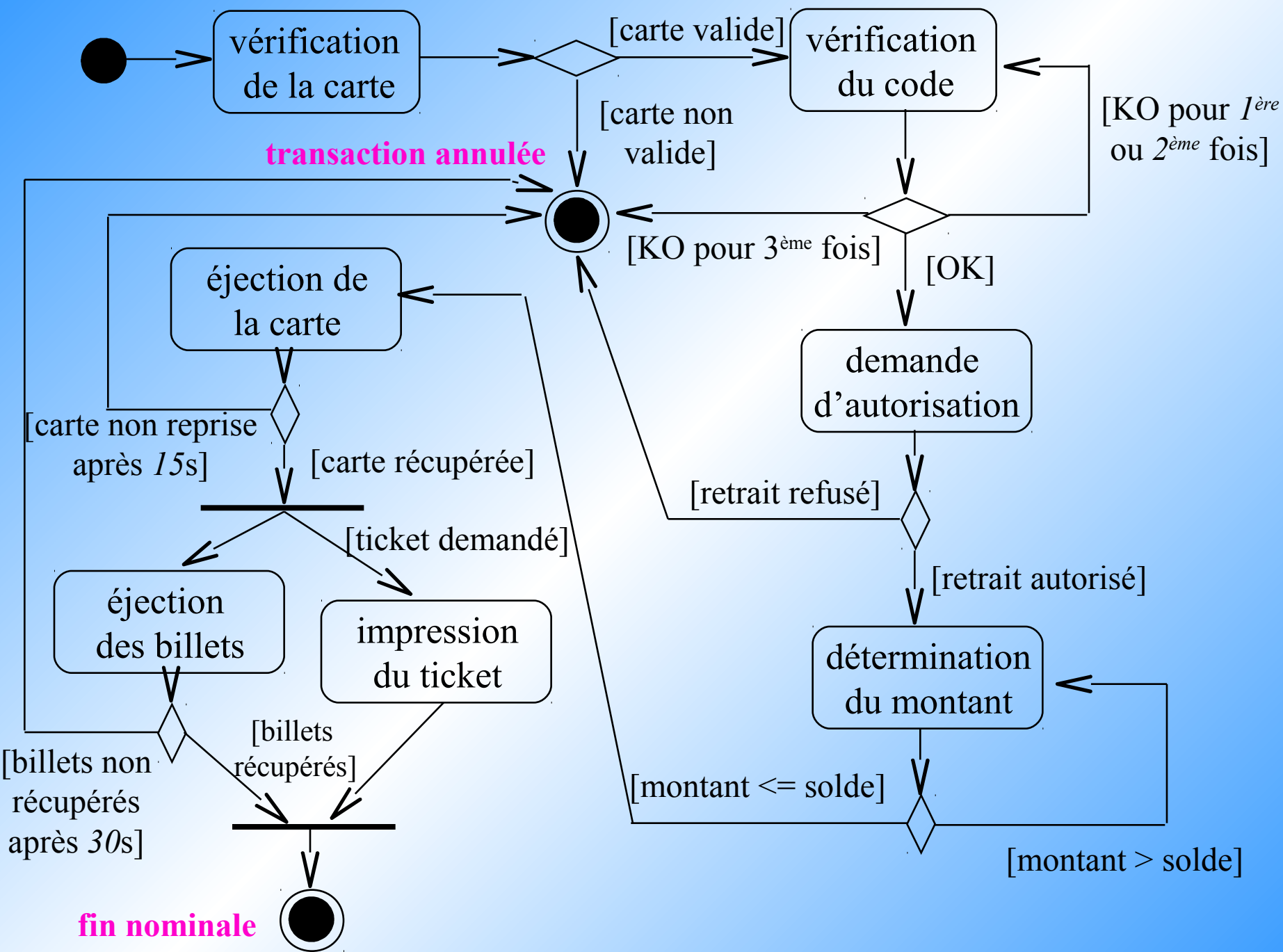












EXERCICE

- Dans le cadre de l'amélioration qu'elle veut apporter à son système d'information, une entreprise souhaite modéliser, dans un premier temps, le processus de formation de ses employés afin que quelques-unes de leurs tâches soient informatisées.

1. Le processus de formation est initialisé lorsque le responsable formation reçoit une demande de formation de la part d'un employé. Cette demande est instruite par le responsable qui la qualifie et transmet son accord ou son désaccord à l'intéressé.

EXERCICE

2. En cas d'accord, le responsable recherche dans le catalogue des formations agréées un stage qui correspond à la demande. Il informe l'employé du contenu de la formation et lui propose une liste des prochaines sessions. Lorsque l'employé a fait son choix, le responsable formation inscrit le participant à la session auprès de l'organisme de formation concerné.
3. En cas d'empêchement, l'employé doit informer le responsable de formation au plus tôt pour annuler l'inscription ou la demande.
4. À la fin de sa formation, l'employé doit remettre au responsable formation une appréciation sur le stage qu'il a effectué, ainsi qu'un document justifiant de sa présence.

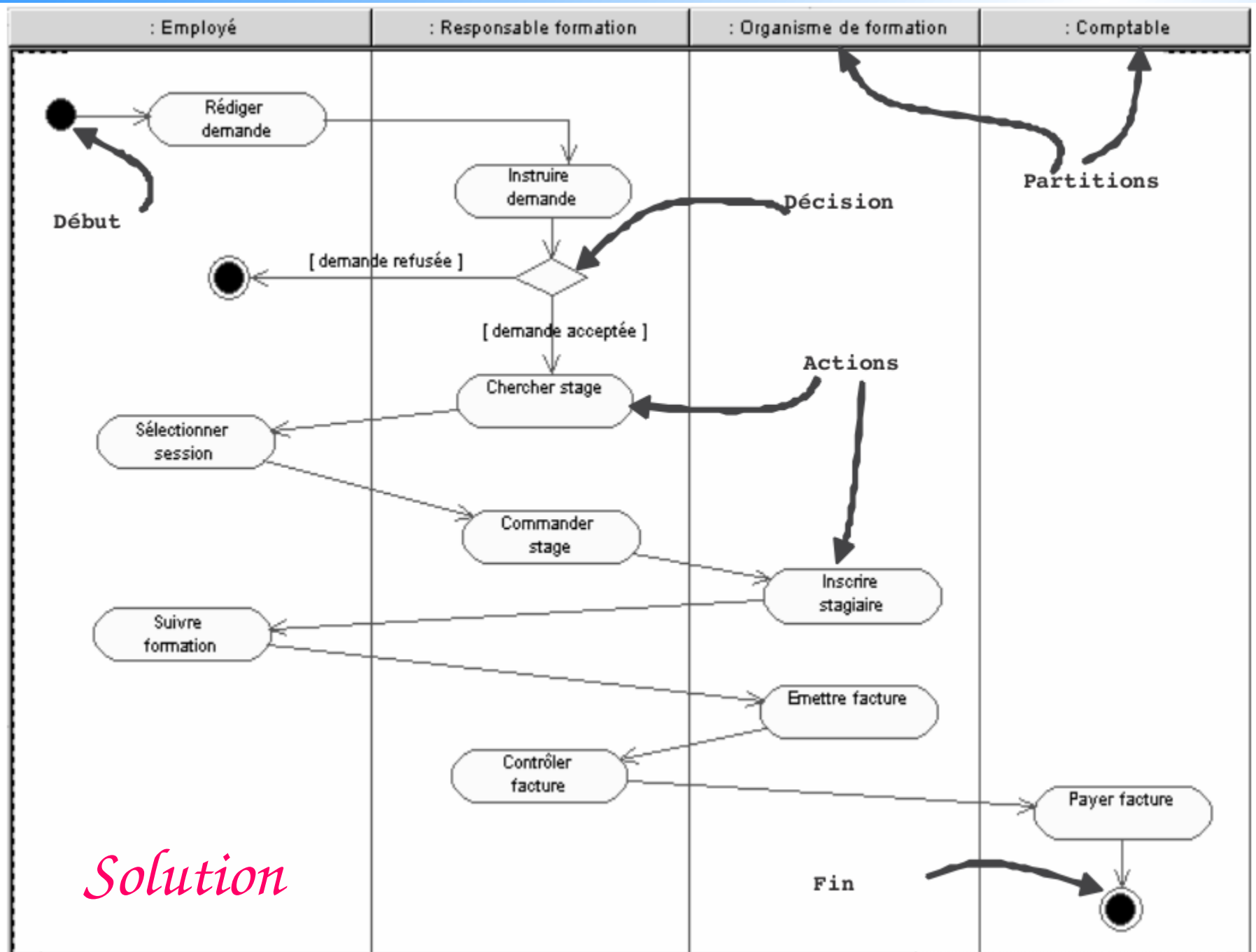
EXERCICE

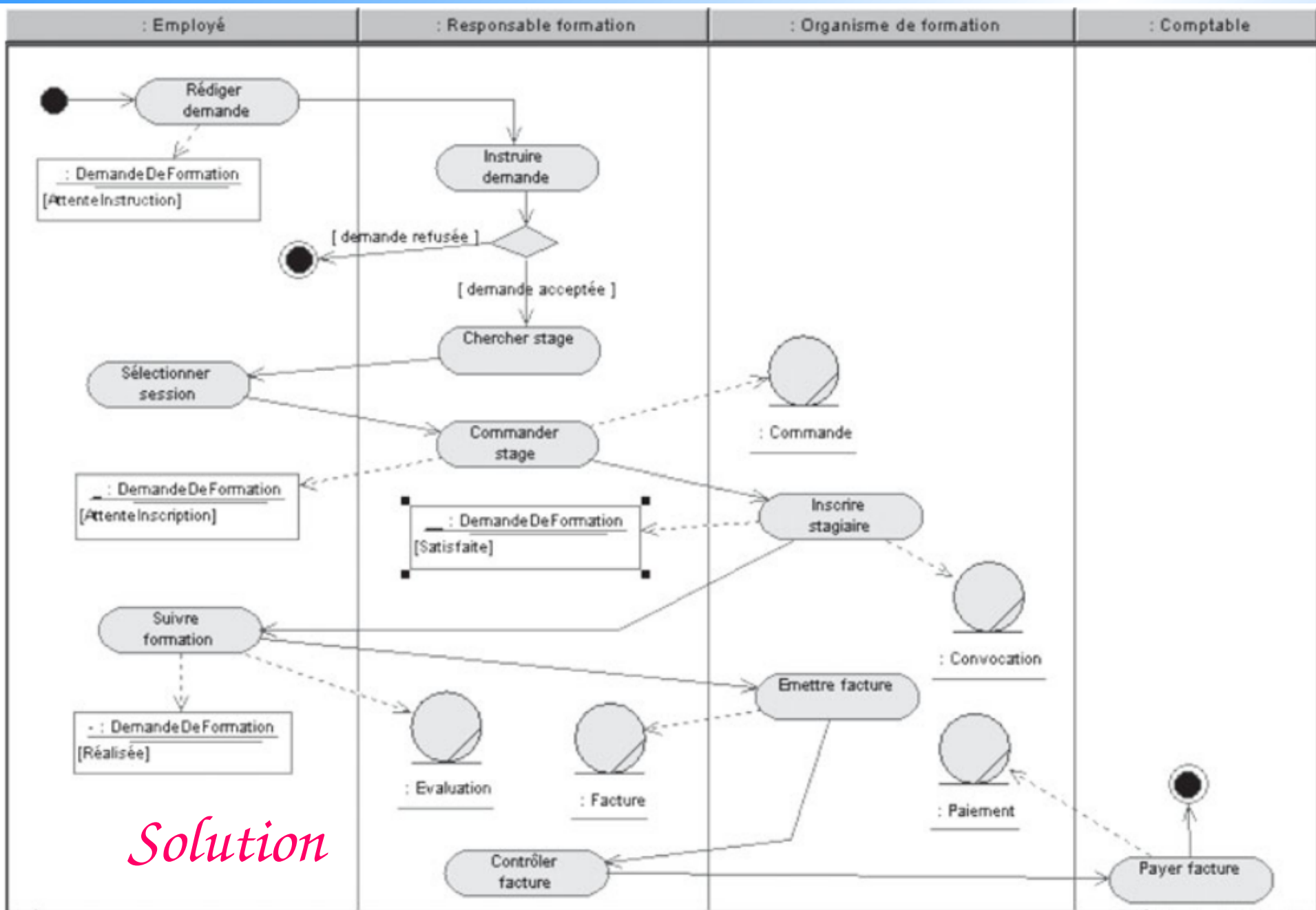
5. Le responsable formation contrôle par la suite la facture que l'organisme de formation lui a envoyée avant de la transmettre au comptable achats.

Décrivez la dynamique du processus de formation au moyen d'un diagramme d'activité.

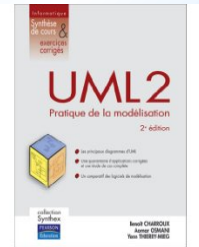
Utilisez des partitions verticales pour affecter les responsabilités aux acteurs.

*Modélisez le processus de formation
avec un diagramme d'activité*





EXERCICE



Les chaînes de caractères du langage C sont codées comme un tableau de caractères non nuls, terminé par un caractère `'\0'`.

Par exemple, la chaîne `s="hello!"` est codée comme suit :

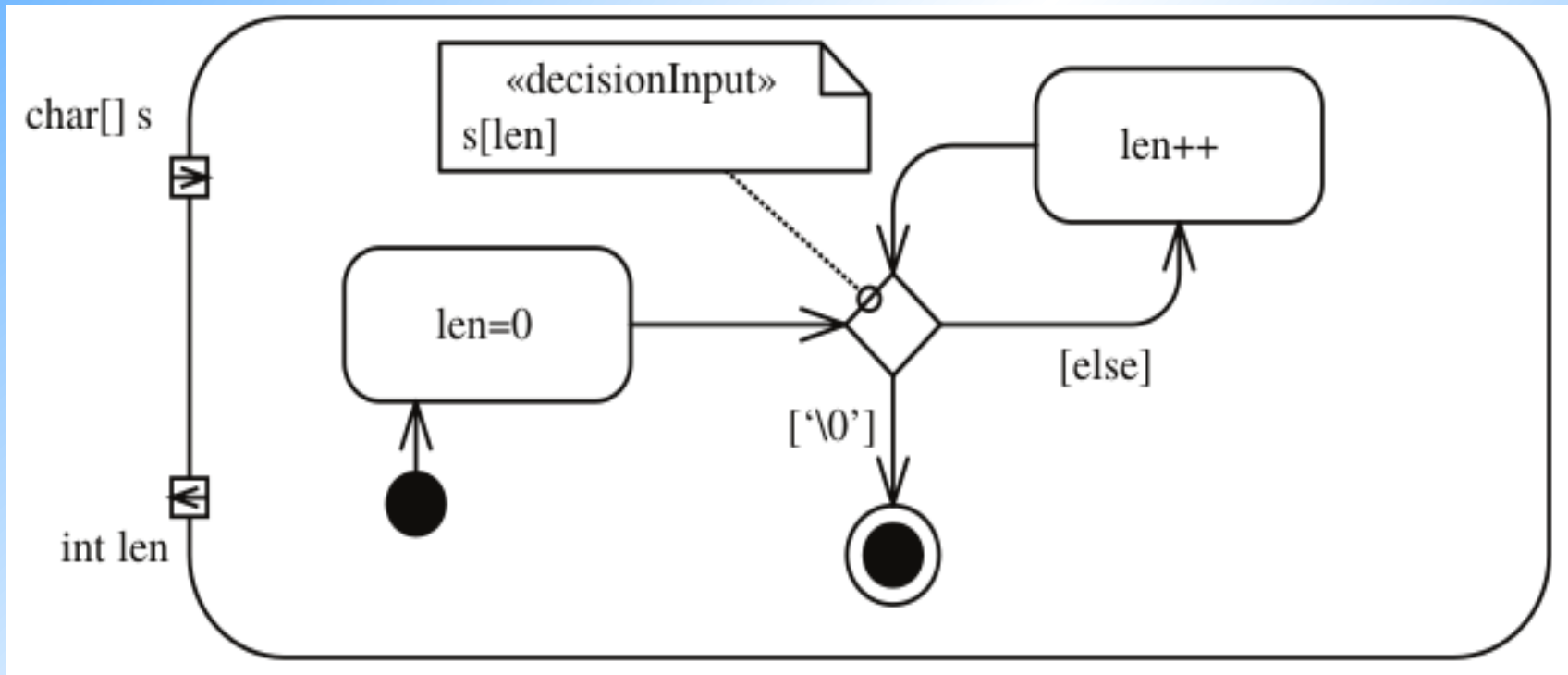
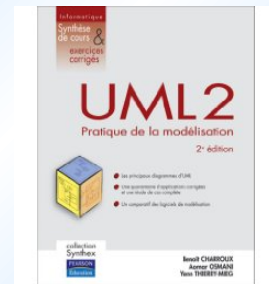
s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	s[6]
'h'	'e'	'l'	'l'	'o'	'!'	'\0'

Décrivez une activité implémentant la fonction *strlen*, qui prend en entrée un tableau de caractères et rend un entier correspondant à la taille de la chaîne.

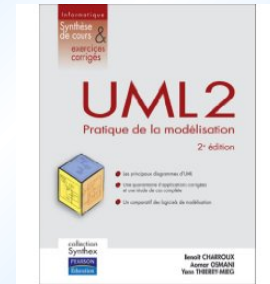
Exemple : `strlen("hello!") = 6` .

Faire le diagramme d'activité

SOLUTION



EXERCICE

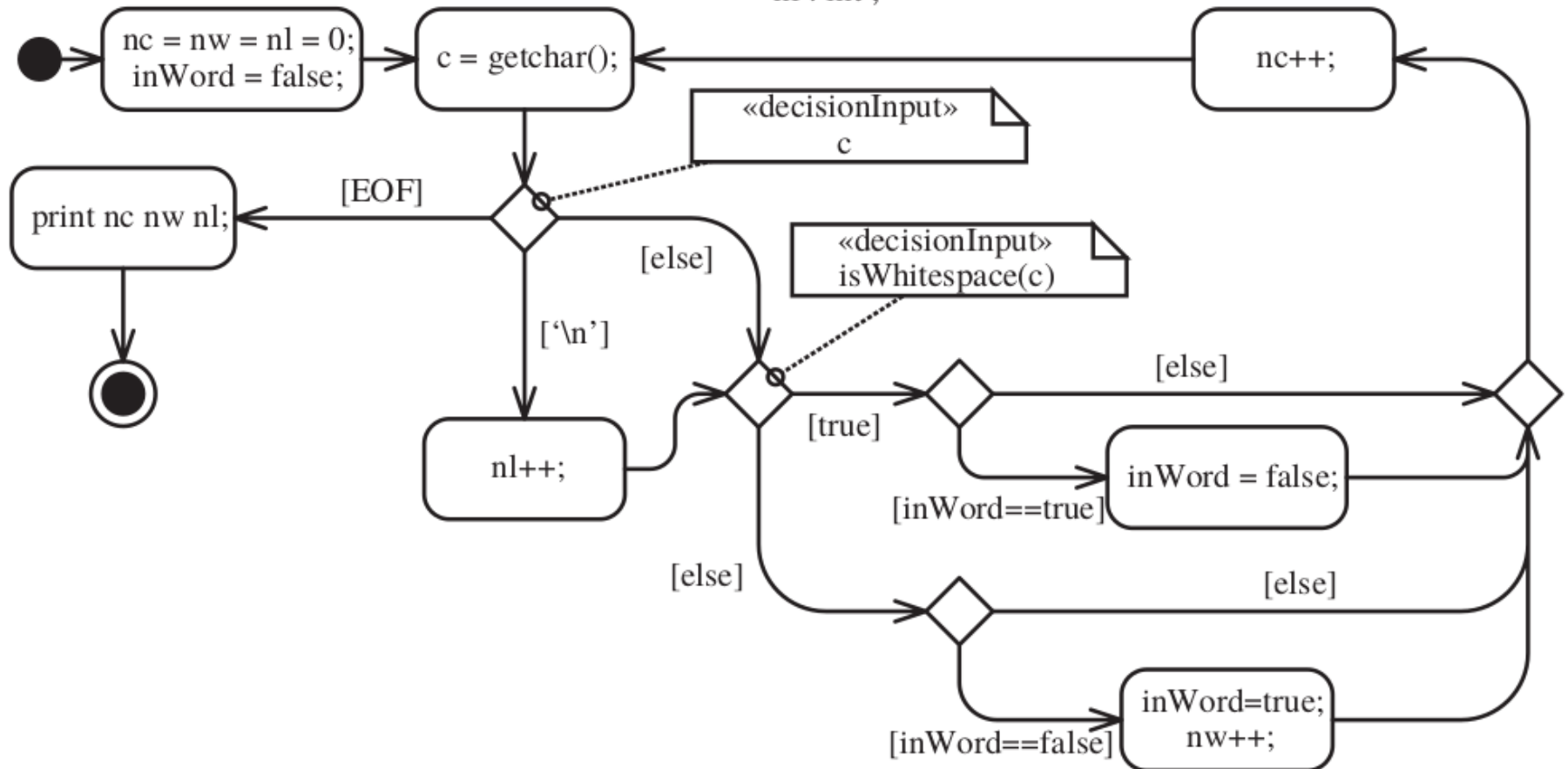


Faire le diagramme d'activité qui compte les mots, les lignes et les caractères de son entrée.

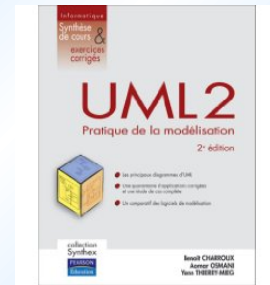
SOLUTION

«activity» wc

nc : int ; c : char ; inWord : bool
 nw : int ;
 nl : int ;



SOLUTION



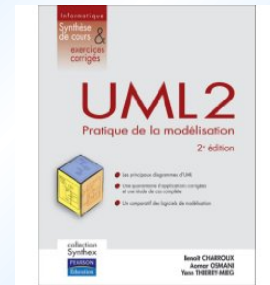
L'activité appelée *wc* réalise le comportement de l'outil Unix *wc* (*word count*). On utilise ici la notation *frame* (*qui sert aussi dans les diagrammes de séquence*) pour la représenter, et le stéréotype *activity*.

On déclare les variables locales de l'activité, *nc*, *nw*, *nl*, qui servent à compter respectivement le nombre de caractères, de mots et de lignes de l'entrée.

On utilise une variable *c* de type *char* pour stocker le dernier caractère lu de l'entrée (*par la fonction `getchar()`*).

L'état courant de la lecture (*dans mot ou hors mot*) est représenté par une variable booléenne *inWord*.

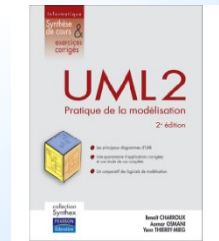
SOLUTION



On suppose ici que *print*, *getchar*, *isWhitespace*, l'opération d'incrément ++ sont fournies (*sous la forme d'actions opaques*).

Comme on le voit sur cet exemple, les diagrammes d'activité d'un niveau de spécification détaillée permettent d'atteindre le pouvoir d'expression de langages de programmation classiques.

EXERCICE

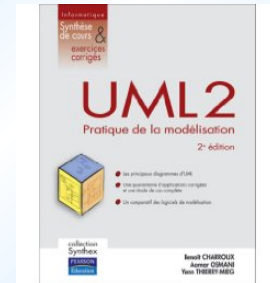


Faire le diagramme de l'activité "gérer une commande"

L'activité débute à la réception d'une commande d'un client. Un dossier est créé et ensuite on va pouvoir simultanément préparer la commande et traiter le paiement du client. Une fois ces deux dernières actions achevées, on clôture le dossier.

Le client est libre d'annuler sa commande à tout moment. S'il passe l'ordre d'annuler, le dossier est effacé. Cependant, une fois que le paiement est effectif et que la commande est prête, il devient impossible de l'annuler : la transaction est dûment enregistrée et le dossier clôturé.

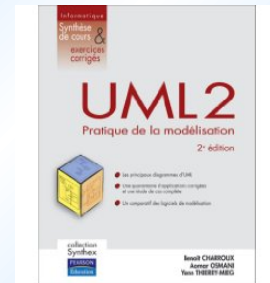
EXERCICE



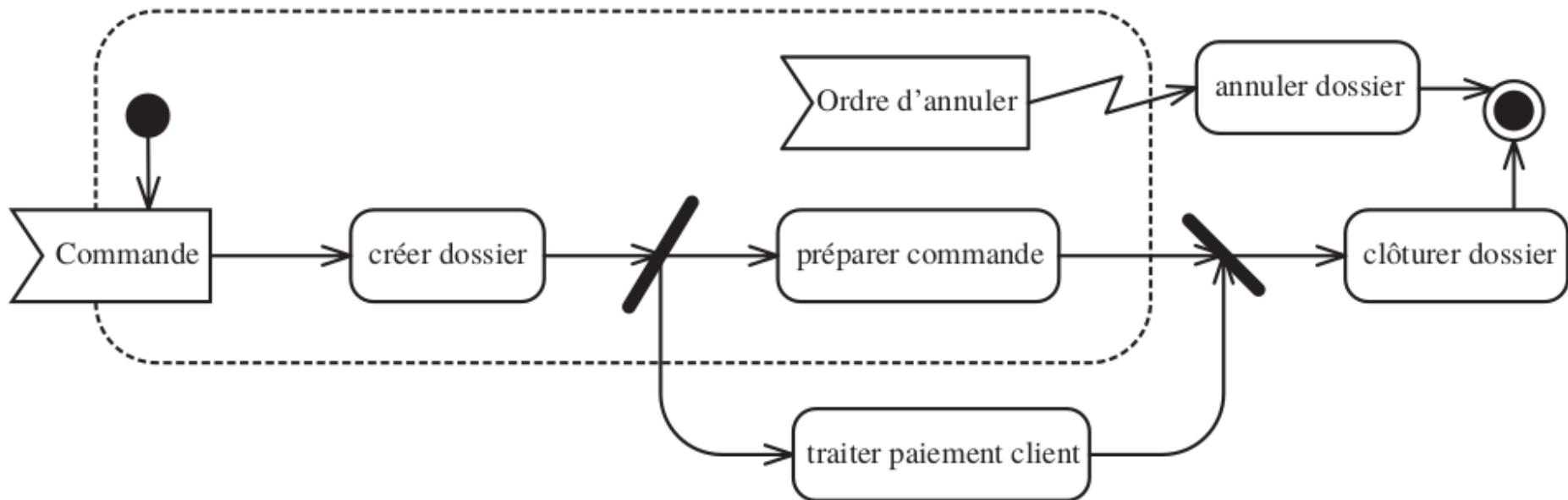
Faire le diagramme de l'activité "gérer une commande"

Après réception d'un ordre d'annulation, et durant l'action "*annuler dossier*," l'action "*traiter paiement*" peut continuer à s'exécuter. Cependant, elle est, elle aussi, interrompue quand le flot de traitement a fini d'annuler le dossier et que l'état final est atteint (*cela termine tous les flots de l'activité*).

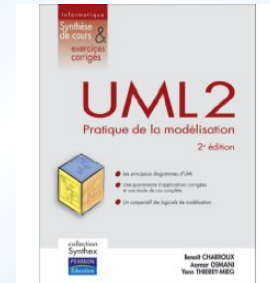
SOLUTION



«activity» gérer commande



EXERCICE



Description du cas « Emprunter une vidéo »

Identification

Nom du cas : « Emprunter une vidéo ».

But : décrire les étapes permettant au client du magasin d'emprunter une cassette vidéo via le distributeur automatique.

Acteur principal : Client.

Acteur secondaire : néant.

Date de création : le 31/12/2004.

Date de mise à jour : le 1/1/2005.

Responsable : M. Dupont.

Version : 1.1.

EXERCICE



Séquencement

Le cas d'utilisation commence lorsqu'un client introduit sa carte.

Pré-conditions

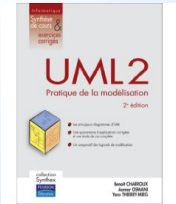
Le client possède une carte qu'il a achetée au magasin.

Le distributeur est alimenté en cassettes.

Enchaînement nominal

1. Le système vérifie la validité de la carte.
2. Le système vérifie que le crédit de la carte est supérieur ou égal à 1 euro.
3. Appel du cas « Rechercher une vidéo ».
4. Le client a choisi une vidéo.
5. Le système indique, d'après la valeur de la carte, pendant combien de temps (tranches de 6 heures) le client peut garder la cassette.
6. Le système délivre la cassette.
7. Le client prend la cassette.
8. Le système rend la carte au client.
9. Le client prend sa carte.

EXERCICE



Enchaînements alternatifs

A1 : Le crédit de la carte est inférieur à 1 euro.

L'enchaînement démarre après le point 2 de la séquence nominale :

3. Le système indique que le crédit de la carte ne permet pas au client d'emprunter une vidéo.
4. Le système invite le client à aller recharger sa carte au magasin.

La séquence nominale reprend au point 8.

Enchaînements d'exception

E1 : La carte introduite n'est pas valide.

L'enchaînement démarre après le point 1 de la séquence nominale :

2. Le système indique que la carte n'est pas reconnue.
3. Le distributeur éjecte la carte.

E2 : La cassette n'est pas prise par le client.

L'enchaînement démarre après le point 6 de la séquence nominale :

7. Au bout de 15 secondes le distributeur avale la cassette.
8. Le système annule la transaction (toutes les opérations mémorisées par le système sont défaites).
9. Le distributeur éjecte la carte.

E3 : La carte n'est pas reprise par le client.

L'enchaînement démarre après le point 8 de la séquence nominale :

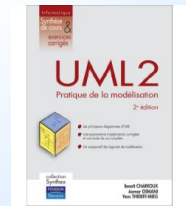
9. Au bout de 15 secondes le distributeur avale la carte.
10. Le système consigne cette erreur (date et heure de la transaction, identifiant du client, identifiant du film).

E4 : Le client a annulé la recherche (il n'a pas choisi de vidéo).

L'enchaînement démarre au point 4 de la séquence nominale :

5. Le distributeur éjecte la carte.

EXERCICE



Post-conditions

Le système a enregistré les informations suivantes :

- La date et l'heure de la transaction, à la minute près : les tranches de 6 heures sont calculées à la minute près.
- L'identifiant du client.
- L'identifiant du film emprunté.

Rubriques optionnelles

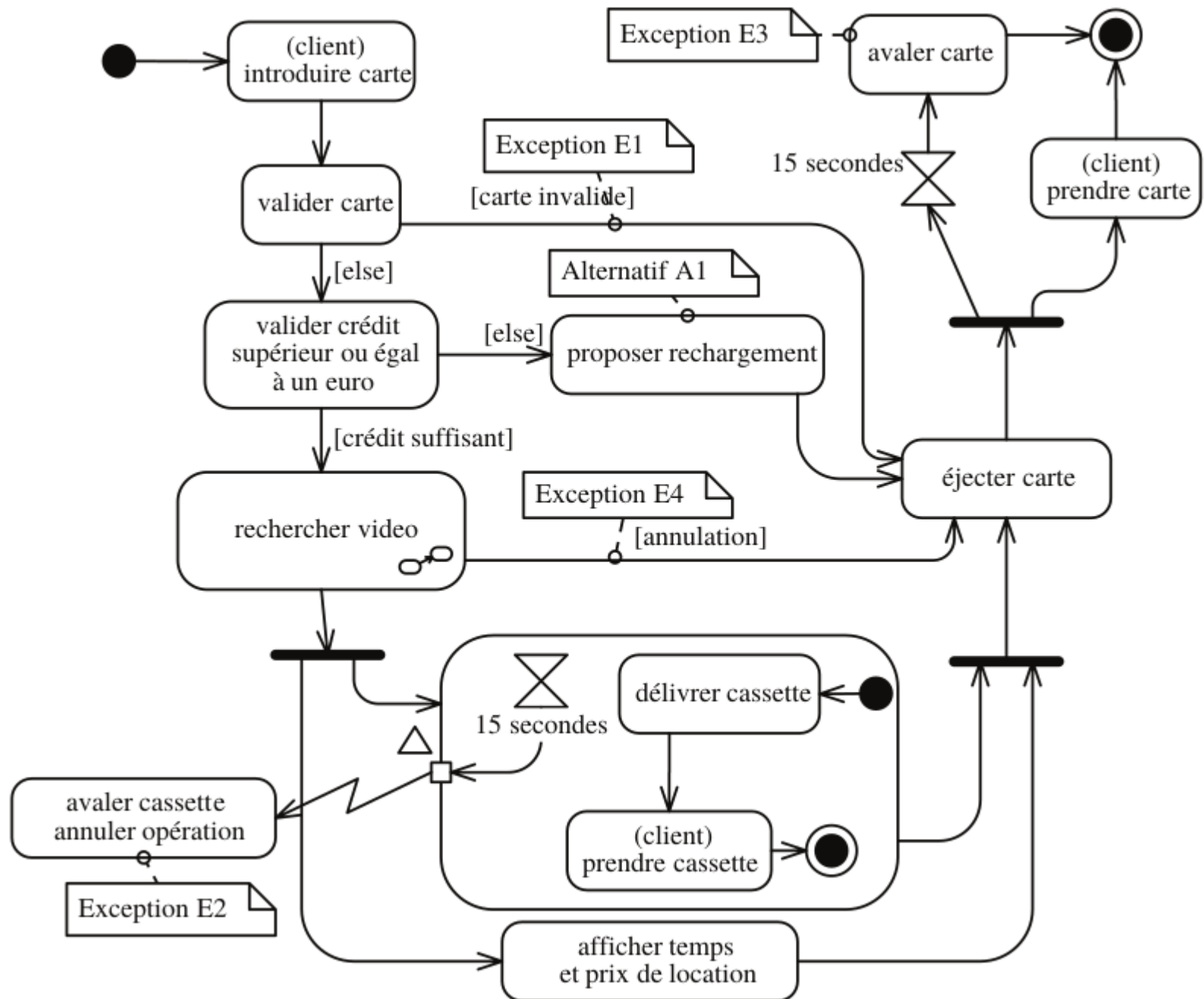
Contraintes non fonctionnelles

Le distributeur doit fonctionner 24 heures sur 24 et 7 jours sur 7.

La vérification de la validité de la carte doit permettre la détection des contrefaçons.

Contrainte liée à l'interface homme-machine

Avant de délivrer la cassette, demander confirmation au client.



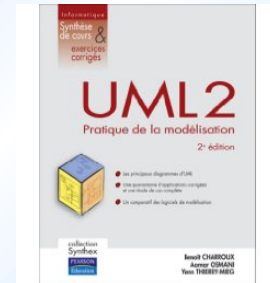
SOLUTION

Deux mécanismes différents sont utilisés pour représenter le délai de 15 secondes pendant lequel le client récupère sa cassette ou sa carte.

Dans le premier cas, une exception gère cet événement : cela permet, en cas de levée de l'exception, de reprendre le traitement au même point que si le client avait récupéré sa cassette et d'enchaîner sur éjecter carte.

Dans le deuxième cas, deux actions sont mises en concurrence : prendre carte et le *time event* de 15 secondes. L'astuce ici est que le premier flot de contrôle qui atteint le nœud final force la terminaison de tous les autres flots. Si ce comportement n'est pas celui souhaité, il faut utiliser un nœud *flow final* (cercle avec une croix) pour terminer un flot sans affecter les autres flots de la région.

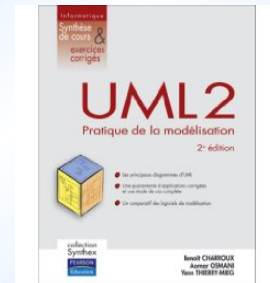
EXERCICE



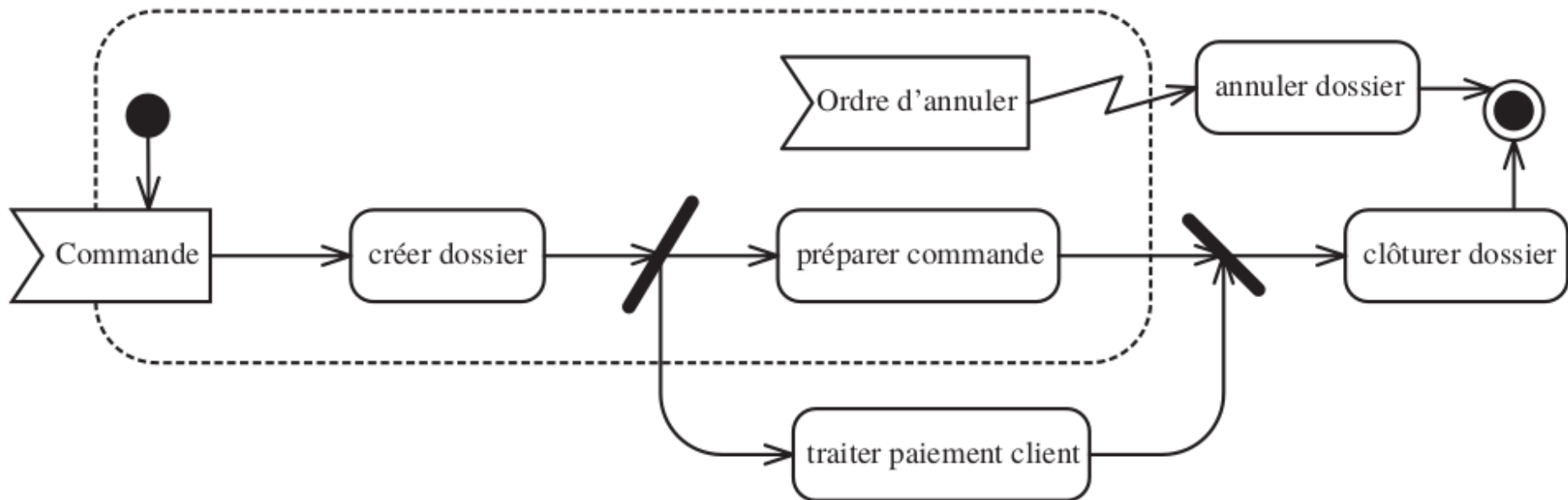
Faire le diagramme de l'activité "gérer une commande"

Après réception d'un ordre d'annulation, et durant l'action "*annuler dossier*," l'action "*traiter paiement*" peut continuer à s'exécuter. Cependant, elle est, elle aussi, interrompue quand le flot de traitement a fini d'annuler le dossier et que l'état final est atteint (*cela termine tous les flots de l'activité*).

SOLUTION



«activity» gérer commande



SOMMAIRE

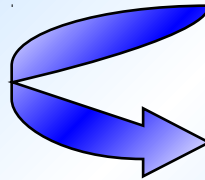
- Introduction
- Diagrammes d'activités
- **Navigation dans le site Web**



INTRODUCTION

Déjà réalisé :

- Identification des cas d'utilisation
- Description détaillée des cas d'utilisation
- Première version du modèle du domaine



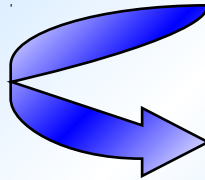
Pour passer en douceur à la conception

Identification

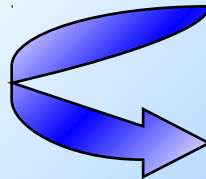
- des principales **classes d'IHM** (*dialogues*)
- des classes décrivant la **cinématique de l'application** (*contrôles*)

INTRODUCTION

- Reste à détailler une **exploitation supplémentaire de la maquette** ou une **extension** éventuelle de celle-ci.



Réalisation de diagrammes dynamiques représentant l'**ensemble des chemins possibles** entre les principaux **écrans** proposés à l'utilisateur.



Diagrammes d'activités de navigation

DIAGRAMME D'ACTIVITÉS

- Pour modéliser la navigation dans un site Web,
 - ✓ on va se servir d'un certain nombre d'éléments standards
 - ✓ mais aussi de conventions particulières.

CONVENTIONS SPÉCIFIQUES

- Utilisation du concept d'activité pour modéliser plusieurs concepts différents :

Pas norme UML

- ✓ une page du site (*<<page>>*)
- ✓ un *frame* à l'intérieur d'une page (*<<frame>>*)
- ✓ une action simple (*par exemple classer des ouvrages*)
(*<<action>>*)
- ✓ une erreur ou un comportement inattendu du système
(*<<exception>>*)
- ✓ une liaison vers un autre diagramme d'activités
(*pour des raisons de structuration et de lisibilité*)
(*<<connector>>*)

CONVENTIONS SPÉCIFIQUES

Pas norme UML

<<page>>

<<frame>>

<<action>>

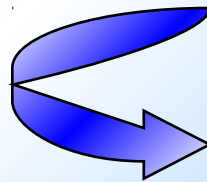
<<exception>>

<<connector>>

- Les concepts de *page* et de *frame* sont directement en rapport avec les classes *dialogues*.
- Le concept d'*action* correspond généralement à une *opération* d'une classe *dialogue* ou d'une classe *contrôle*.

NAVIGATION DANS LE SITE WEB

- **Séparation des tâches** en ensembles maîtrisables et les plus indépendants possibles.
- La navigation du site par l'internaute $\neq$ la navigation du libraire
pas accès aux mêmes fonctionnalités

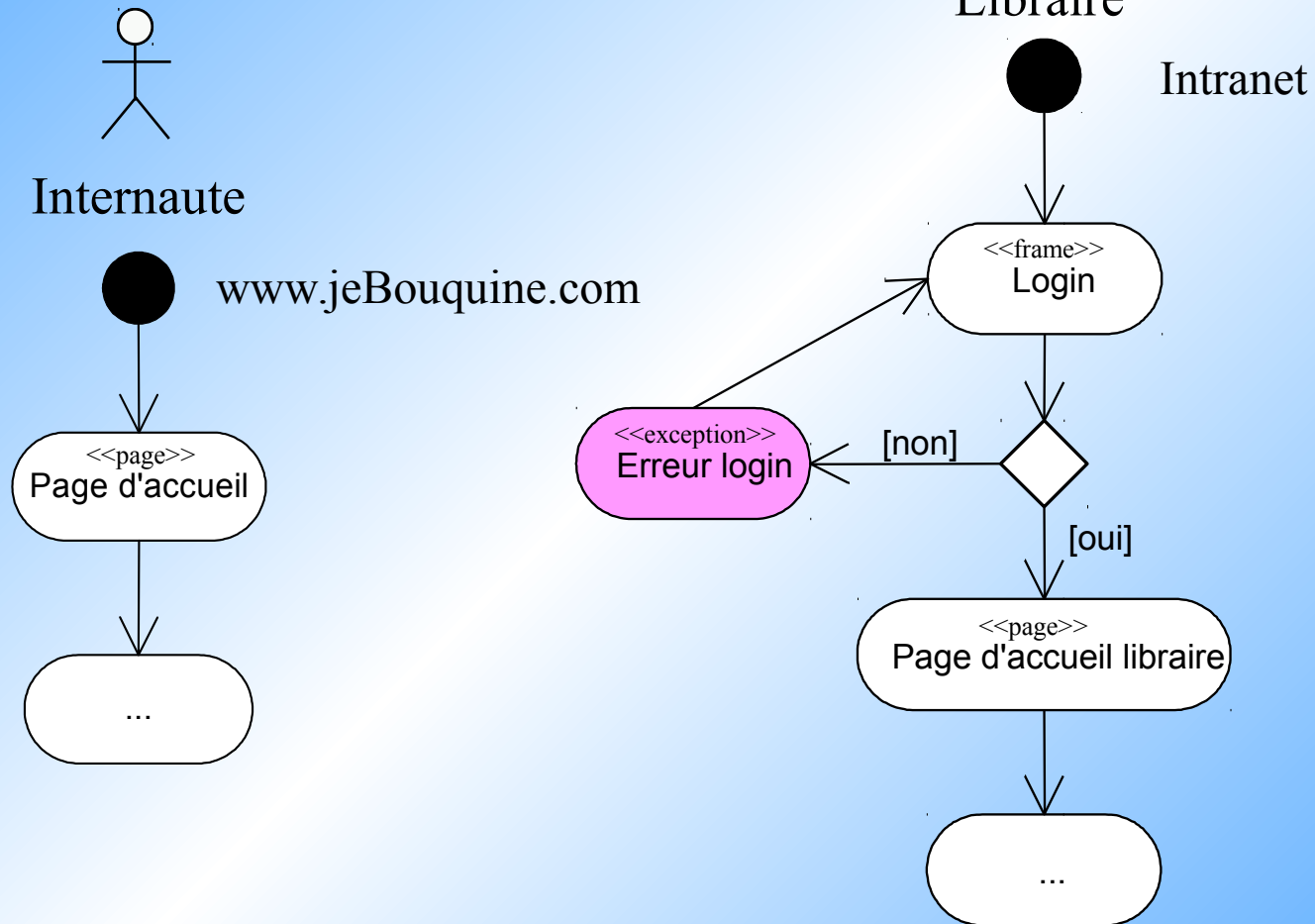


Navigation peut se structurer tout d'abord par acteur

NAVIGATION DANS LE SITE WEB

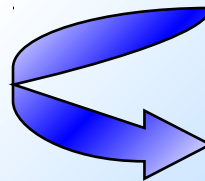
Faire le début du diagramme de navigation
de chaque acteur
(*internaute et libraire*)

SOLUTION



NAVIGATION DANS LE SITE WEB

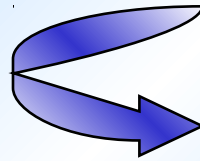
- Navigation **se structure-t-elle** également par **cas d'utilisation** ?
- Si les cas d'utilisation sont complètement indépendants :
 - ✓ réaliser un diagramme de navigation par cas



obtention de diagrammes simples et lisibles

NAVIGATION DANS LE SITE WEB

- Par contre, les cas d'utilisation de l'internaute ont des relations les uns avec les autres :
impossible de commander si le panier est vide, etc.

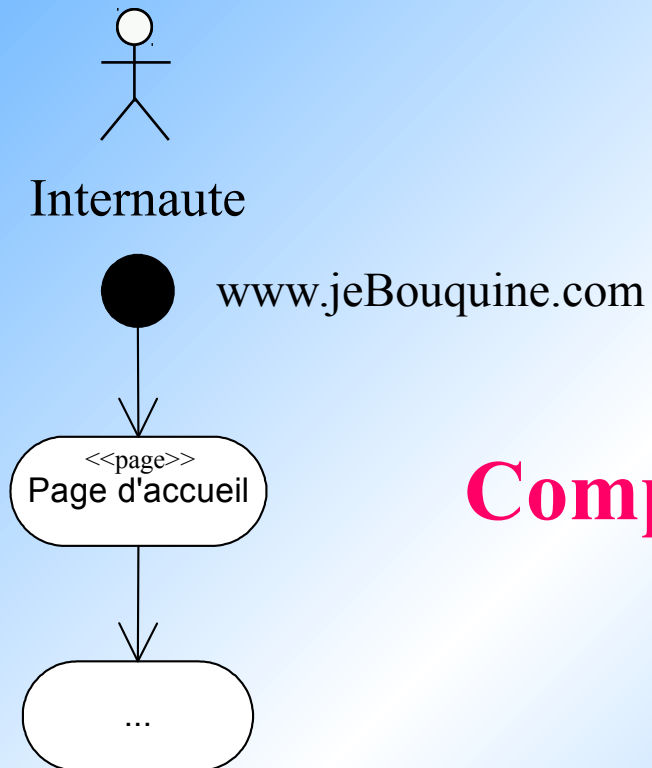


Ces dépendances importantes vont induire très directement des contraintes de navigation entre les pages correspondantes. Et le diagramme d'activités de navigation va nous permettre de les représenter de façon formelle.

RECHERCHER DES OUVRAGES

- L'internaute commence en général par sélectionner des ouvrages pour pouvoir ensuite les commander, après avoir éventuellement géré son panier.
- On part donc de la description détaillée du cas d'utilisation *"Rechercher des ouvrages"*.
- A partir de la page d'accueil, l'internaute doit avoir la possibilité de lancer une recherche rapide, une recherche avancée ou de flâner dans un ensemble de pages telles que : nouveautés, meilleures ventes, etc.

SOLUTION

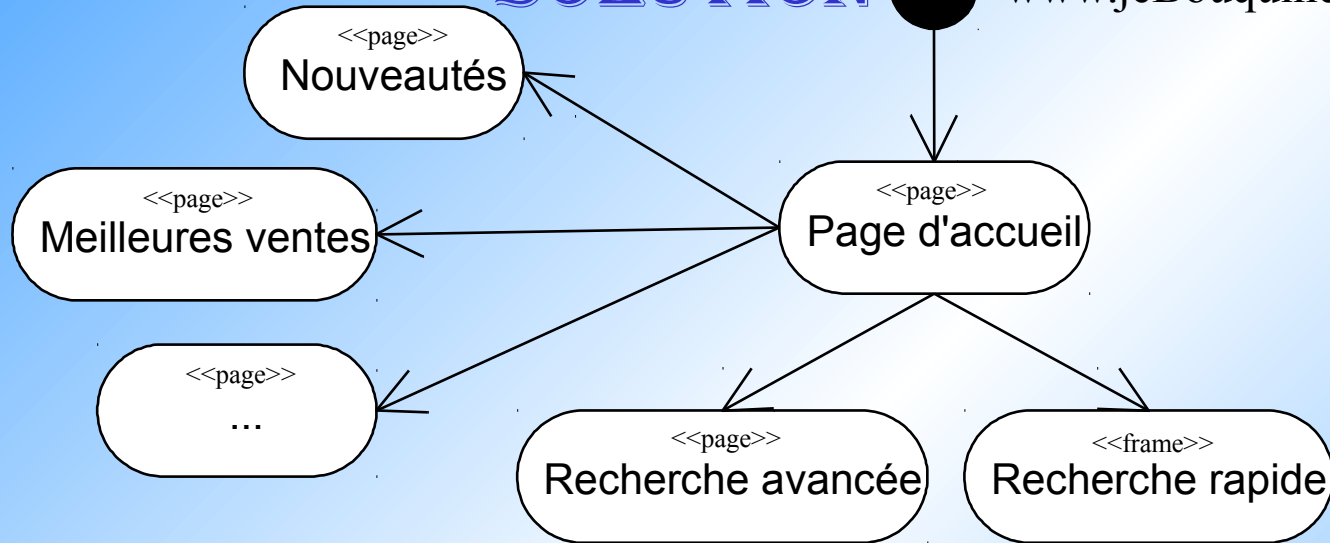


Compléter ce diagramme

SOLUTION



www.jeBouquine.com



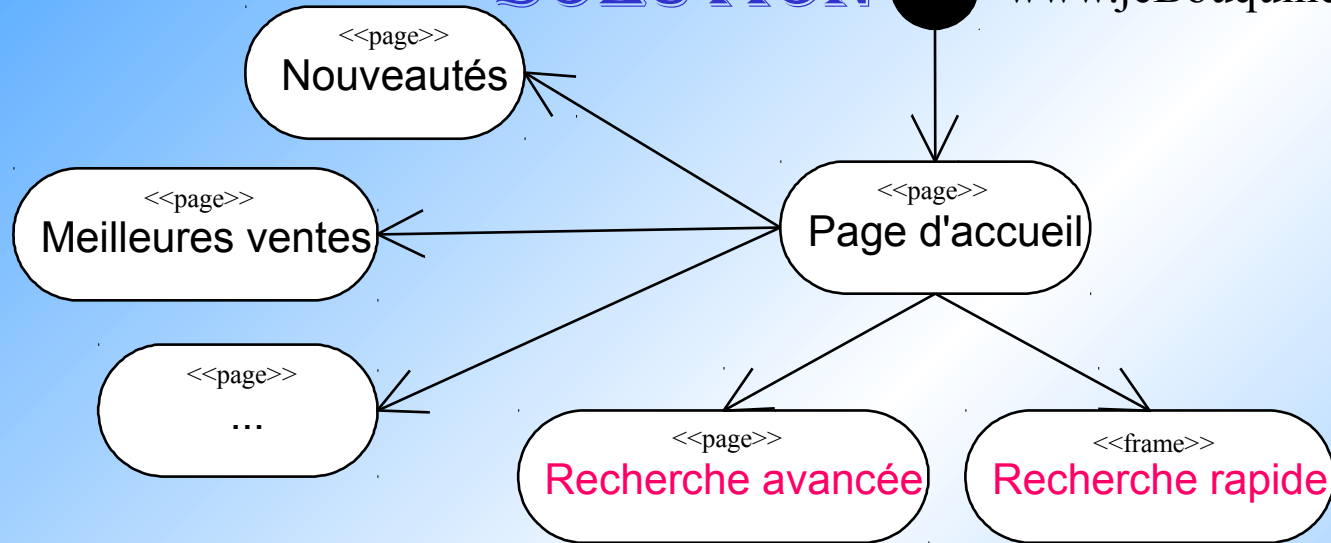
- La recherche rapide est un *frame* toujours présent dans les pages de recherche, contrairement à la recherche avancée qui est une page à part entière.

Se focaliser sur la recherche rapide et avancée

SOLUTION



www.jeBouquine.com

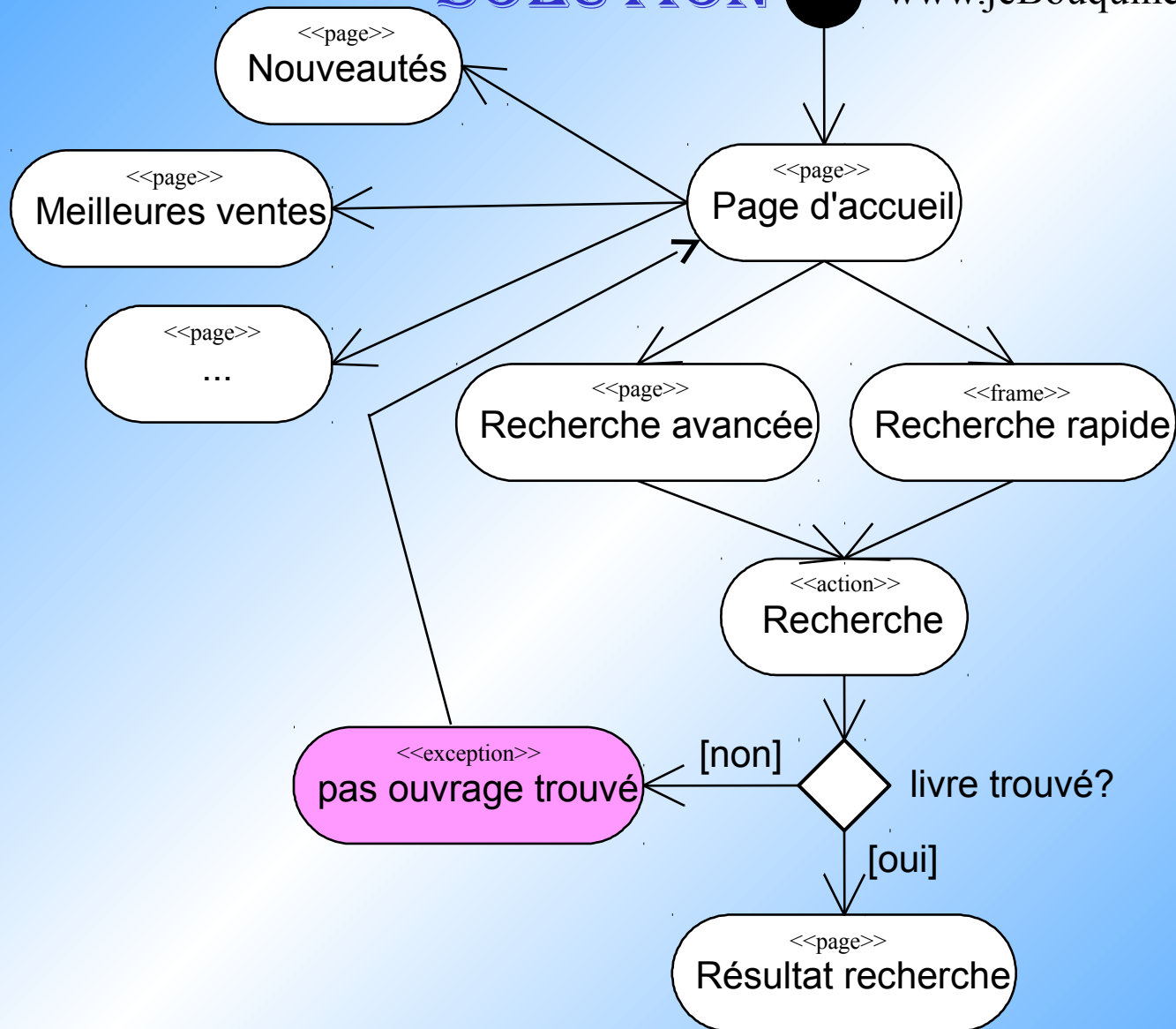


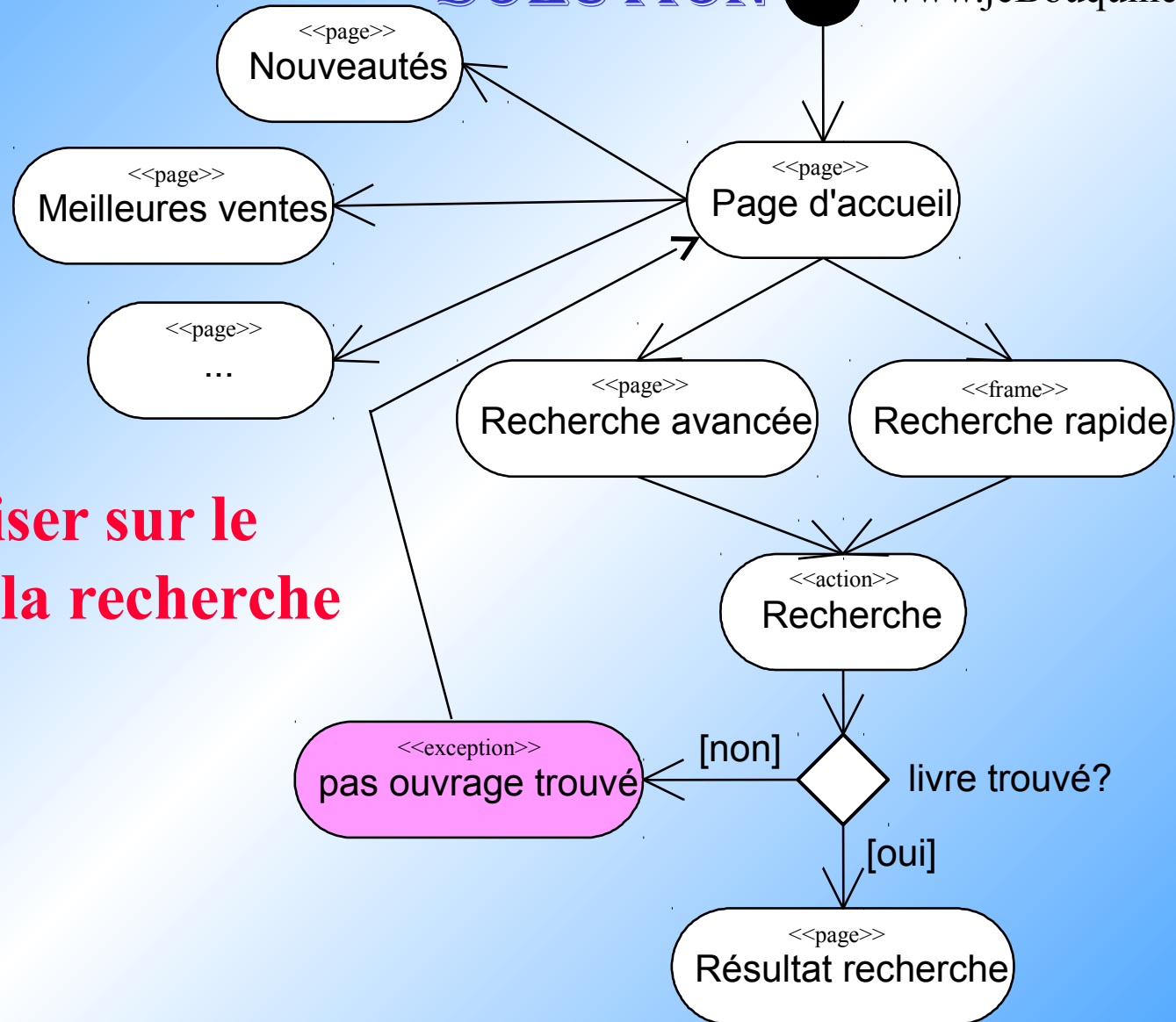
- La recherche elle-même est une action dont le résultat est une page contenant la liste des livres ou bien un message d'erreur indiquant qu'aucun ouvrage n'a été trouvé.

SOLUTION



www.jeBouquine.com





Se focaliser sur le résultat de la recherche

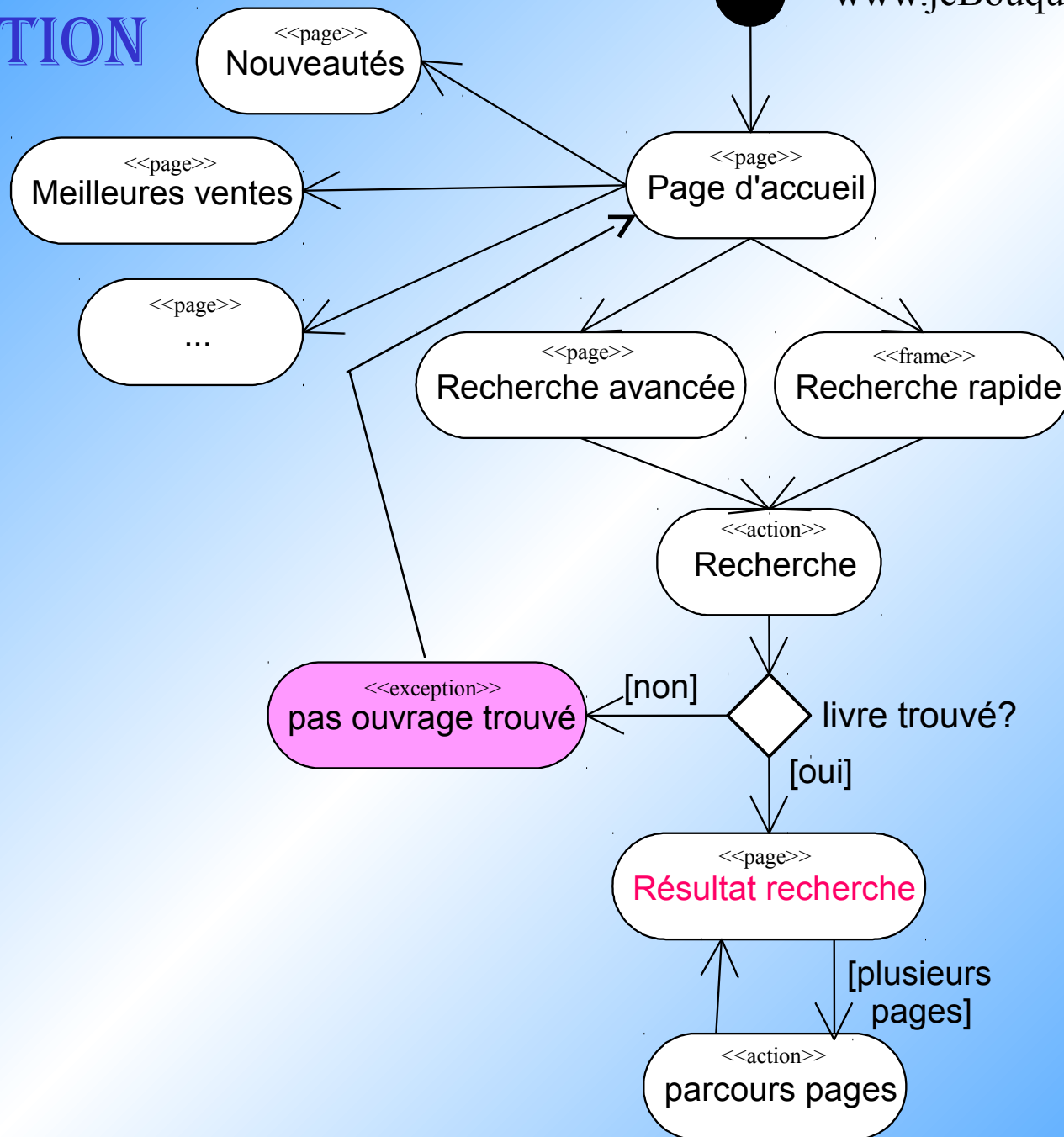
RECHERCHER DES OUVRAGES

- Sur le résultat de la recherche, l'internaute pourra :
 - ✓ soit sélectionner un ouvrage pour obtenir sa fiche détaillée,
 - ✓ soit le mettre directement de côté dans son panier,
 - ✓ soit reclasser les résultats.
- Si le nombre d'ouvrages trouvés est grand, l'internaute pourra naviguer entre les différentes pages de résultats.

SOLUTION

- Sur le résultat de la recherche, l'internaute pourra :
 - ✓ soit sélectionner un ouvrage pour obtenir sa fiche détaillée,
 - ✓ soit le mettre directement de côté dans son panier,
 - ✓ soit reclasser les résultats.
- **Si le nombre d'ouvrages trouvés est grand, l'internaute pourra naviguer entre les différentes pages de résultats.**

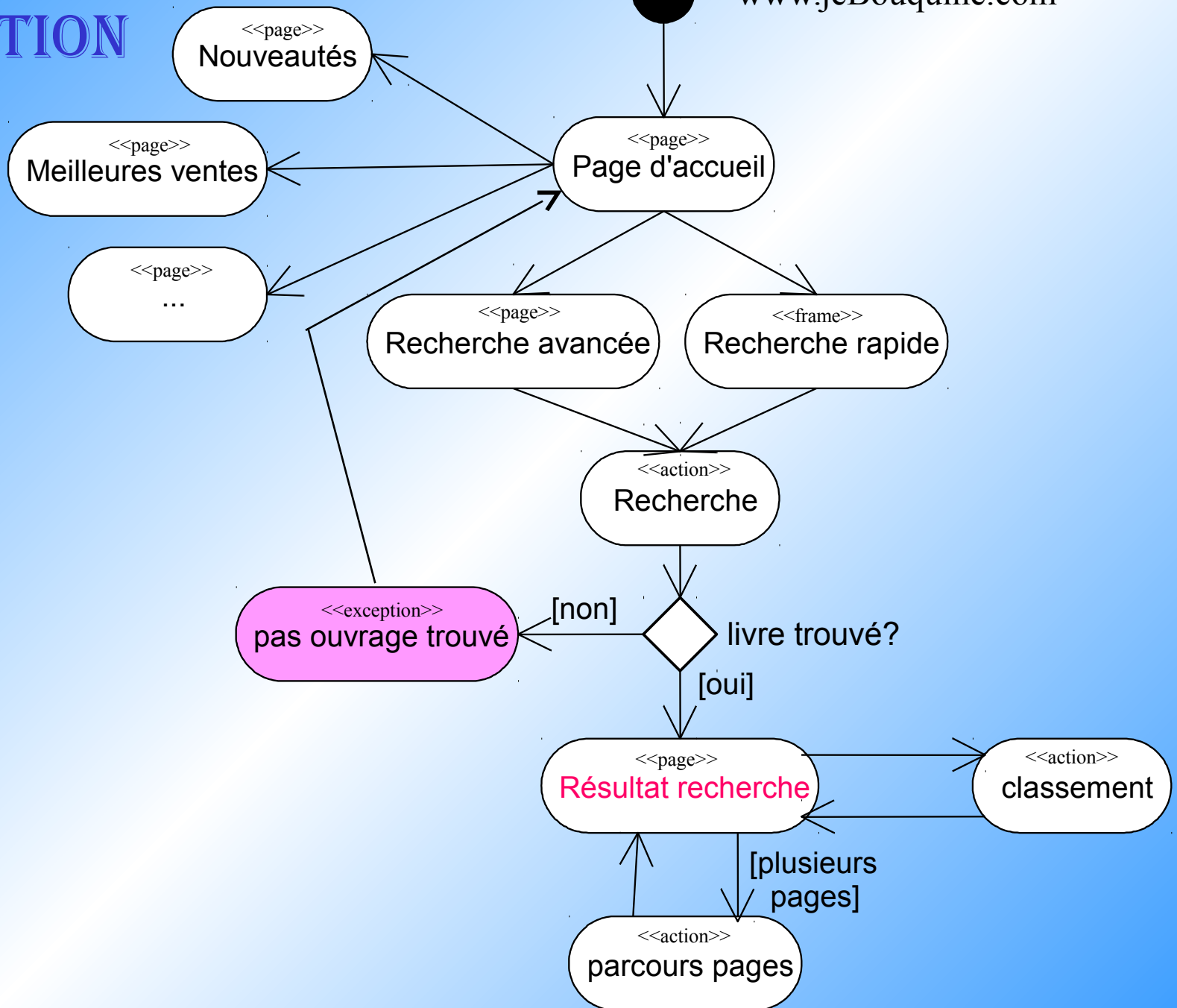
SOLUTION



SOLUTION

- Sur le résultat de la recherche, l'internaute pourra :
 - ✓ soit sélectionner un ouvrage pour obtenir sa fiche détaillée,
 - ✓ soit le mettre directement de côté dans son panier,
 - ✓ **soit reclasser les résultats.**
- Si le nombre d'ouvrages trouvés est grand, l'internaute pourra naviguer entre les différentes pages de résultats.

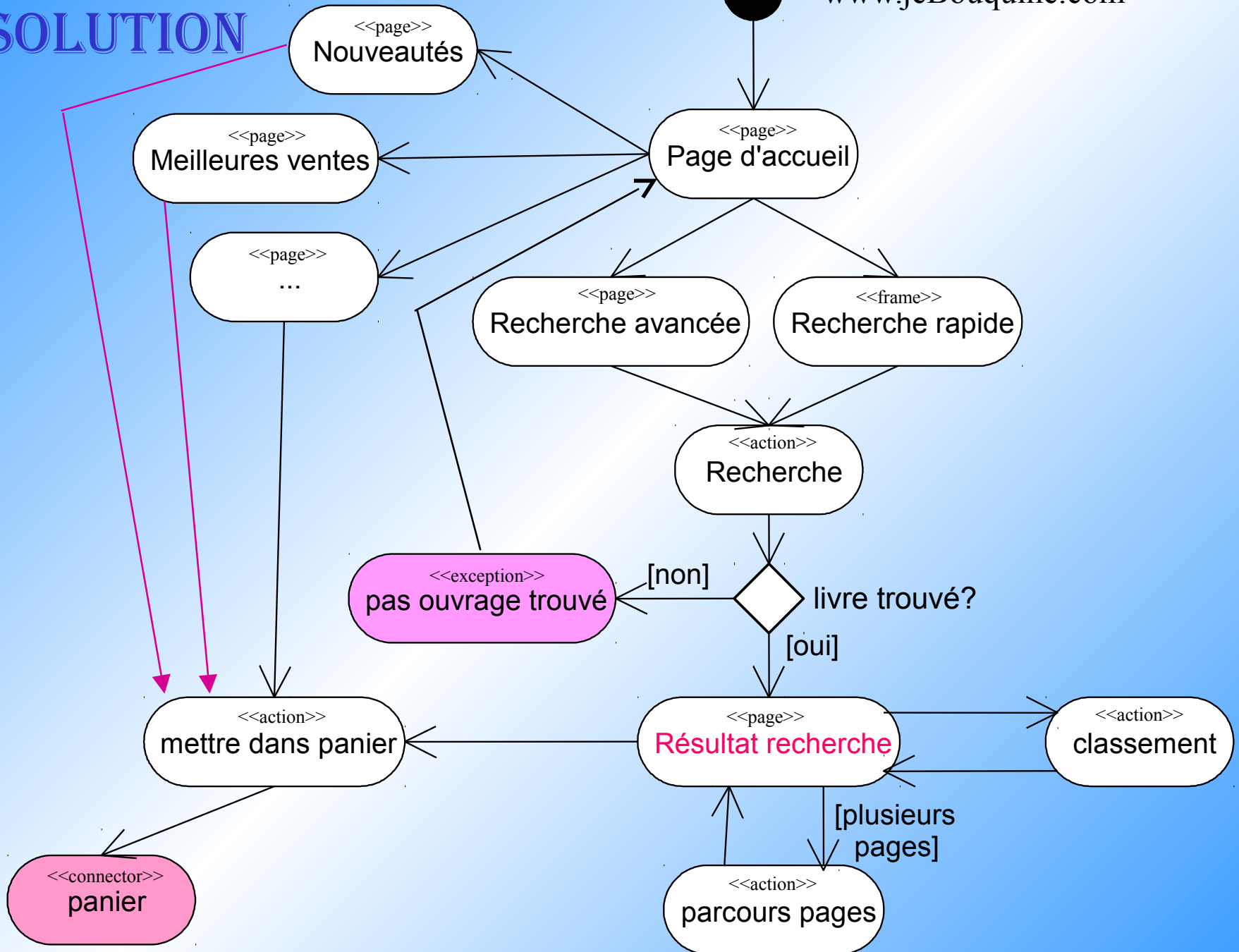
SOLUTION



SOLUTION

- Sur le résultat de la recherche, l'internaute pourra :
 - ✓ soit sélectionner un ouvrage pour obtenir sa fiche détaillée,
 - ✓ **soit le mettre directement de côté dans son panier,**
 - ✓ soit reclasser les résultats.
- Si le nombre d'ouvrages trouvés est grand, l'internaute pourra naviguer entre les différentes pages de résultats.

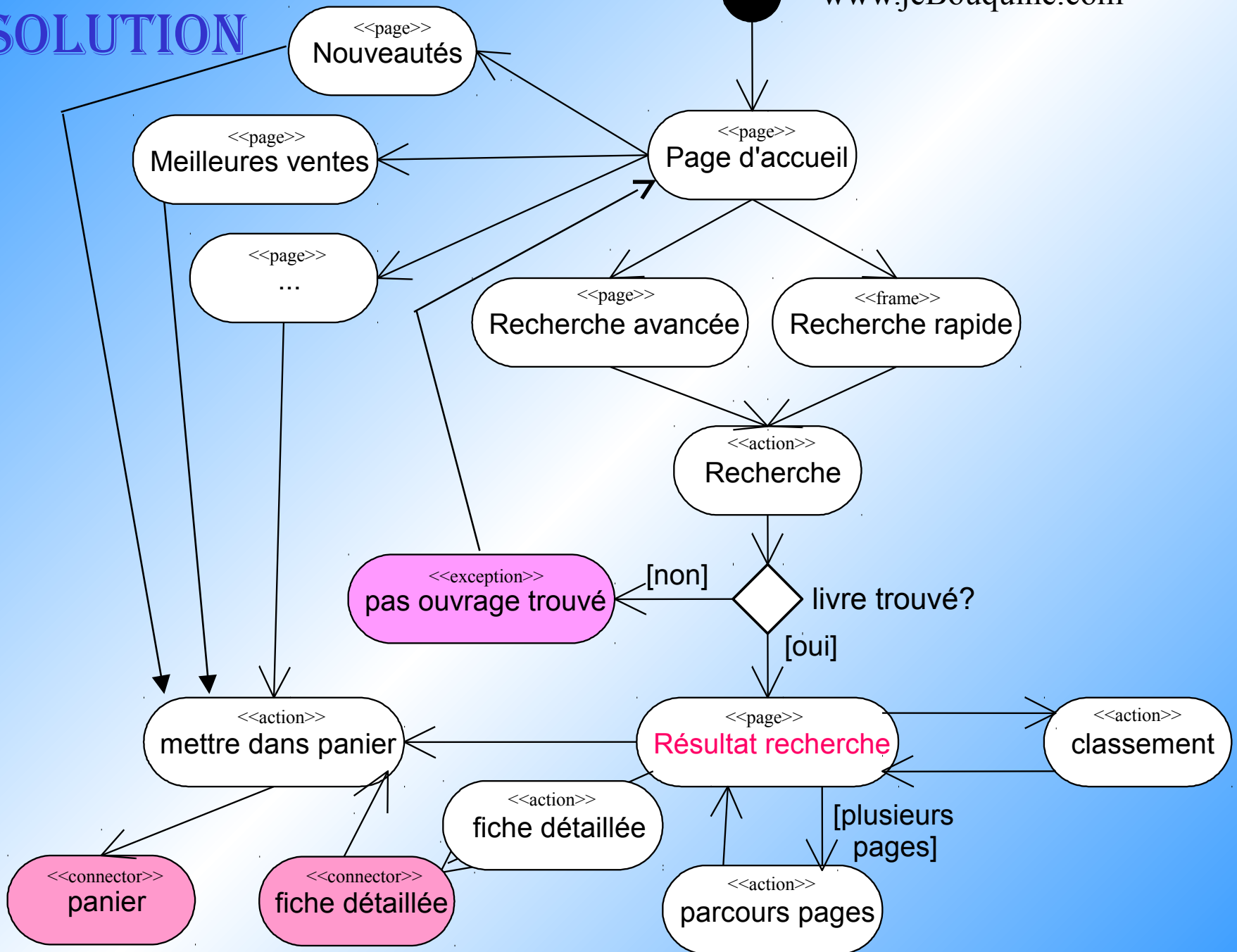
SOLUTION



SOLUTION

- Sur le résultat de la recherche, l'internaute pourra :
 - ✓ **soit sélectionner un ouvrage pour obtenir sa fiche détaillée,**
 - ✓ soit le mettre directement de côté dans son panier,
 - ✓ soit reclasser les résultats.
- Si le nombre d'ouvrages trouvés est grand, l'internaute pourra naviguer entre les différentes pages de résultats.

SOLUTION



SOLUTION

- On a choisi de détailler la navigation à partir de la fiche d'ouvrage dans un autre diagramme d'activités, en utilisant le concept de <<connector>>.
- De la même façon, la gestion du panier est représentée par un <<connector>> afin de ne pas compliquer le diagramme.

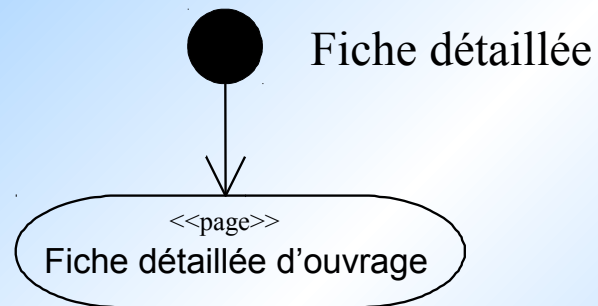
RECHERCHER DES OUVRAGES

- La fiche d'ouvrage permet :
 - ✓ de mettre le livre de côté dans le panier,
 - ✓ d'accéder à la fiche de l'auteur comprenant une biographie et l'ensemble des ouvrages qu'il a écrits.
 - ✓ d'accéder à des informations complémentaires telles que :
 - image agrandissable,
 - commentaires éventuels,
 - table des matières détaillée,
 - extraits de chapitre, etc.

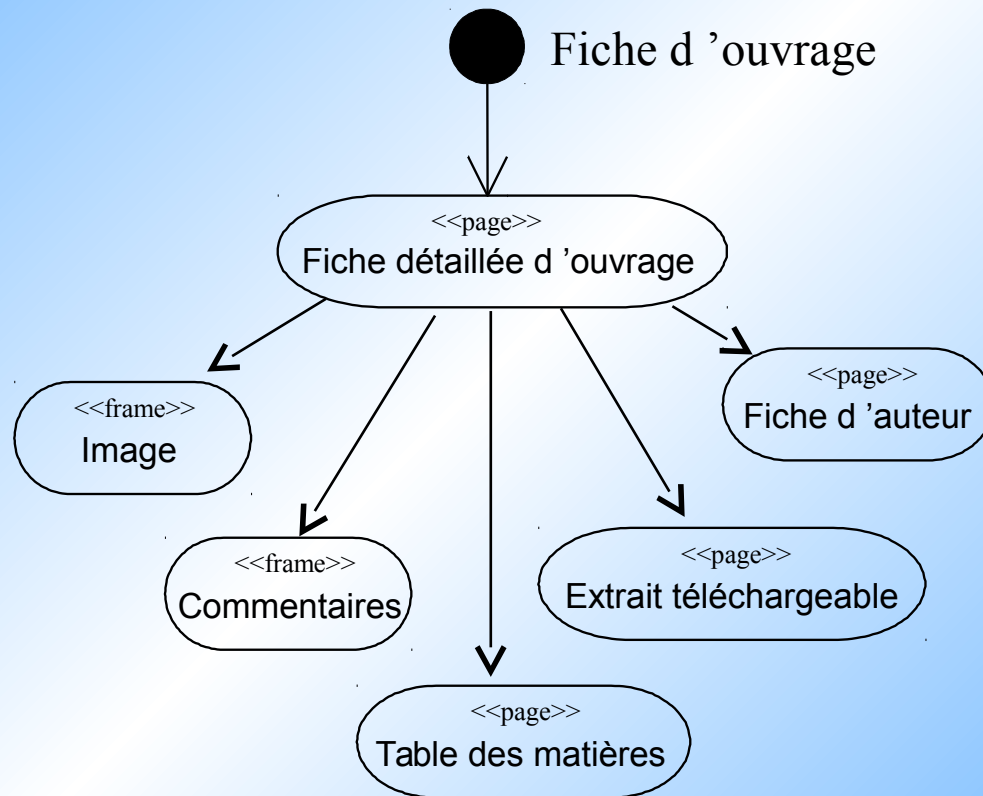
RECHERCHER DES OUVRAGES

Faire le diagramme de navigation pour le
connecteur
fiche détaillée

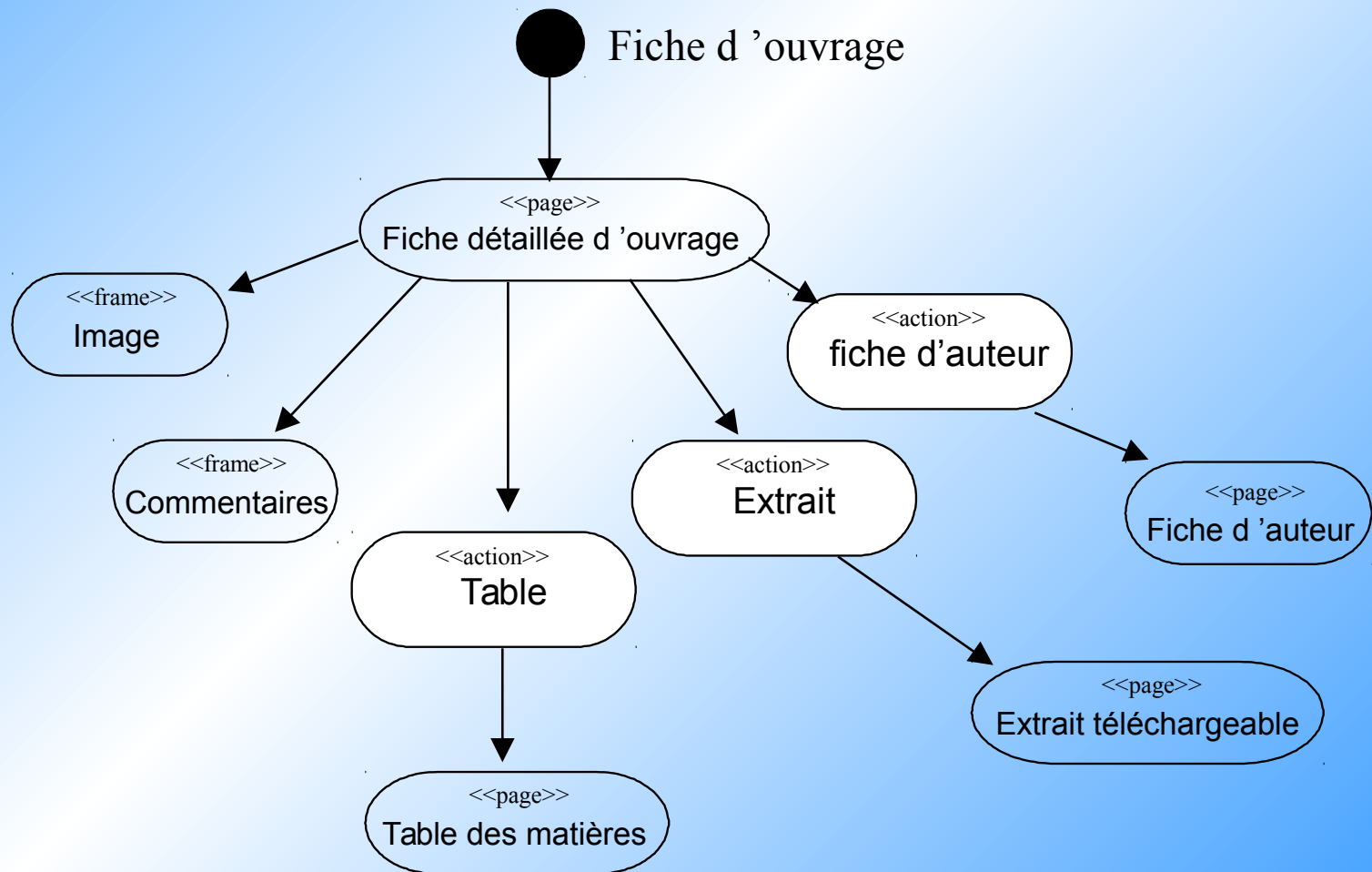
SOLUTION



SOLUTION



AUTRE SOLUTION



SOLUTION

- On n'indique pas tous les retours systématiques à la page précédente qui ajouteraient de nombreuses transitions triviales alourdissant le diagramme.
- De même, l'internaute a accès en permanence à l'aide en ligne. On considère toutes ces transitions comme implicites, pour nous concentrer sur celles qui ont une véritable signification métier.

GÉRER SON PANIER

Dans le diagramme d'activités du cas d'utilisation *Rechercher des ouvrages*, on a un connecteur *Panier*.

Détailler ce connecteur

GÉRER SON PANIER

L'internaute a la possibilité de :

- commander
- établir un devis
- modifier son panier
- vider son panier

L'accès au panier est disponible de partout, y compris de la page d'accueil.

Modéliser le fait qu'il est impossible de commander à partir d'un panier vide.

SOLUTION

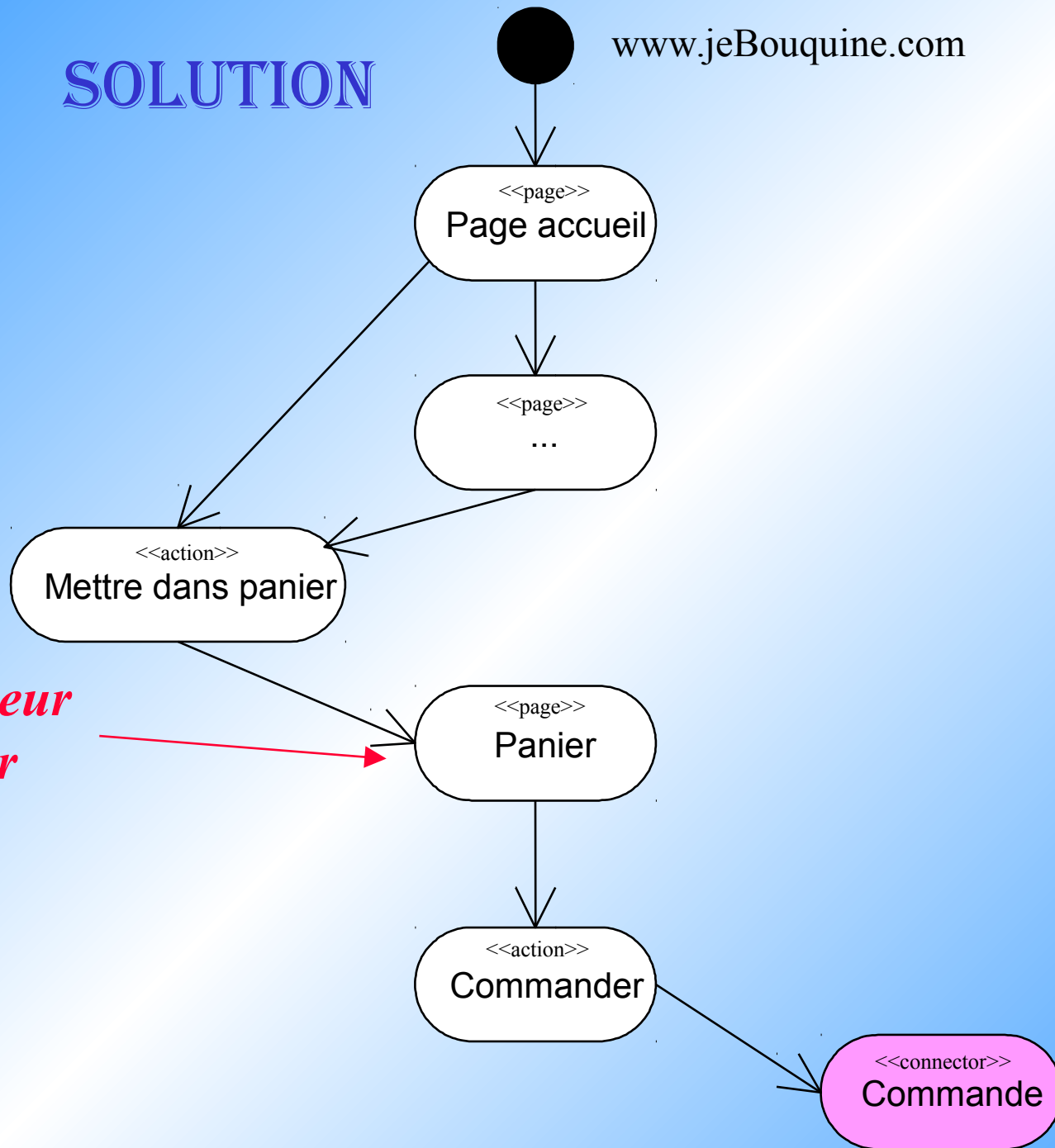
L'internaute a la possibilité de :

- **commander**
- établir un devis
- modifier son panier
- vider son panier

L'accès au panier est disponible de partout, y compris de la page d'accueil.

Modéliser le fait qu'il est impossible de commander à partir d'un panier vide.

SOLUTION



*Connecteur
Panier*

SOLUTION

L'internaute a la possibilité de :

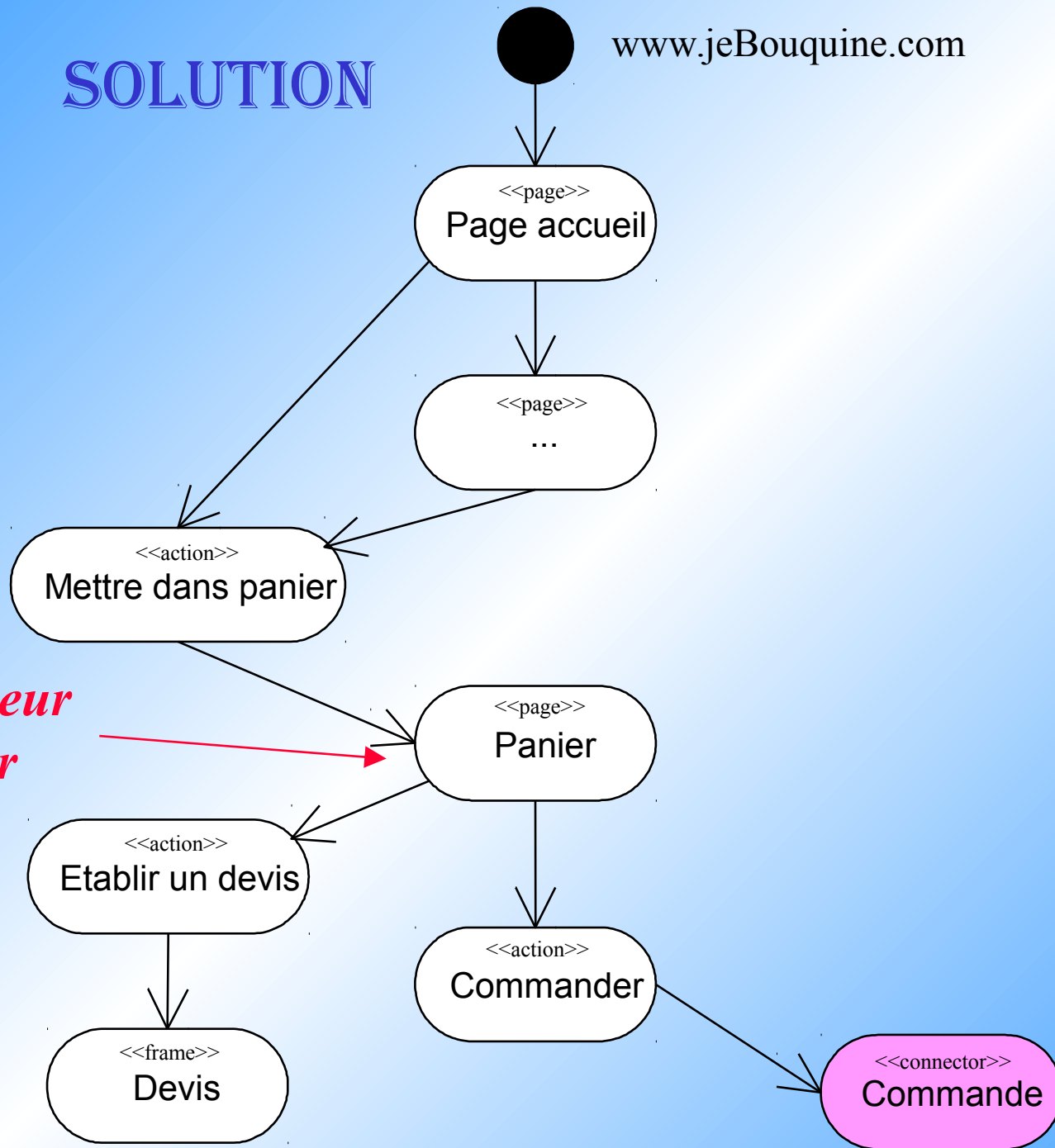
- commander
- **établir un devis**
- modifier son panier
- vider son panier

L'accès au panier est disponible de partout, y compris de la page d'accueil.

Modéliser le fait qu'il est impossible de commander à partir d'un panier vide.

SOLUTION

*Connecteur
Panier*



SOLUTION

L'internaute a la possibilité de :

- commander
- établir un devis
- **modifier son panier**
- vider son panier

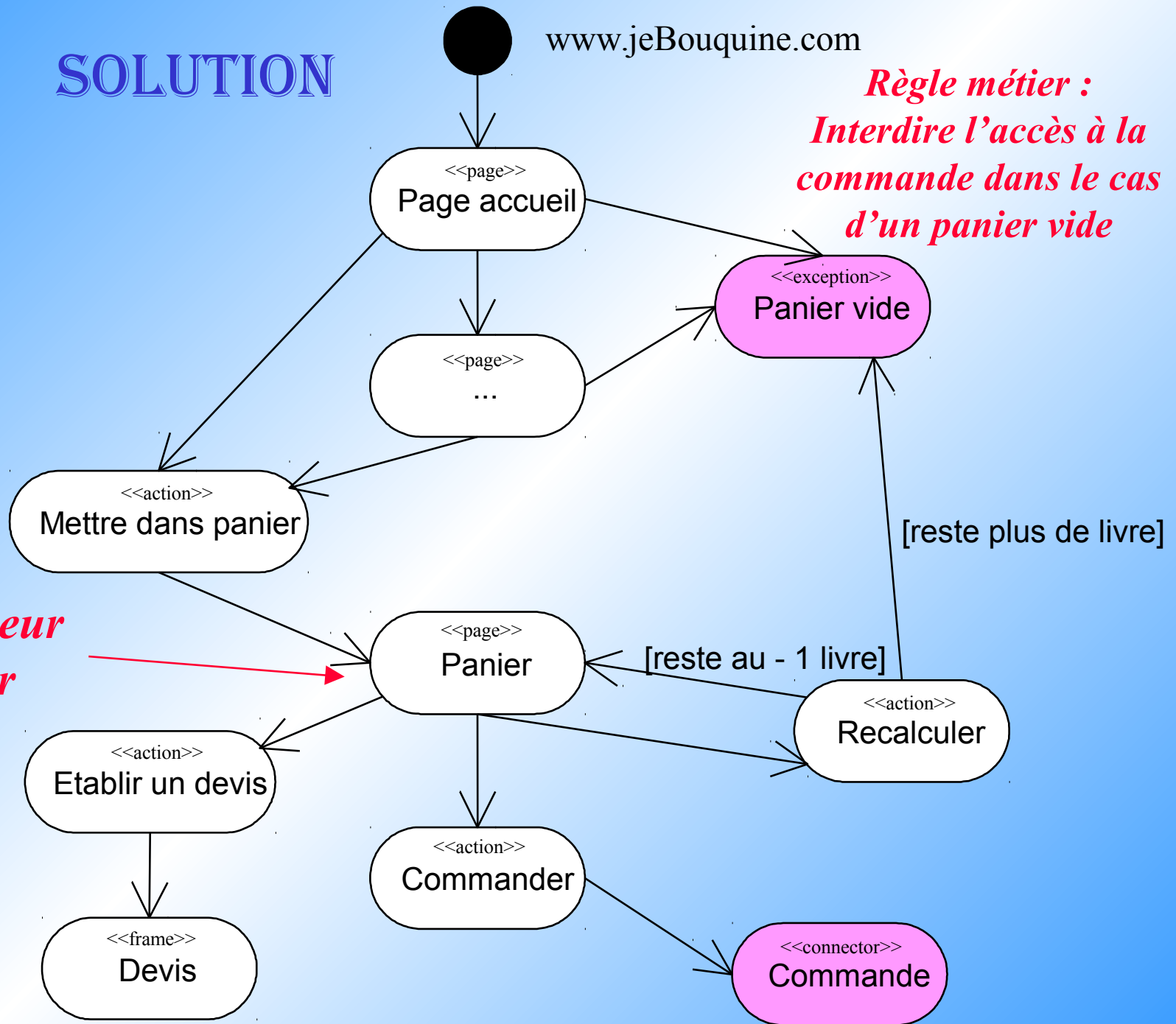
L'accès au panier est disponible de partout, y compris de la page d'accueil.

Modéliser le fait qu'il est impossible de commander à partir d'un panier vide.

SOLUTION

Règle métier :
Interdire l'accès à la
commande dans le cas
d'un panier vide

Connecteur
Panier



SOLUTION

L'internaute a la possibilité de :

- commander
- établir un devis
- modifier son panier
- **vider son panier**

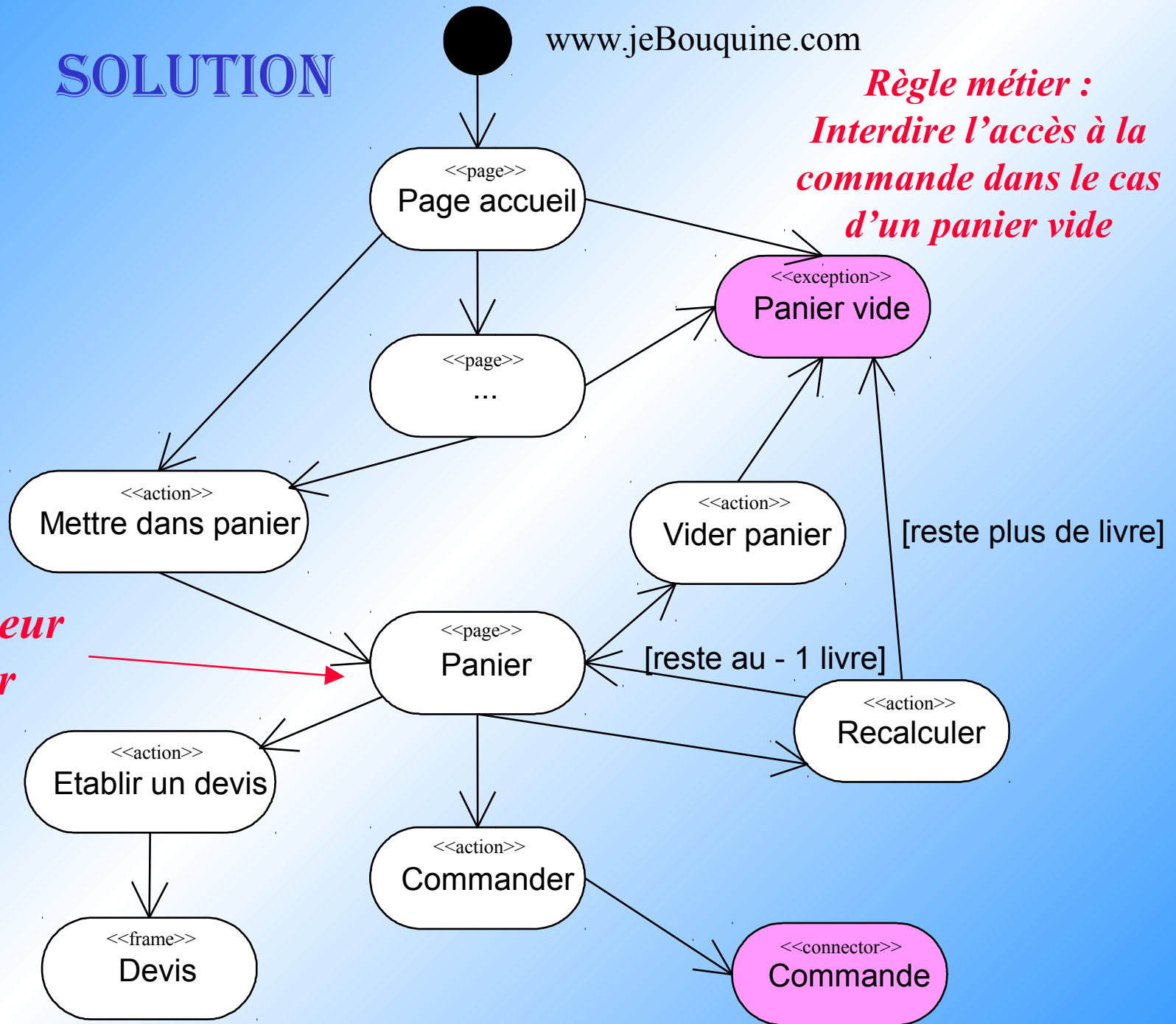
L'accès au panier est disponible de partout, y compris de la page d'accueil.

Modéliser le fait qu'il est impossible de commander à partir d'un panier vide.

SOLUTION

Règle métier :
Interdire l'accès à la
commande dans le cas
d'un panier vide

Connecteur
Panier



SOLUTION

- On repart de la page d'accueil pour envisager le cas du panier vide car l'accès au panier est disponible de partout, y compris de la page d'accueil.
- Il faut donc traiter une *<<exception>> Panier vide* qui empêche de passer à la page de commande.

SOLUTION

- Le <<connector>> *Panier* du diagramme d'activités *Rechercher des ouvrages* correspond à la <<page>> *Panier* du diagramme d'activités *Gérer son panier* et non pas au point d'entrée du diagramme.
- On a donc représenté la règle métier importante qui consiste à interdire l'accès à la commande dans le cas où le panier est vide.

EFFECTUER UNE COMMANDE

Faire le diagramme d'activités

EFFECTUER UNE COMMANDE

L'internaute commence par

- remplir ses informations (*nom, prénom, adresse,...*)
- puis saisit les informations concernant la commande (*adresse de livraison, mode de paiement,...*)

2 cas sont à envisager :

- le client a déjà effectué une commande dans le passé, ses caractéristiques seront donc affichées et il aura la possibilité de les modifier
- le client n'a jamais commandé, il faudra donc qu'il saisisse toutes ses informations

SOLUTION

