

Université de Msila – Département
d'Informatique

Génie logiciel (GL2)

(3ème année SI&ISIL)

Dr BOUNIF M-E

- **7. Le diagramme d'états-transitions**

- **La notion d'état**

L'état d'un objet correspond à un moment de son cycle de vie.

Pendant qu'il se trouve dans un état, un objet peut se contenter d'attendre un signal provenant d'autres objets. Il est alors inactif. Il peut également être actif et réaliser une activité. Une activité est l'exécution d'une série de méthodes et d'interactions avec d'autres objets. Elle est liée à un objectif.

Exemple

Lors d'un concours de saut d'obstacles, le cheval est dans l'état de repos avant de commencer la compétition. Il s'agit d'un état où il est inactif et attend l'ordre de départ.

Lorsqu'il saute un obstacle, le cheval est dans un état où il est actif et qui se termine lorsqu'il a fini de sauter l'obstacle.



L'ensemble des états du cycle de vie d'un objet contient un état initial. Celui-ci correspond à l'état de l'objet juste après sa création. Il peut également contenir un ou plusieurs états finaux. Ceux-ci correspondent à une phase de destruction de l'objet. Il arrive également qu'il n'y ait pas d'état final car un objet peut ne jamais être détruit.

La transition

Une transition est un lien orienté entre deux états qui exprime le fait que l'objet a la possibilité de passer de l'état d'origine de la transition à son état de destination. Lorsque l'objet réalise ce passage de l'état d'origine à l'état de destination, la transition est alors franchie



Une transition est généralement associée à un événement. Dans ce cas, la transition est franchie si l'objet se trouve dans l'état d'origine de la transition et s'il reçoit l'événement. Ce franchissement a lieu que l'objet soit actif ou non. S'il est inactif, le franchissement a lieu immédiatement. S'il est actif, le franchissement a lieu dès que l'activité associée à l'état est terminée.

Exemple

Lorsque le cavalier et le cheval reçoivent l'ordre de départ du juge, ils passent de l'état d'attente à l'état de course.

Une transition automatique n'est pas associée à un événement. Elle est franchie dès que l'objet a terminé l'activité liée à l'état d'origine. Dans ce cas, il est nécessaire d'associer une activité à l'état d'origine.

Exemple

Lorsque le cavalier et le cheval ont terminé le parcours, ils passent automatiquement dans l'état final.



Une transition réflexive possède le même état d'origine et de destination. Si la transition est associée à un événement, la réception de celui-ci ne fait pas changer l'état. Si la transition est réflexive et automatique, elle est alors utile pour réaliser une activité en boucle. Si un événement associé à une transition réflexive ne fait pas changer l'état de l'objet, sa réception peut provoquer une réaction comme l'appel d'une méthode de l'objet ou l'envoi d'un signal à d'autres objets.

Exemple

Tant que le cheval refuse de sauter un obstacle, il reste dans l'état de la course précédant cet obstacle. À chaque fois, le nombre de tentatives est augmenté de un.

L'élaboration du diagramme d'états-transitions

Le diagramme d'états-transitions représente le cycle de vie des instances d'une classe. Il décrit les états, les transitions qui les lient et les Événements qui provoquent le franchissement des transitions. Un tel diagramme n'est utile que pour les objets qui ont un cycle de vie. D'autres objets, purement porteurs d'information, ne changent pas d'état au cours de leur vie. Pour ces objets, il est inutile de concevoir un diagramme d'états-transitions.

1. La représentation graphique des éléments de base

Un état est représenté par un rectangle aux coins arrondis contenant son nom. La figure 7.1 montre la représentation graphique d'un état.

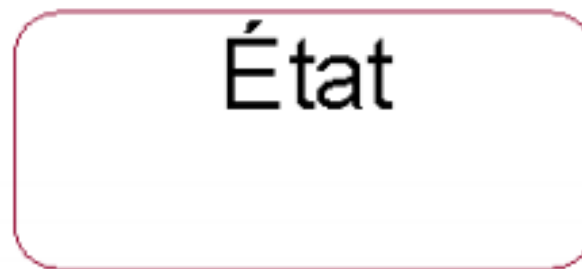


Figure 7.1 Représentation graphique d'un état

Dans un diagramme d'états-transitions, le premier état correspond à l'état initial de l'objet à l'issue de sa phase de création.

Cet état est unique dans un diagramme d'états-transitions. L'état initial est représenté par un point noir (voir figure 7.2).

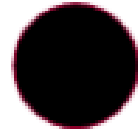


Figure 7.2 Représentation graphique de l'état initial

Un état final correspond à une étape où l'objet n'est plus nécessaire dans le système et où il est détruit. Tous les objets n'ont pas d'état final. C'est notamment le cas des objets permanents dans le système. Un état final est représenté par un point noir entouré d'un cercle (voir figure 7.3).



Figure 8.4 Représentation graphique d'un état final

Une transition entre deux états est représentée par un trait droit fléché reliant ces deux états (voir figure 7.4). L'événement qui détermine le franchissement de la transition est indiqué à proximité de la transition. Si la transition est automatique, aucun événement n'est indiqué.

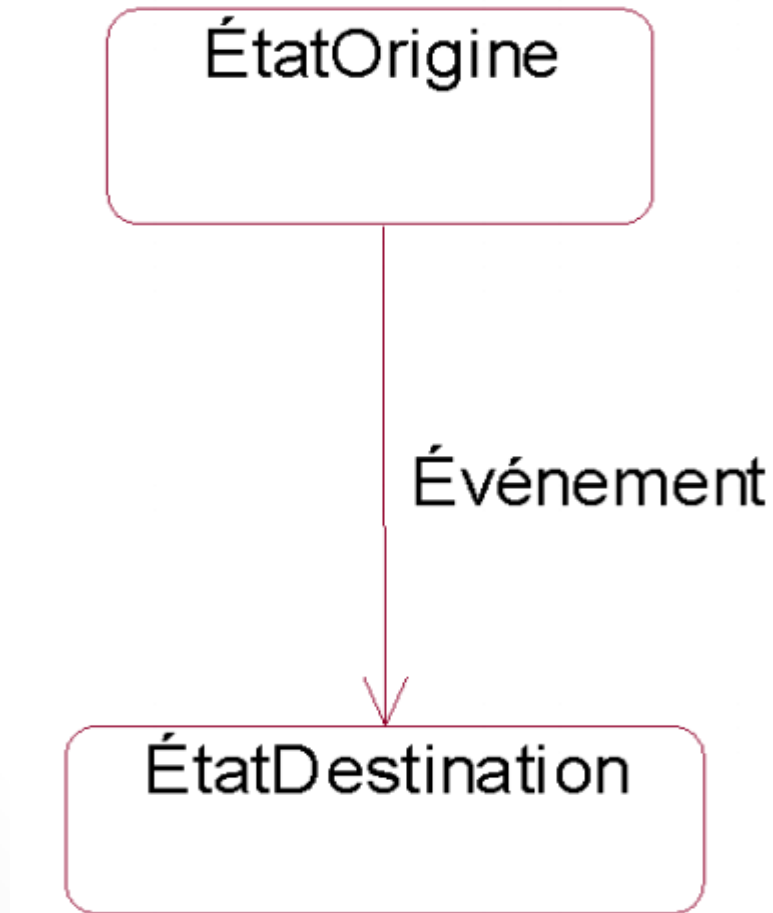


Figure 7.4 Représentation graphique d'une transition

Une transition réflexive possède le même état d'origine et de destination (voir figure 7.5).

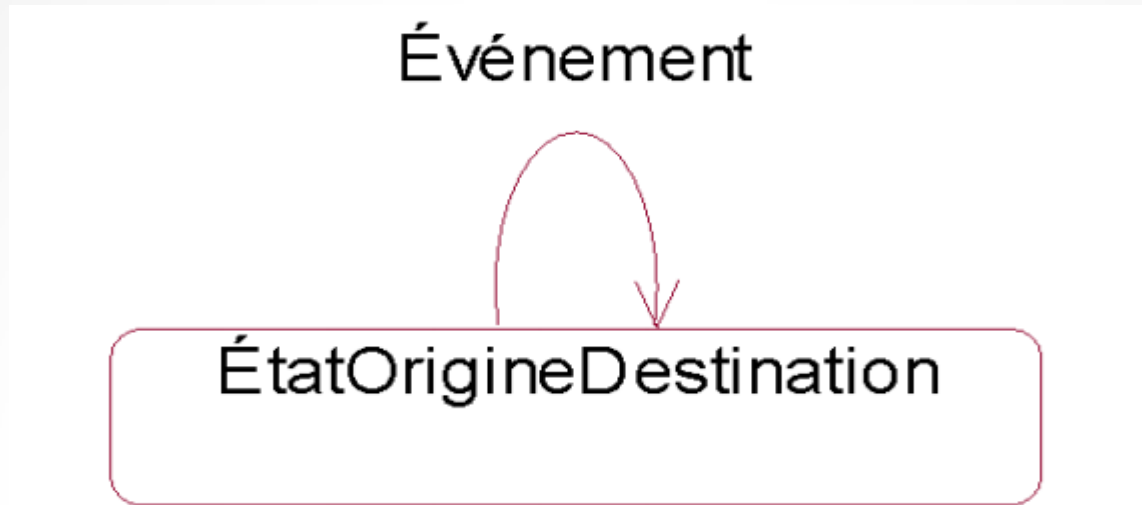
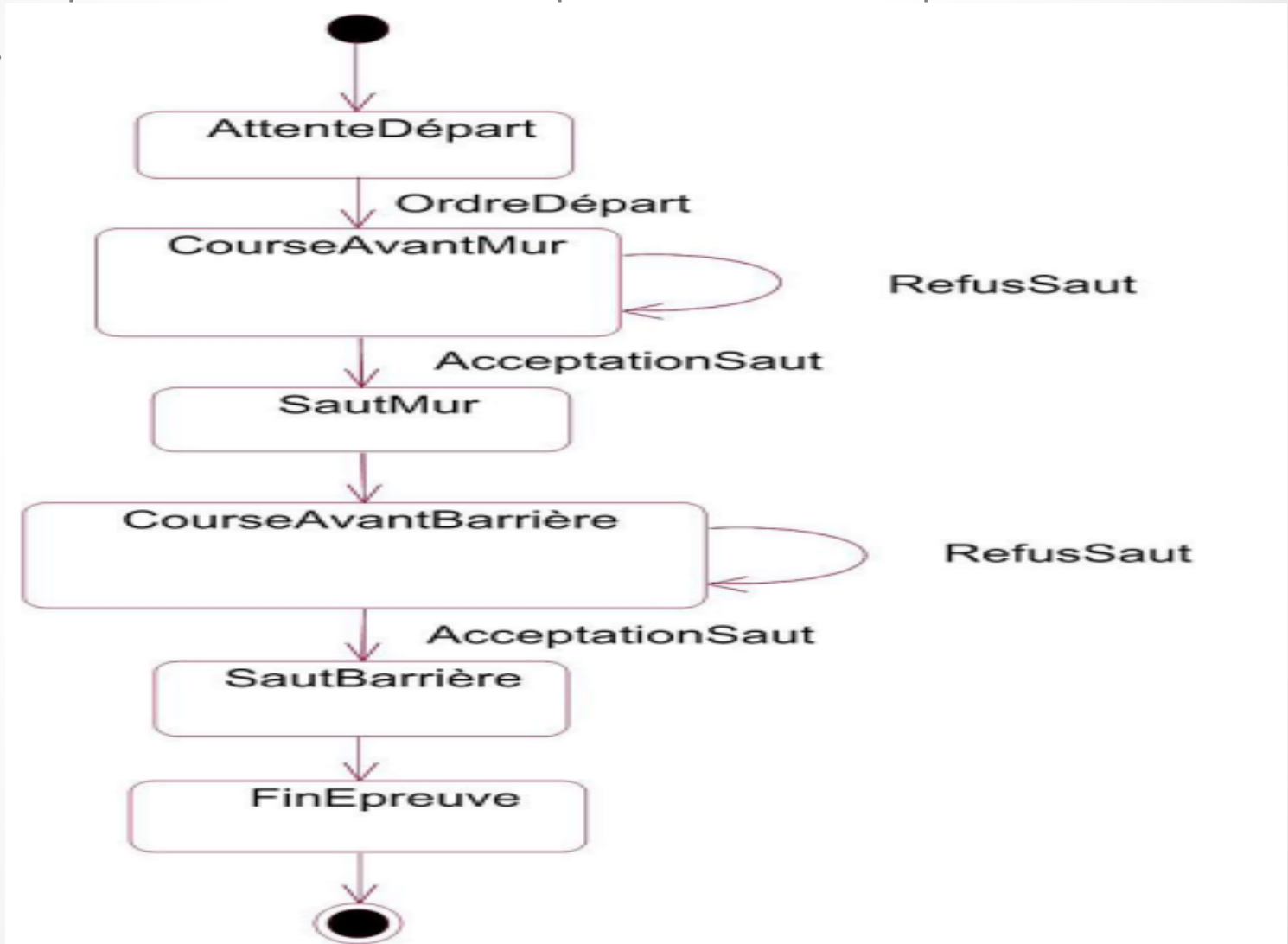


Figure 7.5 Représentation graphique d'une transition réflexive

Exemple

Dans un concours d'obstacles, l'épreuve consiste à demander à chaque concurrent de sauter deux ou trois obstacles différents. Il arrive que le cheval refuse de sauter un obstacle. Le concurrent peut alors recommencer le saut. La figure 7.6 représente le diagramme d'états-transitions décrivant une telle épreuve pour l'objet "concurrent de l'épreuve". Les deux obstacles sont respectivement le mur et la barrière. Ce diagramme contient des transitions réflexives et automatiques.

Le droit de sauter une nouvelle fois un obstacle est limité à deux tentatives après la tentative initiale et avant l'élimination de l'épreuve. Nous verrons par la suite comment prendre en compte cette contrainte.



• Figure 7.6 Exemple de diagramme d'états-transitions •

Les conditions de garde

Il est possible d'associer une condition à une transition, qui est alors appelée condition de garde. Pour que la transition soit franchie, il faut que la condition soit remplie en plus de la réception de l'événement associé, si celui-ci existe.

Une condition de garde est exprimée entre crochets. Si un événement est associé à la transition, la condition est exprimée à la droite du nom de l'événement (voir figure 7.7).

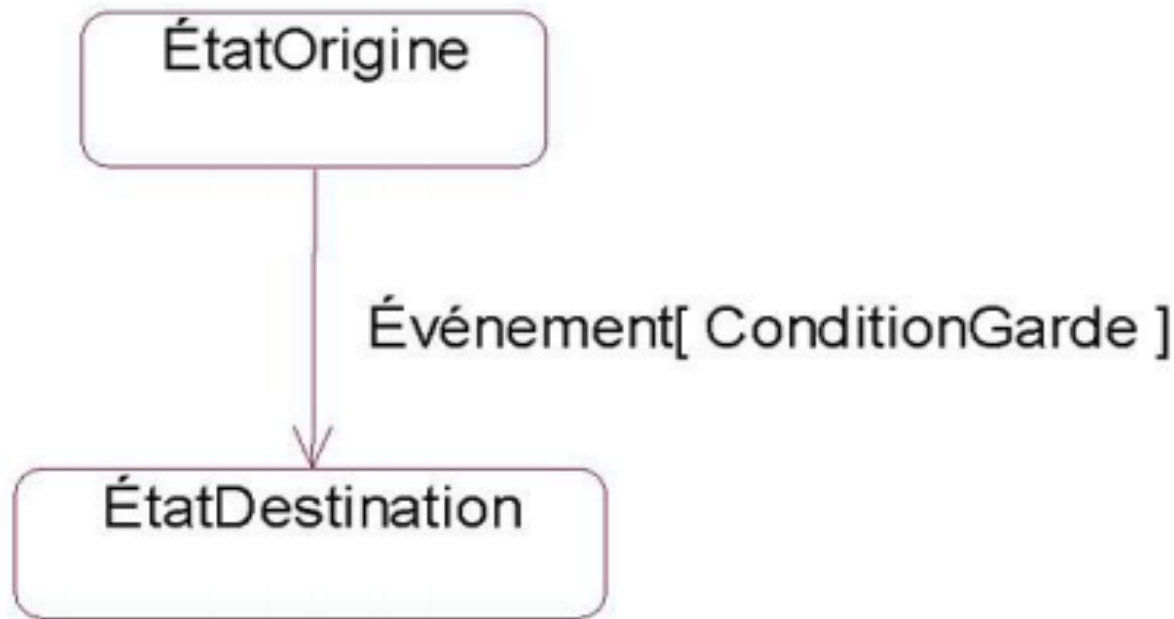


Figure 7.7 Condition de garde

Exemple

En cas de refus de sauter un obstacle, le concurrent a le droit de recommencer deux fois. Il est donc disqualifié après la troisième tentative si elle se solde par un refus.

La figure 8.9 illustre la prise en compte de ce nombre maximal de refus en reprenant l'exemple de la figure 8.6 (seul le premier obstacle est montré).

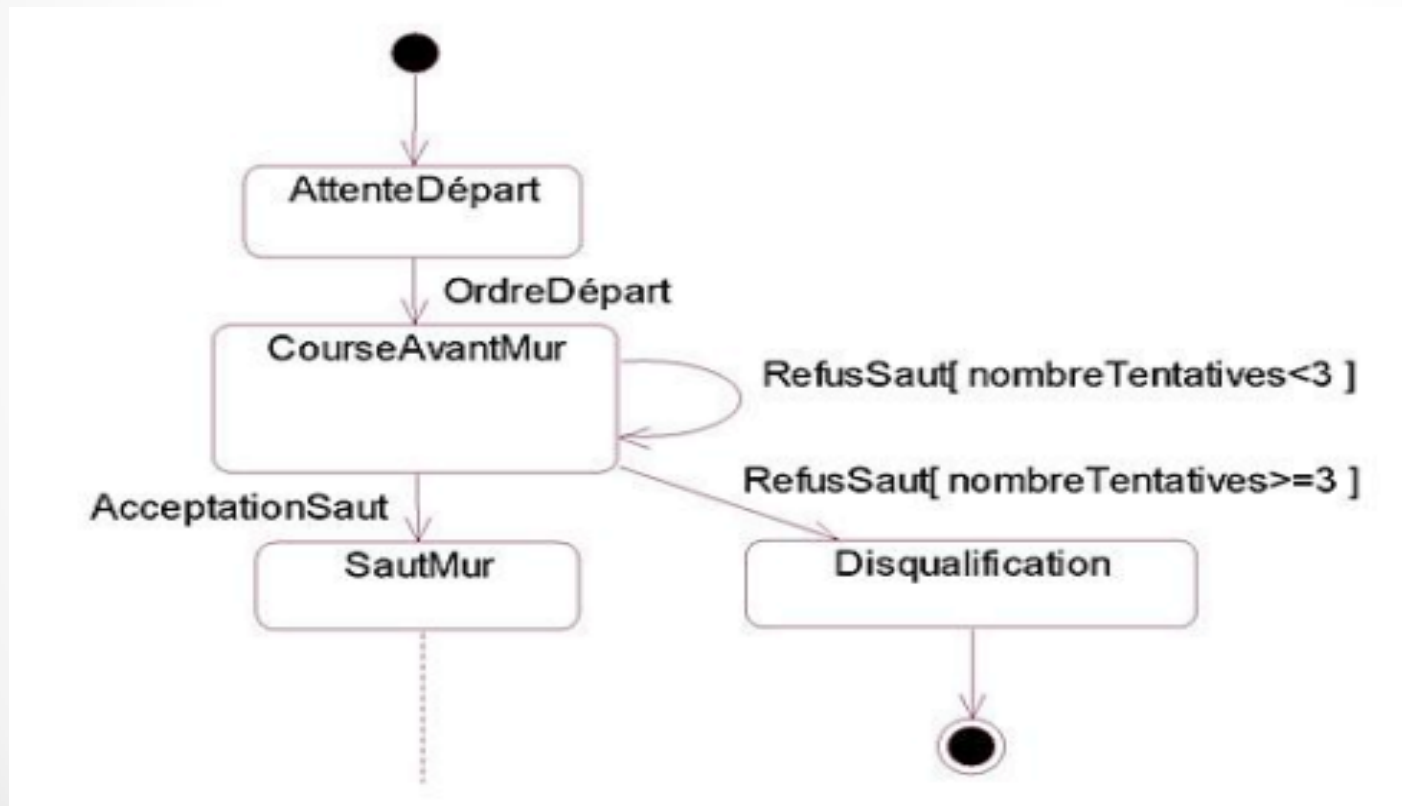


Figure 7.8 Exemple d'utilisation des conditions de garde

Activités liées à un état ou à un franchissement de transition

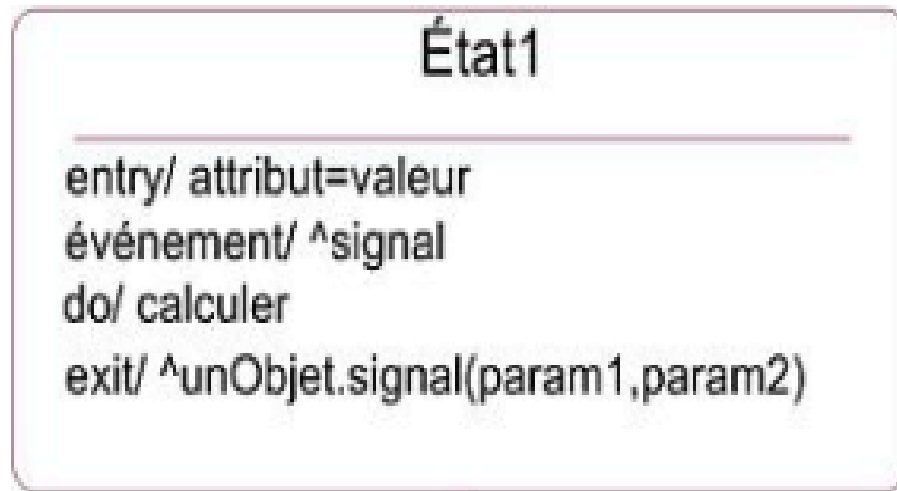
Il est possible de spécifier différentes activités :

- pendant un état ;
- lors du franchissement d'une transition ;
- à l'entrée et à la sortie d'un état ;
- au sein d'un état, lors de la réception d'un événement.

Une activité est une série d'actions. Une action consiste à affecter une valeur à un attribut, créer ou détruire un objet, effectuer une opération, envoyer un signal à un autre objet ou à soi-même, etc.

On désignera l'autre objet par son nom comme dans les diagrammes d'interaction étudiés au chapitre La modélisation de la dynamique.

La figure 7.9 illustre la représentation graphique de ces différentes possibilités. Une activité précédée du mot clé `entry/` est exécutée lors de l'entrée dans l'état. Une activité précédée du nom d'un événement est exécutée si cet événement est reçu. Le mot clé `do/` introduit l'activité réalisée pendant l'état. Une activité précédée du mot clé `exit/` est exécutée lors de la sortie de l'état. L'envoi d'un signal est précédé d'un `^` suivi du nom du signal. Il est également possible de spécifier une activité lors du franchissement d'une transition, qu'il faut faire précéder d'un `/`, et à la suite de l'événement et de la condition de garde, s'ils existent.



Événement[conditionGarde]/^unObjet.signal(param1,param2)



Figure 7.9 Activités exécutées pendant un état ou lors du franchissement d'une transition

Exemple

La figure 8.11 illustre l'utilisation de ces activités au sein d'un état ou lors du franchissement d'une transition. Ceci permet notamment de gérer la valeur des attributs `nombrePointsPénalité` et `nombreTentatives` de la classe `Concurrent`. Le nombre de points de pénalité est augmenté si le mur est renversé, ce qui se traduit par la réception de l'événement `renversement` pendant l'état `SautMur`. Le nombre de tentatives est initialisé à un, puis augmenté à chaque refus de sauter lors de la transition correspondante.

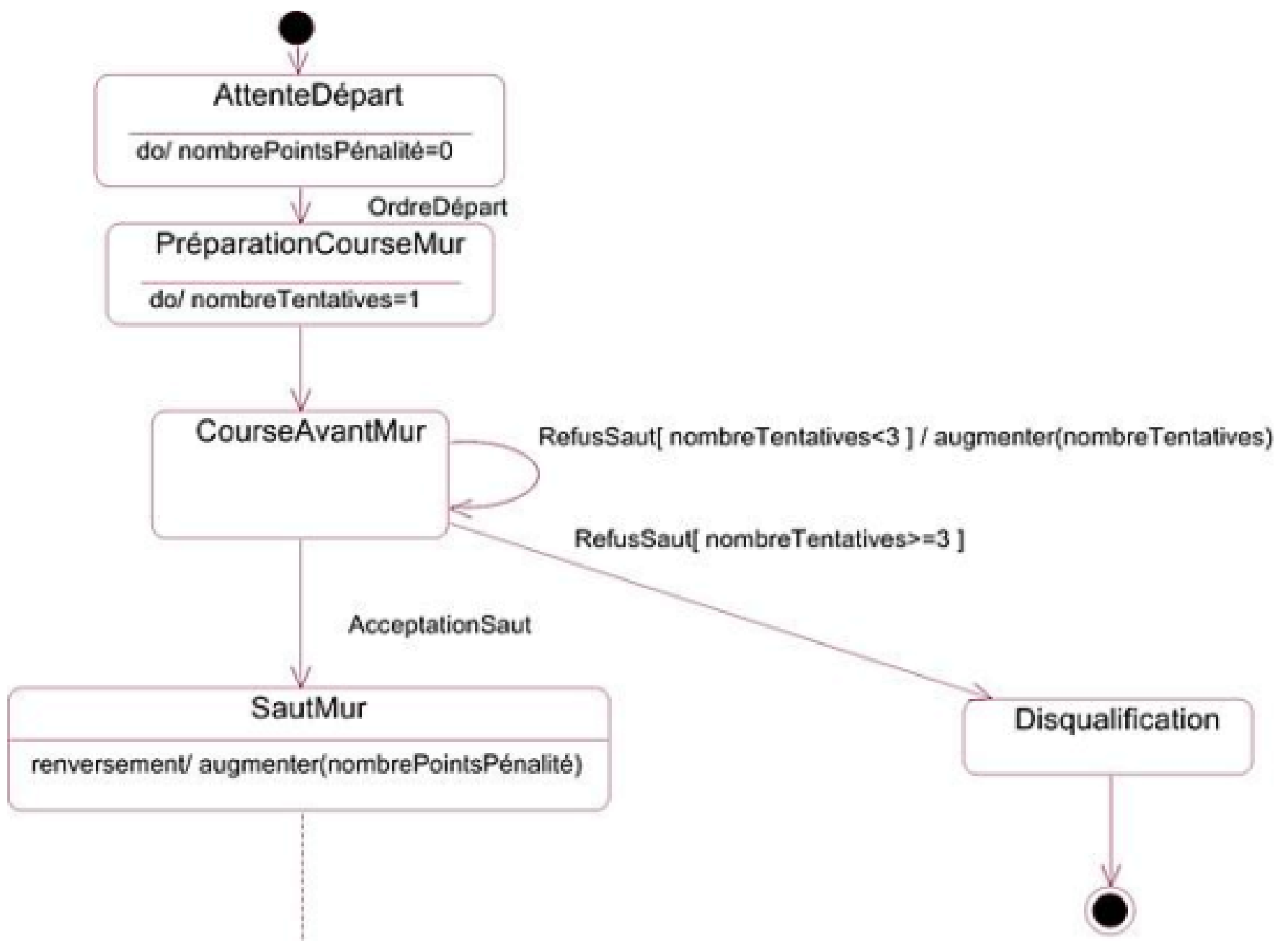


Figure 7.10 Exemple d'activités au sein d'un état ou lors du franchissement d'une transition

États composés

Un état peut être décrit lui-même par un diagramme d'états-transitions. Un tel état est appelé un état composé. Les états qui le composent sont appelés sous-états. Le principe est simple : dès que l'objet passe dans l'état composé, il passe également dans le sous-état initial du diagramme interne d'états-transitions. Si l'objet franchit une transition qui fait sortir de l'état composé, il quitte également les sous-états.

La figure 7.11 illustre un état composé. Un diagramme interne d'états transitions peut soit ne pas avoir d'état final, soit avoir un ou plusieurs états finaux.

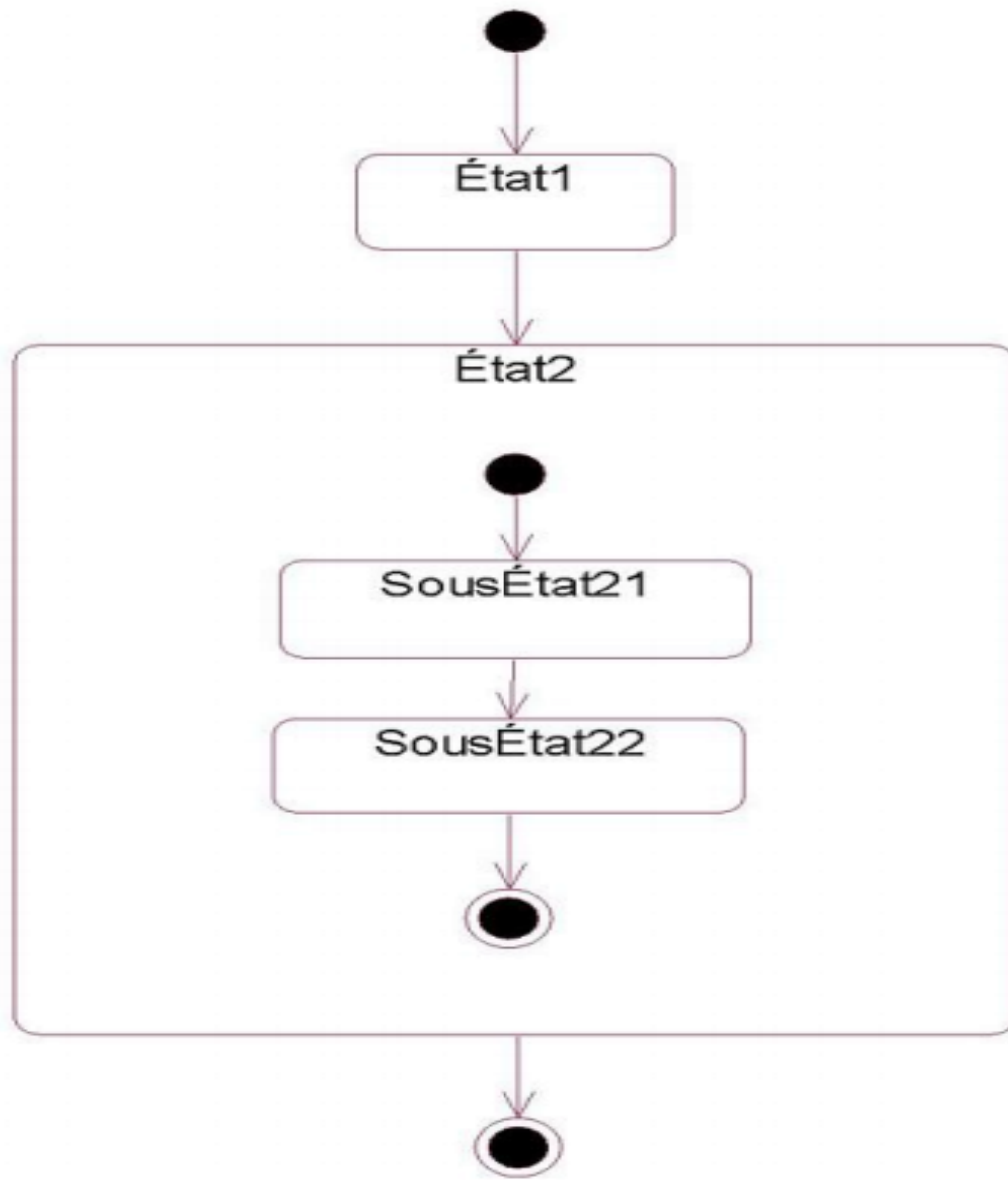


Figure 6.42 Interface et sous-classes distinctes de réalisation

Il est possible, lorsque un objet quitte un état composé, de mémoriser le sous état actif pour y revenir. Pour cela, il faut utiliser le sous-état spécial de mémoire H qui représente dans l'état composé le dernier sous-état actif mémorisé. La figure 7.12 illustre ce sous-état spécial de mémoire.

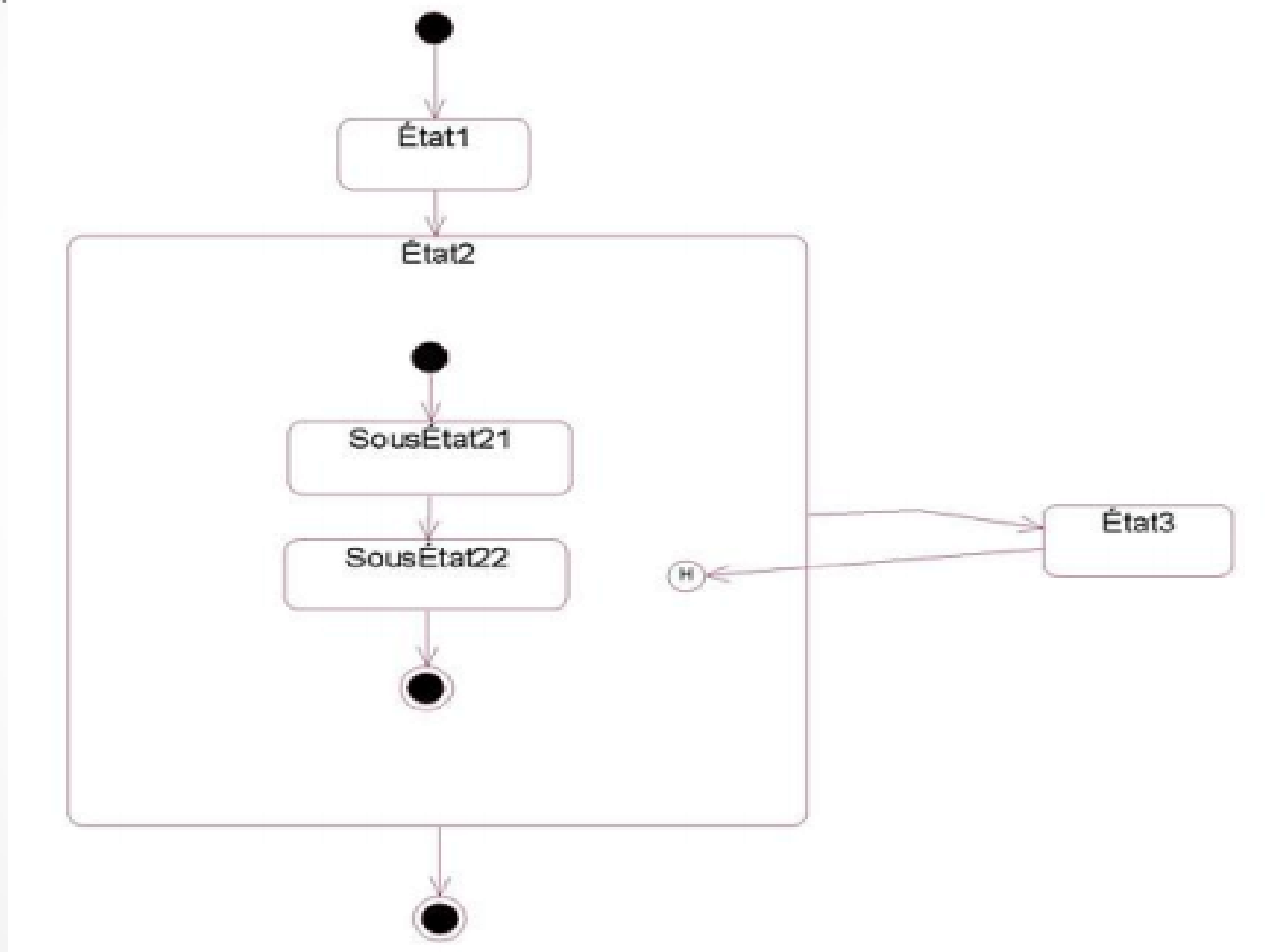


Figure 7.12 Sous-état de mémoire

Un sous-état peut être lui-même composé de sous-états. Dans ce cas, il existe deux sous-états de mémoire H et H*. Le premier permet de revenir au sous-état qui se trouve au niveau le plus élevé tandis que le second permet de revenir au sous-état imbriqué.

Exemple

Après l'ordre de départ et jusqu'au dernier saut, un concurrent est dans l'état Concours. À tout moment, il peut être disqualifié mais cette disqualification doit être confirmée (par exemple, en cas de contestation). Si elle est annulée, l'épreuve repart de l'état dans lequel elle s'était arrêtée.

La figure 7.13 illustre ce fonctionnement en utilisant un sous-état de mémoire pour revenir au dernier sous-état du concours si une disqualification est annulée.

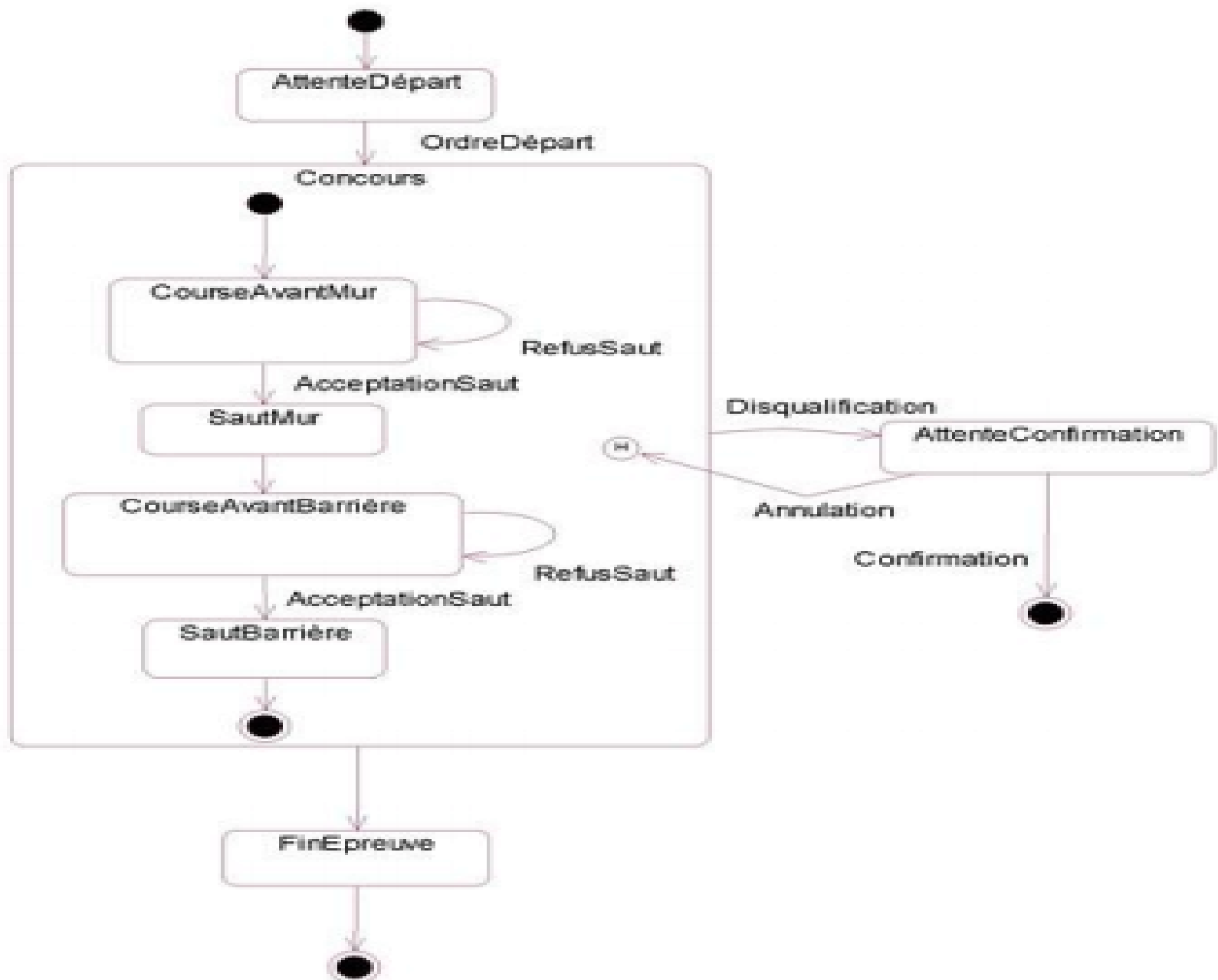


Figure 7.13 Exemple de sousétats de mémoire

Au sein d'un objet composé, il est possible d'avoir des sous-états qui évoluent en parallèle. Pour cela, il existe une transition de type fourche qui possède plusieurs sous-états de destination. Une fois franchie, l'objet se trouve dans tous les sous-états de destination. La transition de type synchronisation possède plusieurs sous-états d'origine et un seul état de destination. Il faut que l'objet se trouve dans tous les sous-états d'origine pour que la transition soit franchie. La figure 7.14 fournit la représentation de ces deux types de transition.

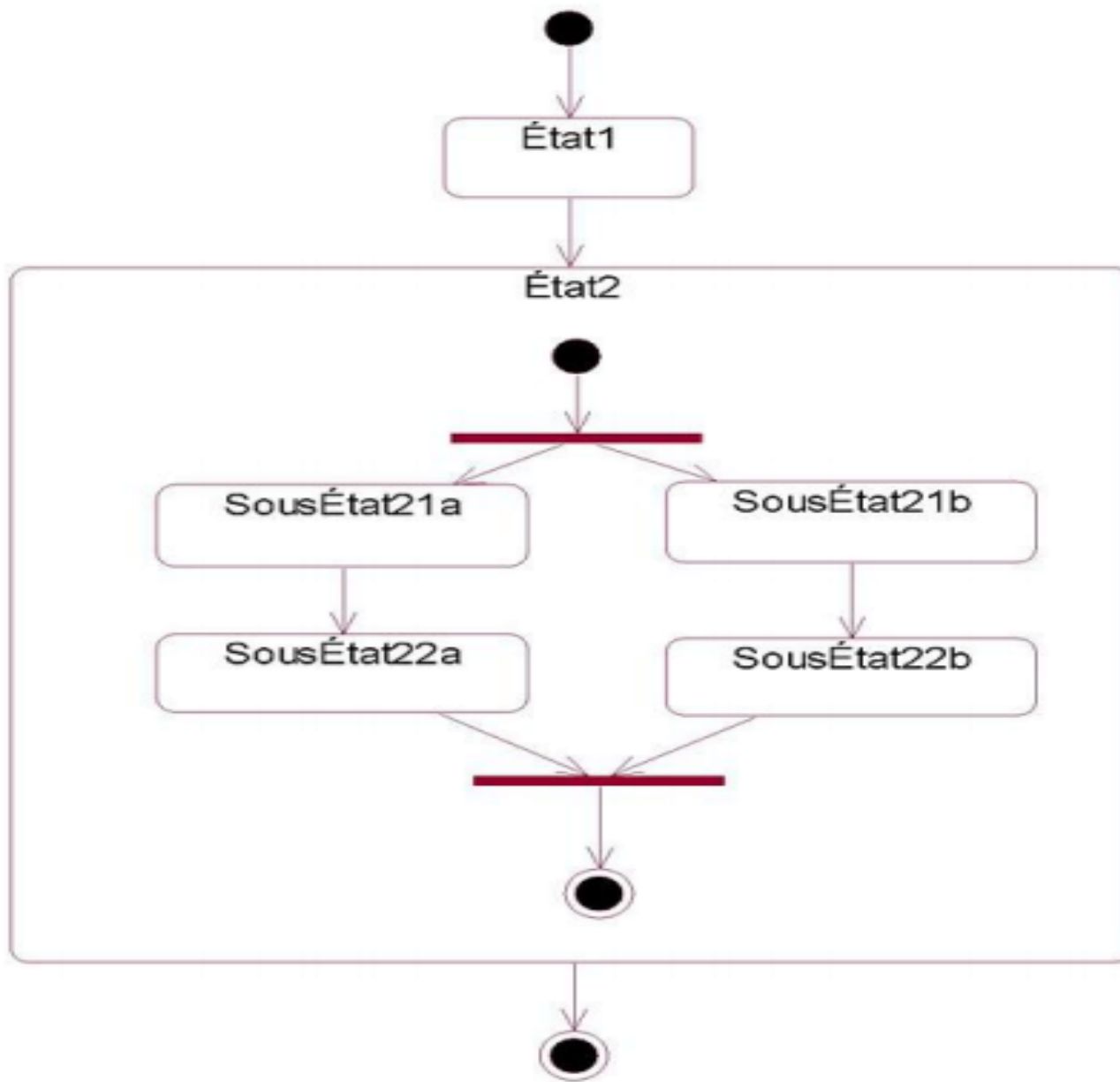


Figure 7.14 Sous-états parallèles et transitions de fourche et de synchronisation

Exemple

Un saut peut être décomposé en sous-états qui sont différents pour le cheval et le cavalier mais qui ont lieu simultanément. Ceci est illustré à la figure 7.15. Dans l'état SautMur, une transition de fourche permet de distinguer les sous-états du cheval et du cavalier. Le cheval est d'abord dans le sous-état Approche puis il se soulève lorsqu'il se situe dans le sous-état Enlevé. Enfin, il passe dans le sous-état Plané. Le cavalier reste dans le sous-état PositionSaut pendant le saut.

Une transition de synchronisation est franchie à la fin du saut.

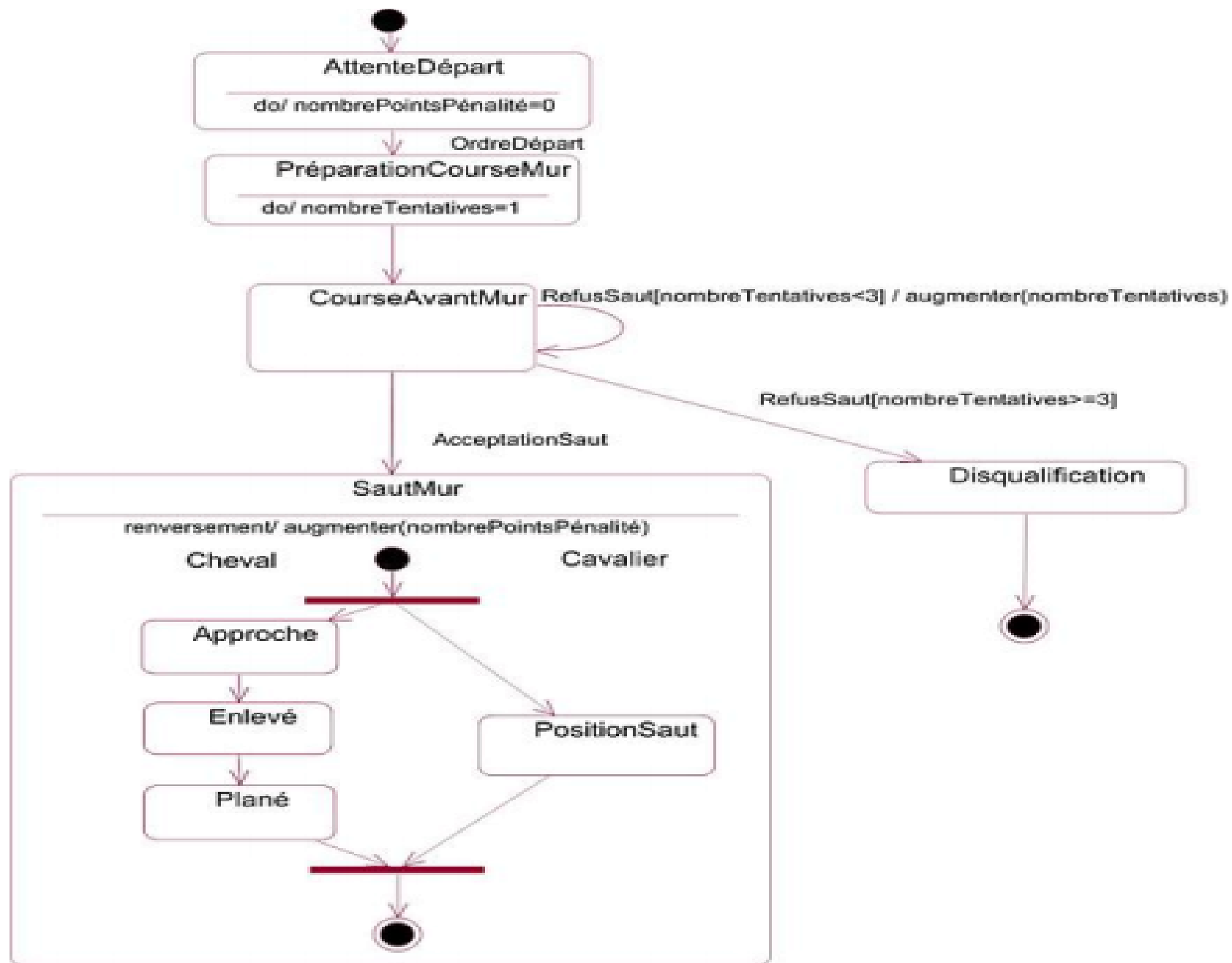


Figure 7.15 Exemple de sousétats parallèles

- **Conclusion**

Le diagramme d'états-transitions décrit le cycle de vie des objets chargés d'assurer la dynamique du système.

Cette description du cycle de vie est réalisée séparément pour chacun de ces objets. Cette modélisation est très importante pour s'assurer que les objets puissent répondre aux interactions décrites dans les diagrammes de séquence .