

GÉNIE LOGICIEL ORIENTÉ OBJET (GLO-2004)
ANALYSE ET CONCEPTION DES SYSTÈMES ORIENTÉS OBJETS (IFT-2007)

Automne 2016

Module 13 – Diagrammes d'états

Martin.Savoie@ift.ulaval.ca

Bachelier Génie logiciel, Chargé de cours,
département d'informatique et de génie logiciel

UML : 14 types de diagrammes (ouf!)

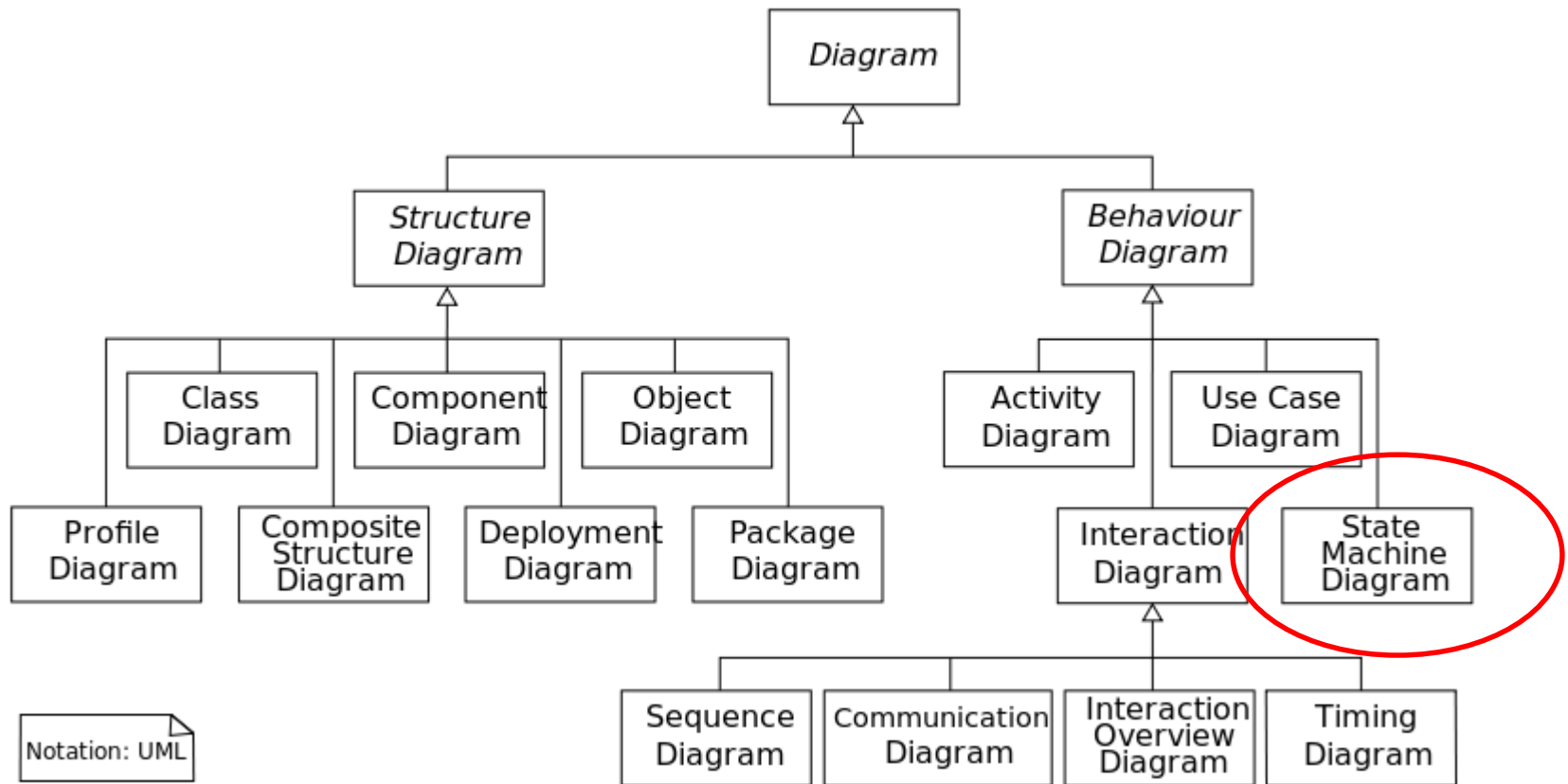


Diagramme d'états / machine à états

- Un **diagramme d'état** permet de représenter, pour un objet, un système, un sous-système, ou un acteur:
 - les **états** que peut prendre cet objet;
 - les changements d'états possibles (**transitions**) en réaction à des **événements**;
 - son comportement (**action**) face à un événement selon l'état dans lequel il se trouve;
- Autrement dit, on représente le **cycle de vie** de l'objet.

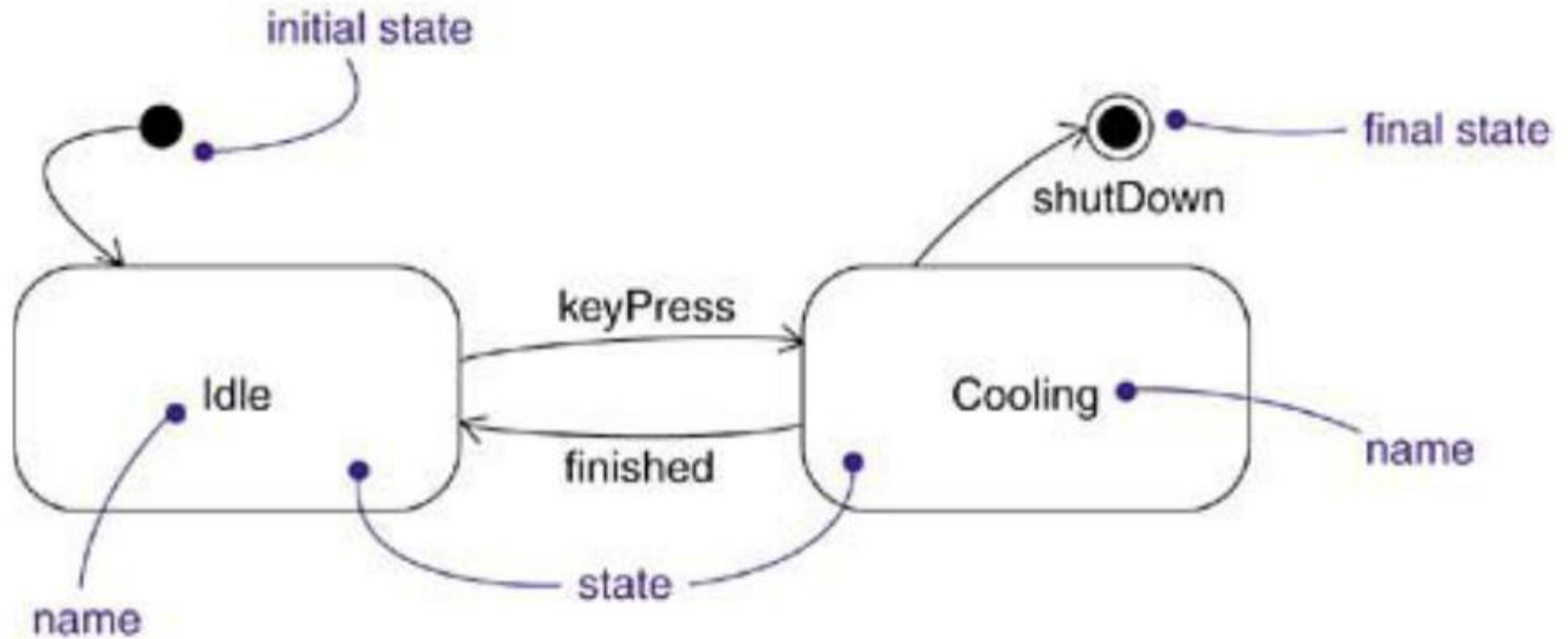
Dans le processus unifié

- Obligatoire nulle part
- Peut servir à clarifier différents aspects liés à n'importe quel modèle (ça dépend du projet):
 - Modélisation métier / modélisation des processus d'affaires
 - Modèle de conception (surtout celui-là!)
 - Etc..

	Activités (appelées <i>disciplines</i> dans le Processus Unifié)	Modèles et artefacts générés
Analyse	Modélisation domaine d'affaires / Business modeling / Modélisation métier	Modèle du domaine: (1) diagramme de classe « conceptuel », (2) parfois un diagramme d'activités
	Analyse des besoins / Exigences / Requirements	(3) Énoncé de vision
		Modèle de cas d'utilisation / Use-case model : (4) diagramme des cas d'utilisation, (5) texte des cas d'utilisation, (6) diagramme de séquence système
		(7) Spécifications supplémentaires
		(8) Glossaire
	Design / Conception	Modèle de conception / Design model : (9) diagrammes de classes, (10) diagrammes d'interaction, (11) tout autre diagramme UML pertinent selon le contexte
	Implémentation	(12) Code
	...	

Surtout utile ici

Exemple– Système de refroidissement



Est-ce que tout objet ou système devrait avoir son diagramme d'état?

- Non!
- La plupart des objets sont **mono-état** (également appelé **état-indépendants / state-independent**): ils réagissent toujours de la même façon suite à un événement
- D'autres objets sont **multi-états / état-dépendant / state-dependent**: leur comportement varie selon l'état / mode dans lequel ils se trouvent.
Par exemple:
 - Équipements physiques contrôlés par des logiciels
 - Transactions et objets métier apparentés
 - Une personne
 - Conversation, communication, protocole

États

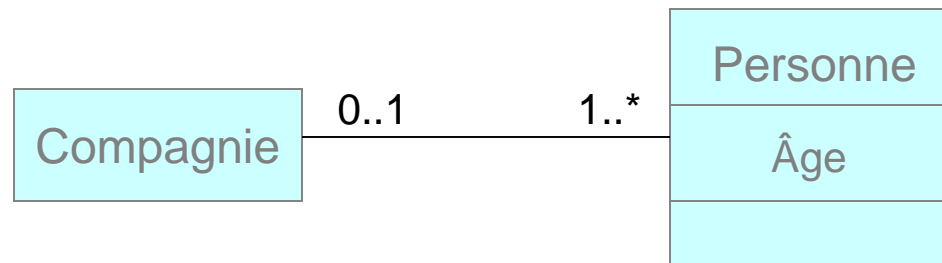
- Un état représente une situation bien définie durant la vie de l'objet
- Un état est généralement **durable** et **stable**.
- Un objet est généralement dans un état donné pour un certain temps.
- Un état possède un nom
- Un état est représenté par un rectangle dont les coins sont arrondis



Nom état

États

- L'état d'un objet est 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



Transitions et événements

- Les états sont reliés par des connexions unidirectionnelles appelées **transitions**
- Un objet, dans un état donné, attend l'occurrence d'un événement pour réaliser la transition



- Un événement correspond à l'occurrence d'une situation particulière dans le domaine du problème
- Le passage d'un état à un autre est supposé instantané car le système doit être toujours dans un état connu

La signature des événements

- Elle consiste en un **nom** d'événement, une **liste de paramètres** séparés par des virgules.

Exemples:

```
imprime (facture)  
dessine (f:Forme, c:couleur)  
redessine ()  
redessine
```

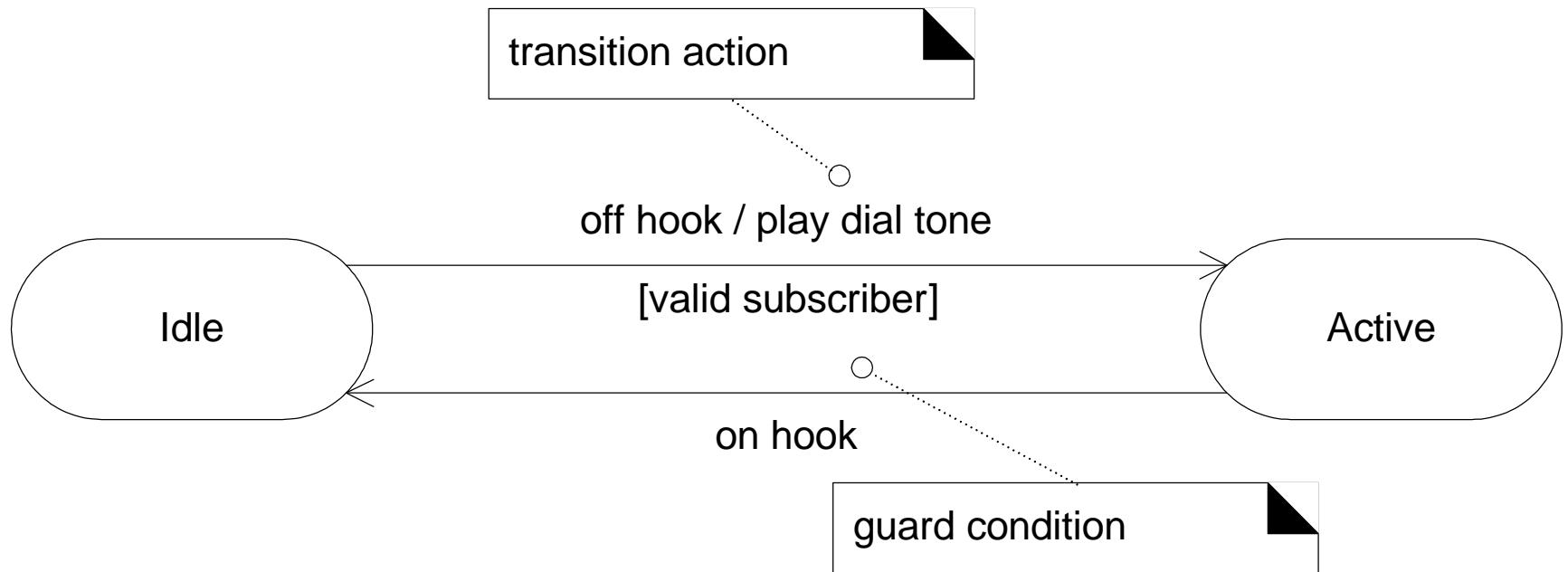
Rappel: l'événement correspond parfois à un appel de méthode (comme ici), mais parfois à un concept abstrait (comme à la page précédente)



Événements

- Différents types d'événements
 - Événement appel (page précédente)
 - Causé par la réception d'un appel d'opération (méthode)
 - Événement modification
 - Exprimé par une expression booléenne et émis dès qu'une expression donnée passe de faux à vrai. L'événement persiste même si l'expression repasse à faux
 - Événement temporel
 - Causé par l'expiration d'un délai
 - Événement signal
 - Causé par la réception d'un signal asynchrone venant d'un autre sous-système

Condition de garde et Action



Condition de garde

- C'est une **expression booléenne** placée entre crochets sur une transition
- Elle doit être valide pour que la transition se produise
- Si elle est combinée à un événement **les deux** doivent être vérifiés pour que la transition ait lieu.
- Trois exemples de conditions de garde:

[t = 15 sec]

[nombre de factures = 20]

retrait(montant) [solde >= montant]

Conditions seules

Avec événement



Action sur une transition

- Elle consiste à une **action procédurale** qui s'exécute lors de la transition.
- Il peut y avoir **plusieurs actions** lors d'une transition mais il faut les séparer par un « / »
- Exemples de transitions avec actions:

Augmente () / $n := n + 1$ / $m := m + 1$

additionne (n) / $\text{somme} := \text{somme} + n$

/clignote

Avec
événement

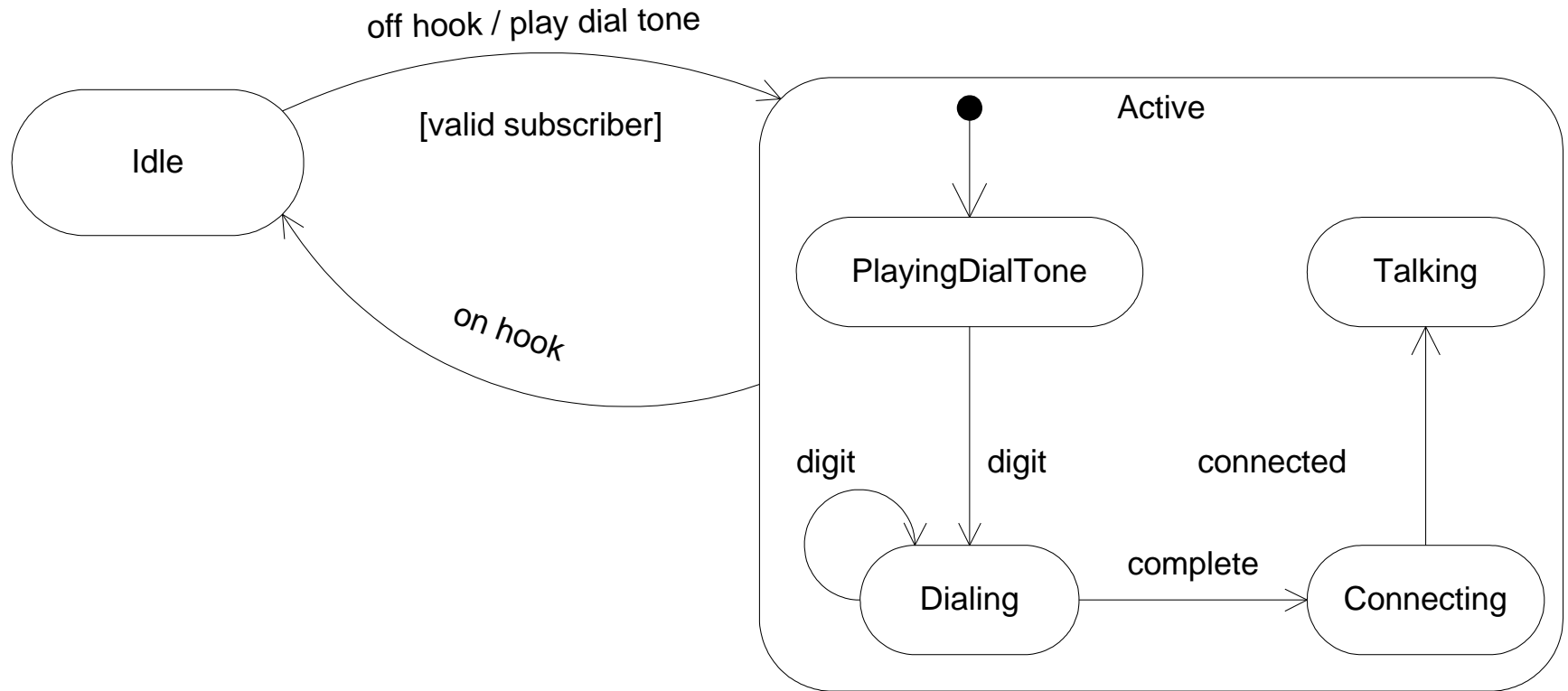
Sans



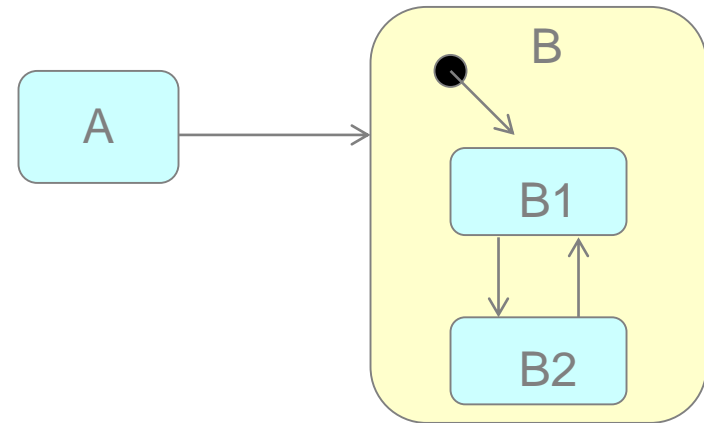
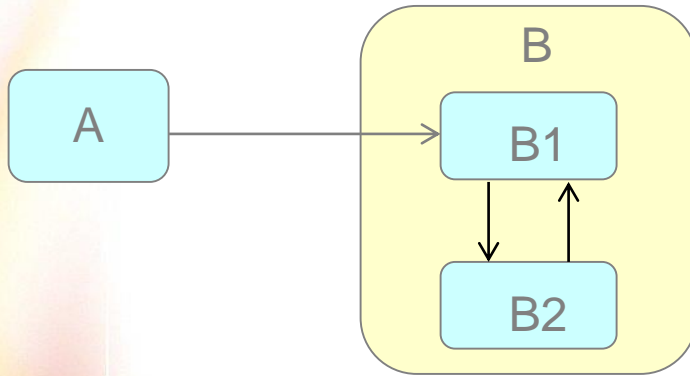
États composites

- Le diagramme d'états peut devenir illisible lorsque le nombre de connexions entre les états devient élevé
- **Solution** : Utilisation des **états composites**
- Un état composite (ou englobant) en est un qui est décomposé en sous-états. Ces sous-états peuvent à leur tour être composites. Cette démarche permet une décomposition hiérarchique du diagramme
- Les sous-états sont soit **concurrents**, soit **disjoints** et mutuellement exclusifs
- Un état non décomposable en sous-états est dit **simple**

États composites / imbriqués - Téléphone

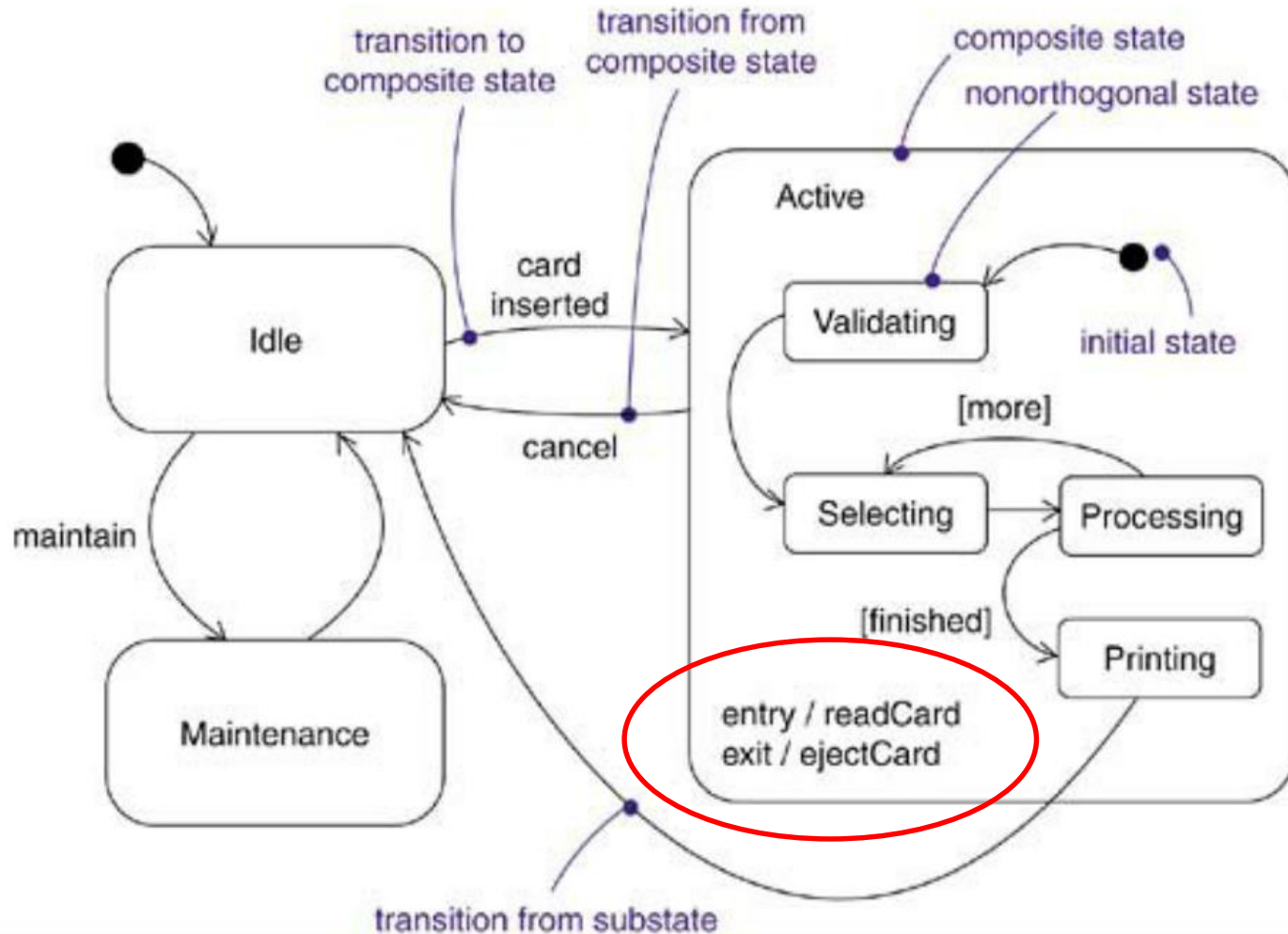


États composites – Représentations alternatives



Question: quels sont les avantages de l'une et de l'autre?

Actions sur entrée et sorties d'un état

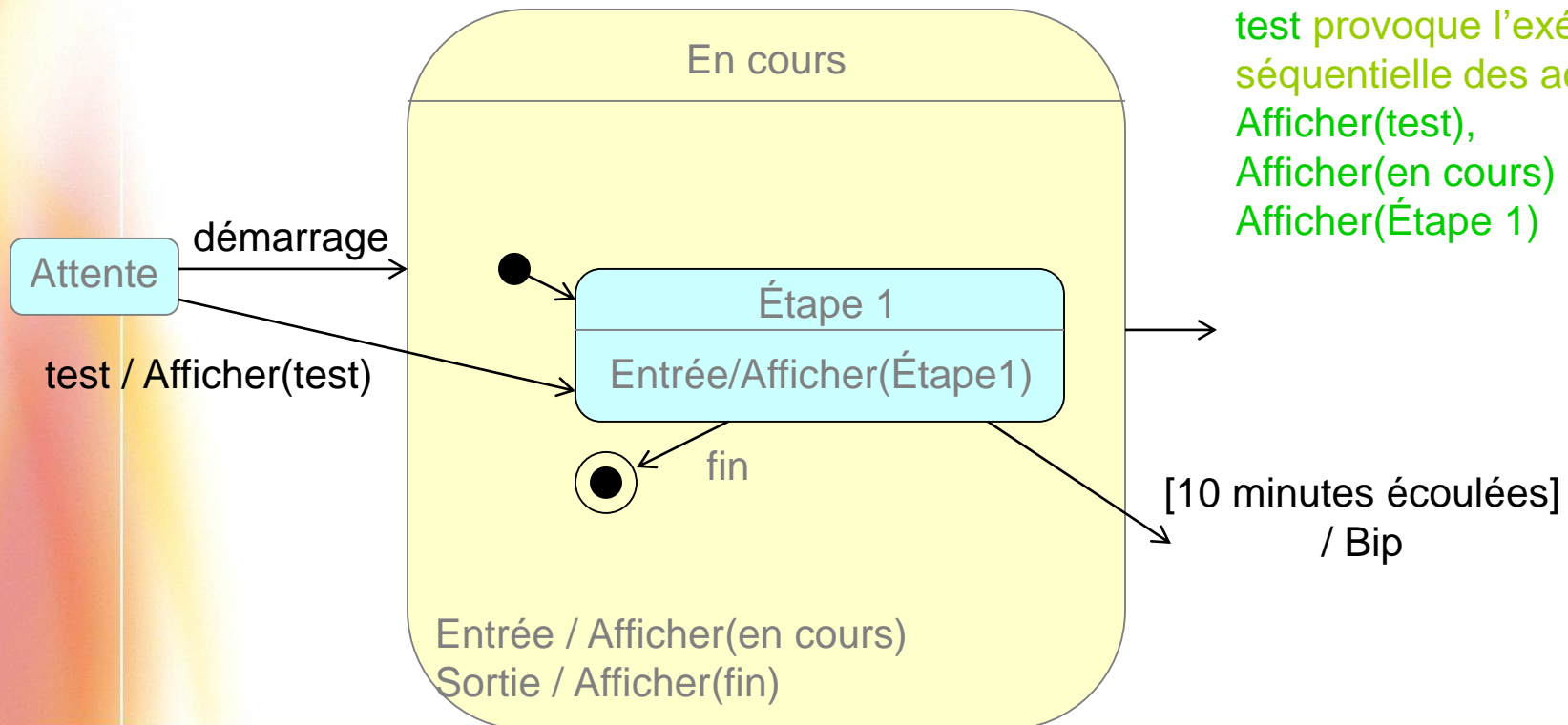


États composites

- Transitions entre états composites

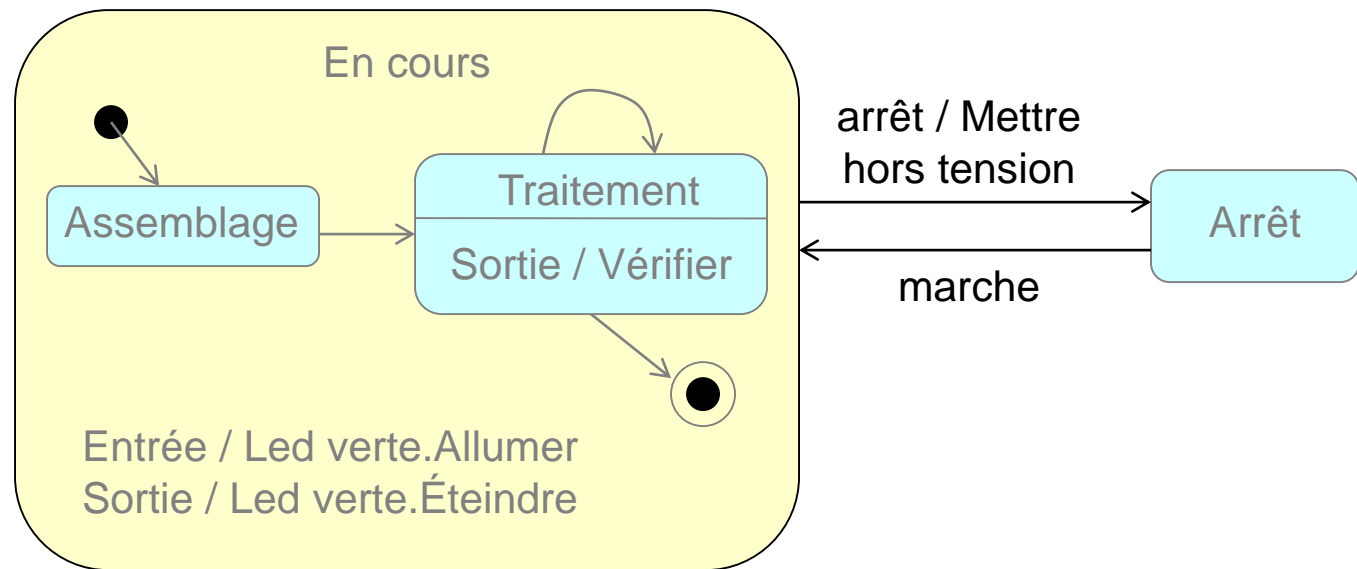
- Les actions d'entrée des états sont toujours effectuées de manière séquentielle dans l'ordre d'accès (du niveau hiérarchique le plus élevé au plus bas)

Lorsque l'automate est dans l'état Attente, l'événement test provoque l'exécution séquentielle des actions Afficher(test), Afficher(en cours) et Afficher(Étape 1)



États composites

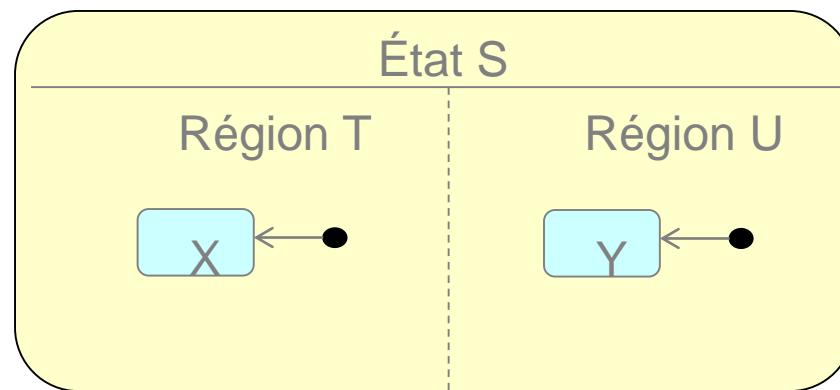
- Transitions entre états composites
 - Exemple



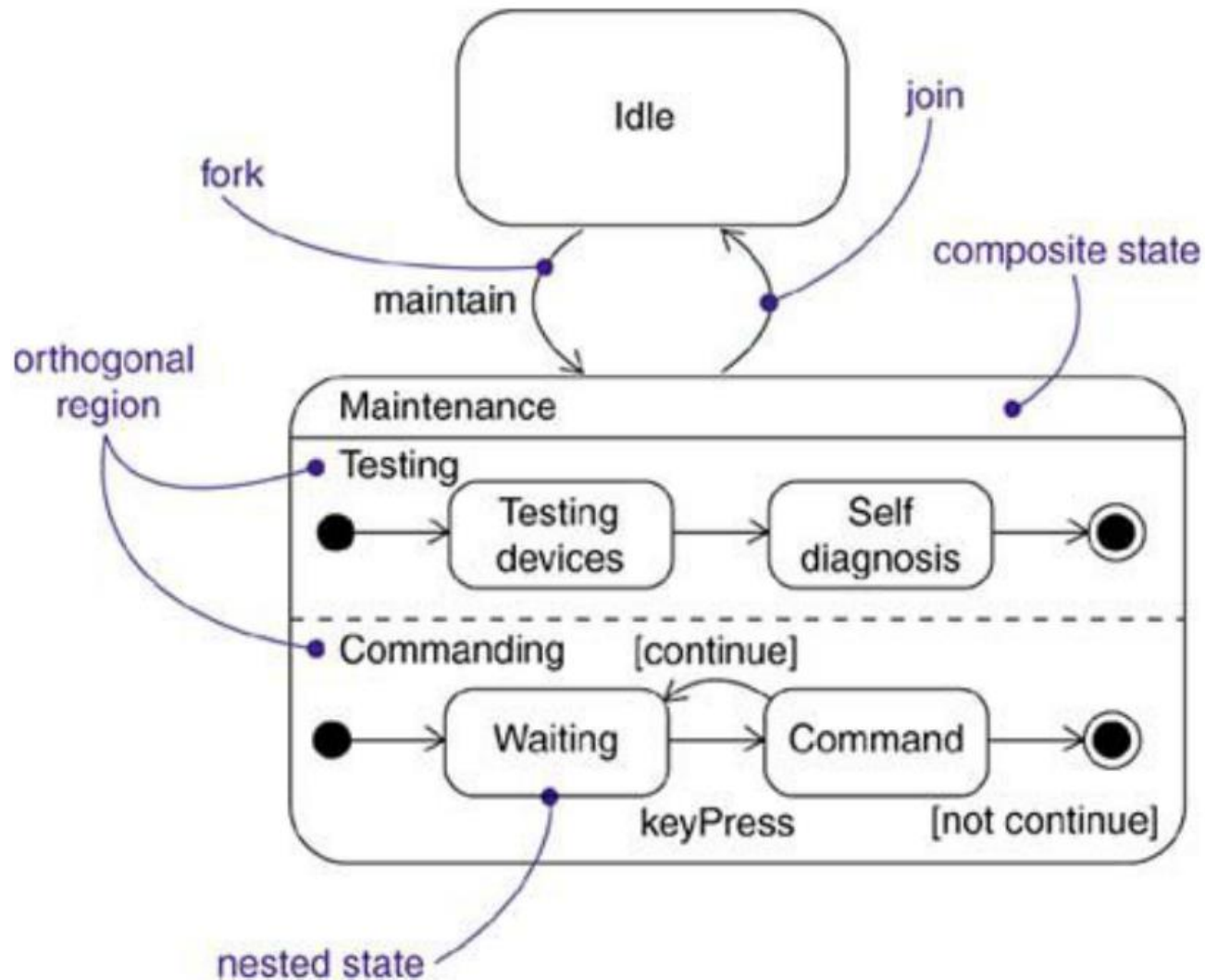
Si l'automate est dans l'état **Traitement**, l'événement **arrêt** force l'exécution de l'action **Vérifier** puis **Led verte.Éteindre** et enfin **Mettre hors tension**

États concurrents

- Un état peut être composé de plusieurs sous-états **concurrents**. Ces sous-états sont alors appelés **régions**
- Représente une forme de parallélisme
- La séparation entre les différentes régions se fait par des lignes en pointillées (horizontales ou verticales)
- Il est possible d'attribuer un nom à une région donnée (optionnel)
- Le déclenchement d'une transition vers l'état composite entraîne l'activation de tous les états initiaux des diverses régions



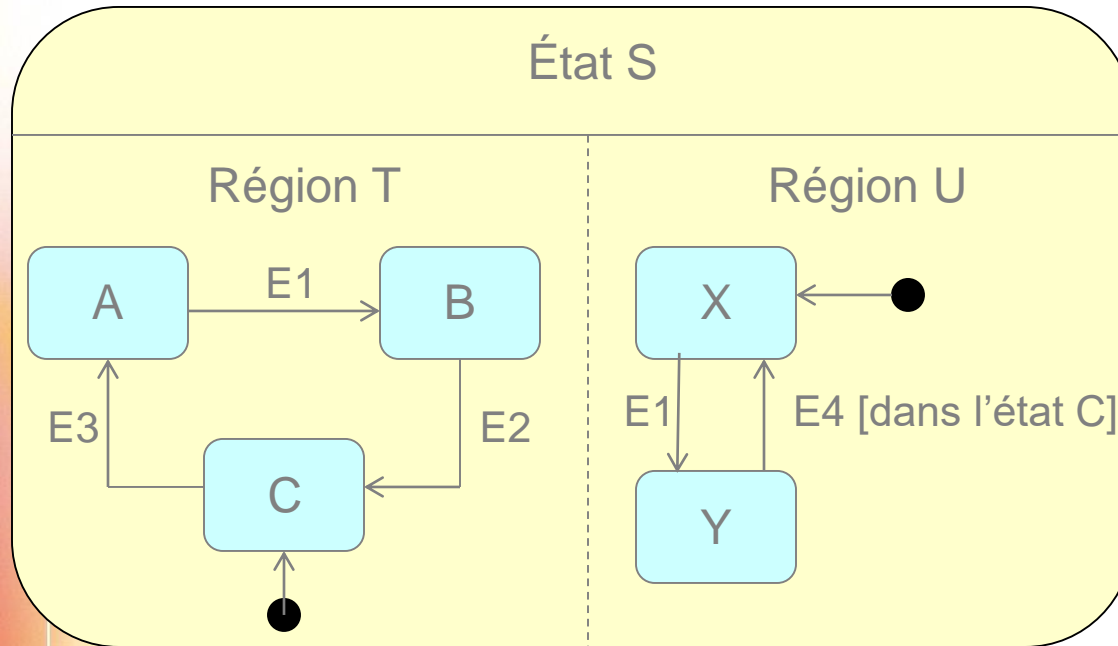
Cet objet mène une double vie !



États concurrents

- États concurrents

- Exemple



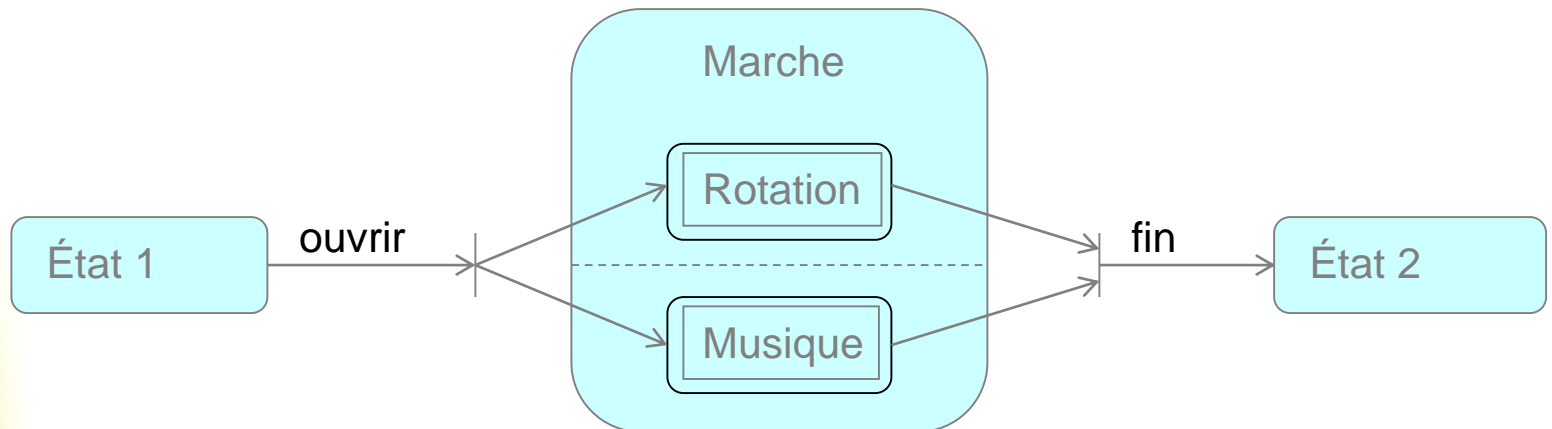
Si un événement mène à **S**, **T** et **U** seront simultanément activées. L'objet est alors placé dans l'état composite **(C,X)**

Si l'événement **E3** arrive, **T** et **U** évoluent indépendamment. L'objet passe alors à l'état composite **(A,X)**

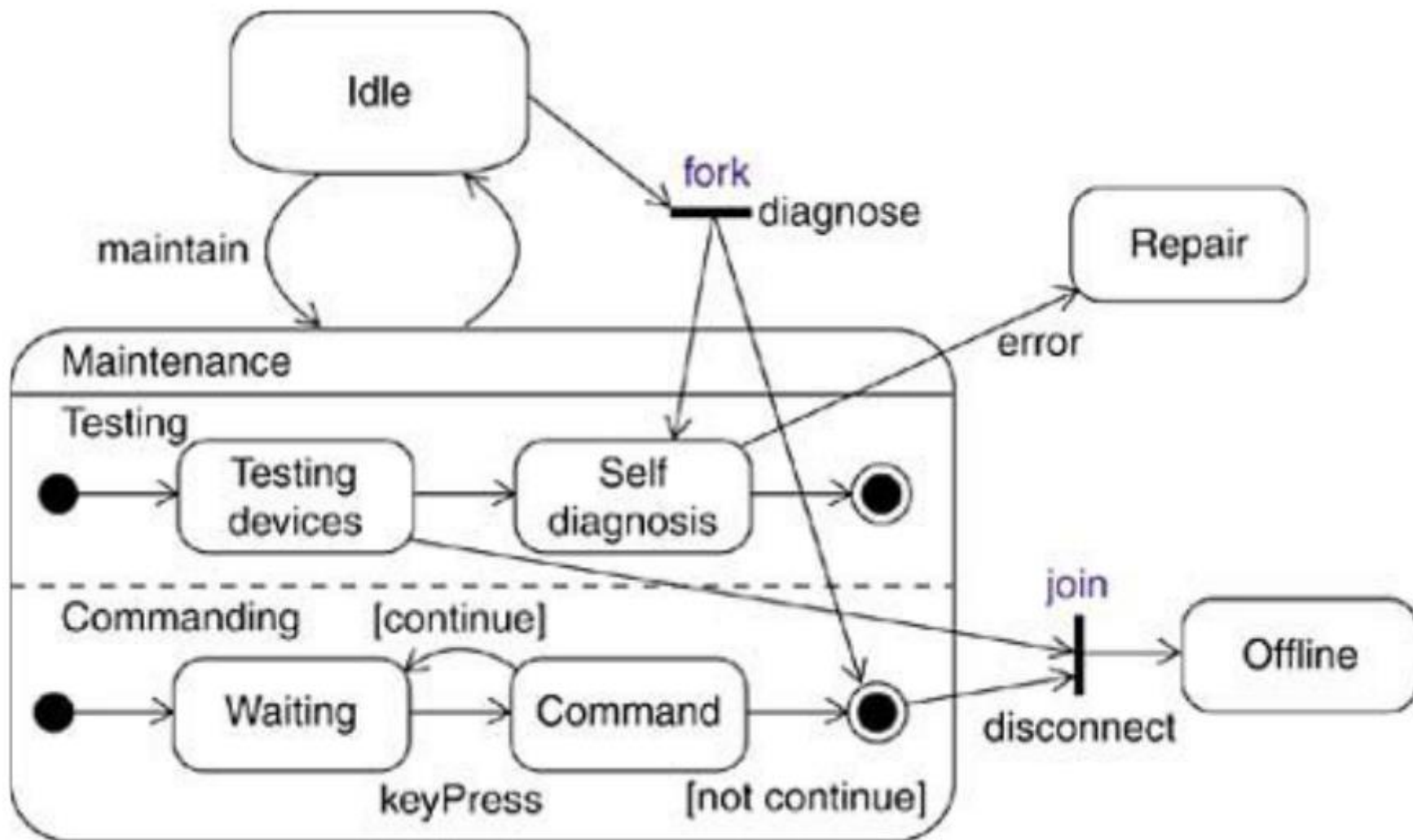
T et **U** évoluent simultanément lorsque l'objet reçoit l'événement **E1** alors qu'il est à l'état **(A,X)** auquel cas il passe à **(B,Y)**

Transitions entre états concurrents

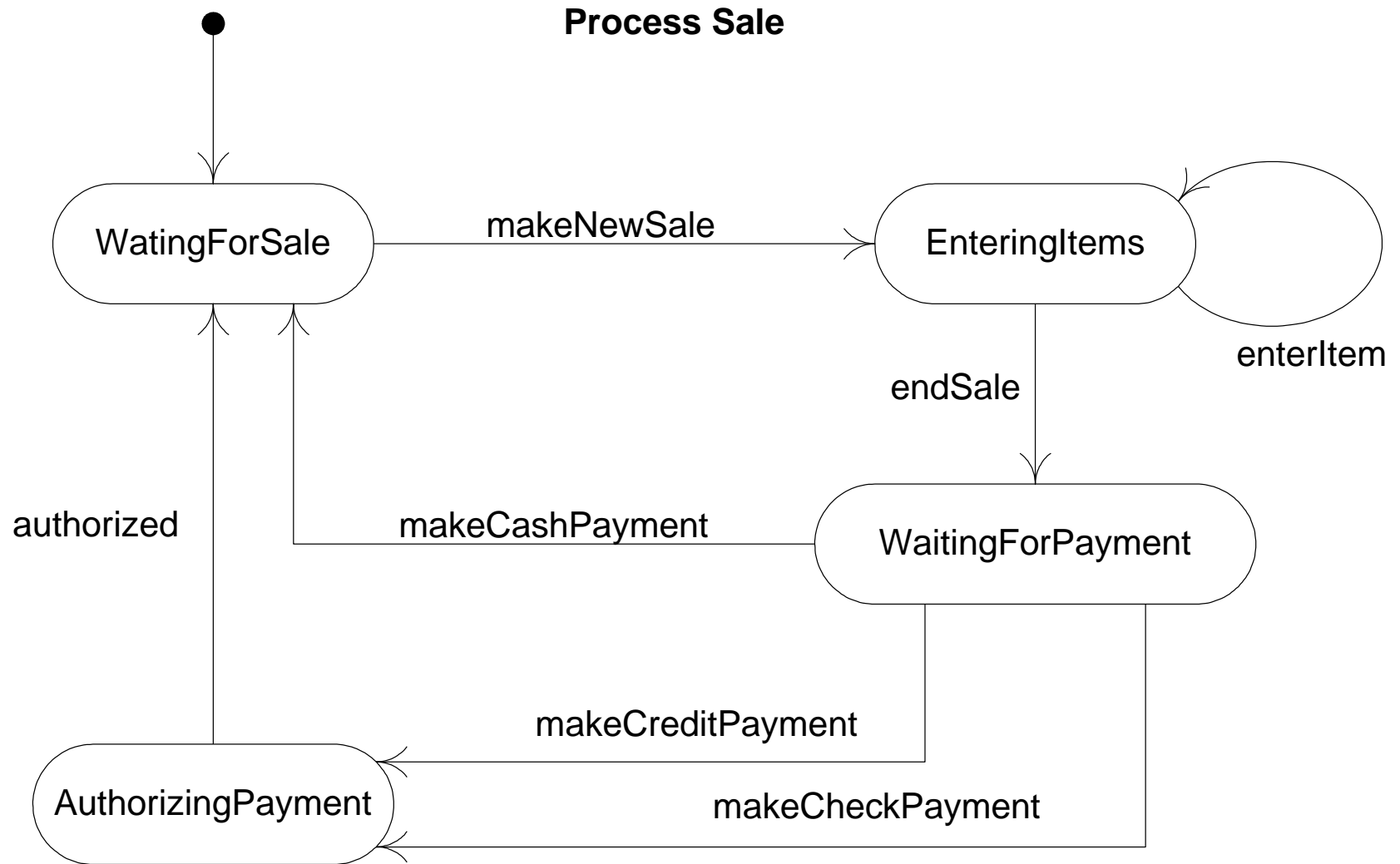
- La transition concurrente est déclenchée lorsque tous les états sources sont atteints et entraîne ainsi l'activation de tous les états cibles
- Exemple: Une table tournante...



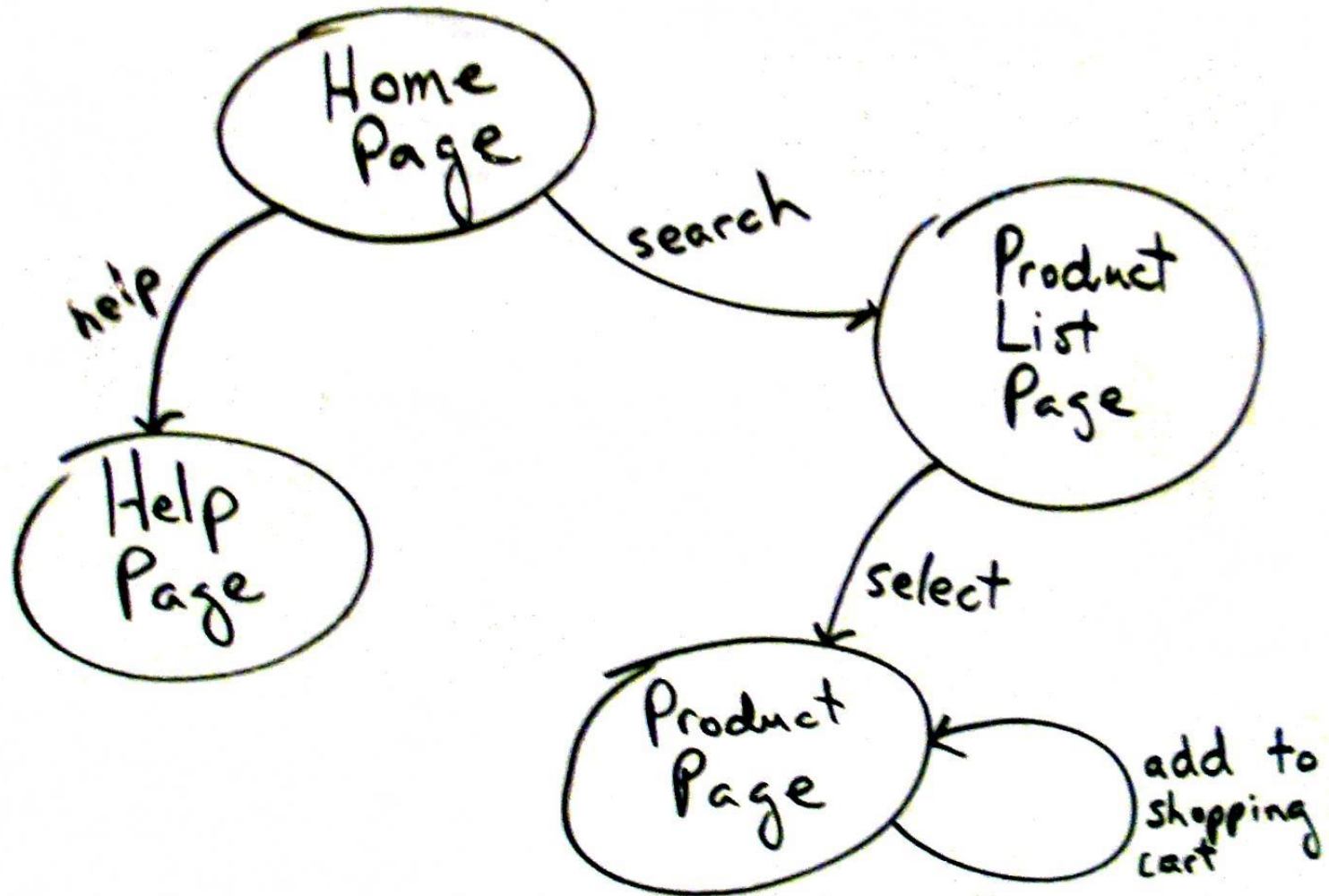
Autre exemple



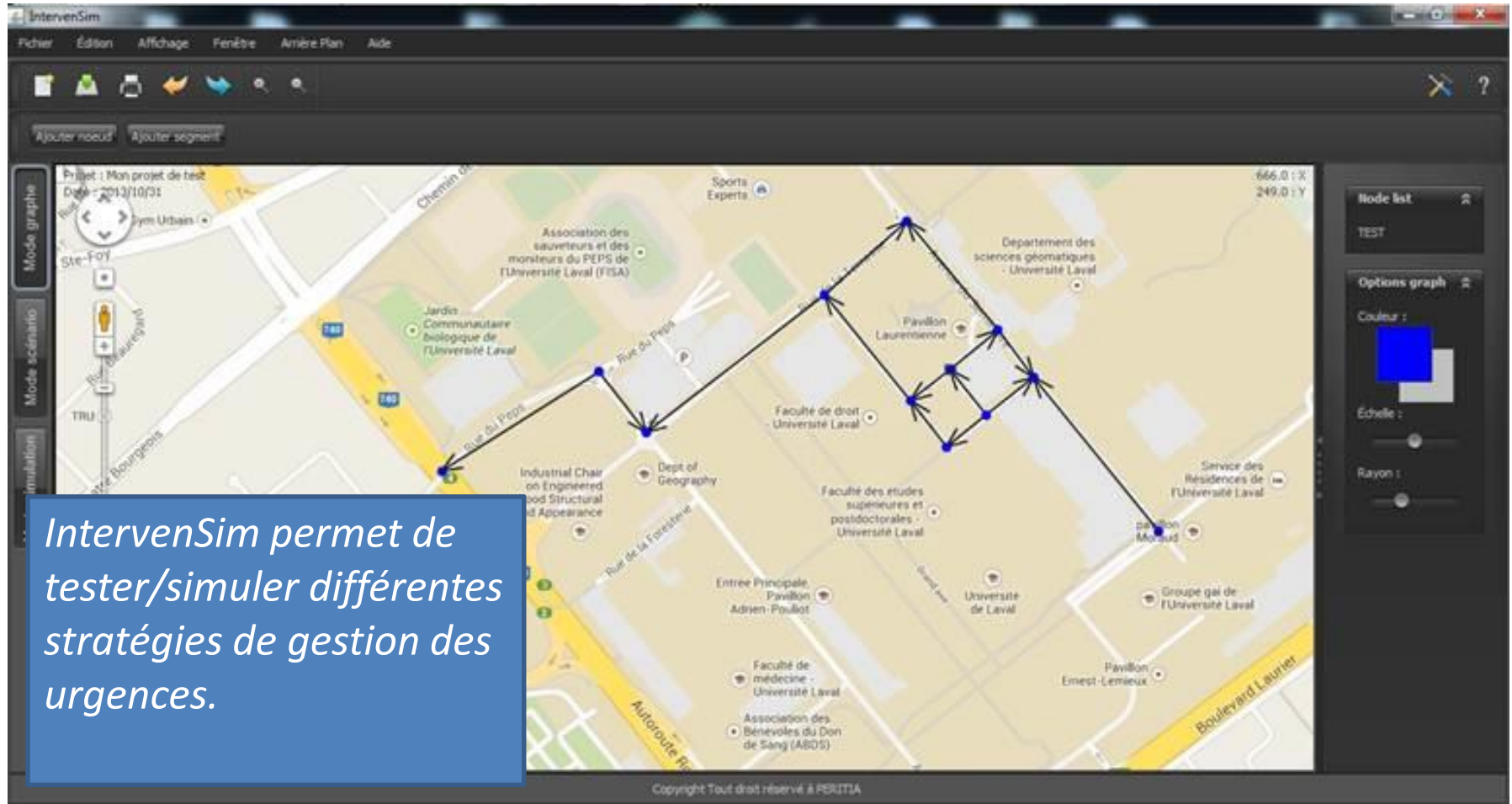
Exemple – NexGen POS



Exemple – Achats sur le Web



Exemple – Cycle de vie d'un véhicule (IntervenSim)



Prix Yves-Roy (A2013): Pascal Dumoulin, Jean-Philippe Lapierre, Guillaume Lorquet et Martin Savoie

Exemple – Cycle de vie d'un véhicule (IntervenSim)

- Un itinéraire de véhicule est une séquence ordonnées de noeuds d'un point de départ vers une destination finale.
- Cycle de vie d'un véhicule d'urgence:
 - Lorsqu'il n'y a pas d'urgence le véhicule est en attente au noeud courant.
 - Lorsqu'il y a un appel:
 - le véhicule obtient l'itinéraire vers la destination finale;
 - le véhicule se rend alors sur les lieux en suivant chacune des étapes de l'itinéraire;
 - une fois sur place, il traite l'urgence;
 - une fois l'urgence traitée, il est prêt à répondre à un autre appel.

Exemple – Cycle de vie d'un véhicule (IntervenSim)

- Considérez qu'un objet *itineraire* est obtenu par la méthode *getItineraire(destinationFinale)*.
- Un objet *itineraire* est un itérateur:
 - *itineraire.premiereEtape* peut nous permettre d'obtenir la première destination dans la séquence de noeuds.
 - *itineraire.prochaineEtape* nous permet de passer à la prochaine étape à partir du noeud courant.

Exemple – Cycle de vie d'un véhicule (IntervenSim)

Visual Paradigm for UML Standard Edition (Université Laval)

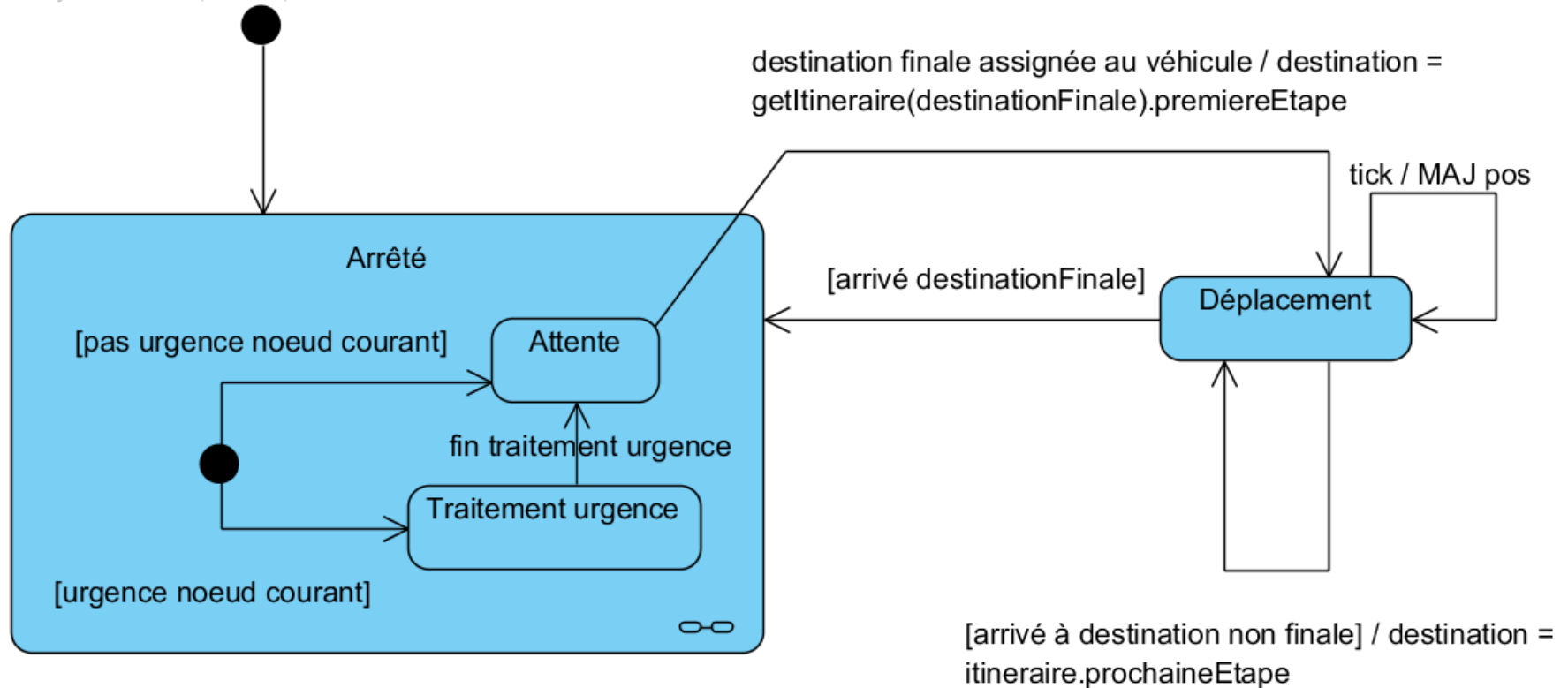
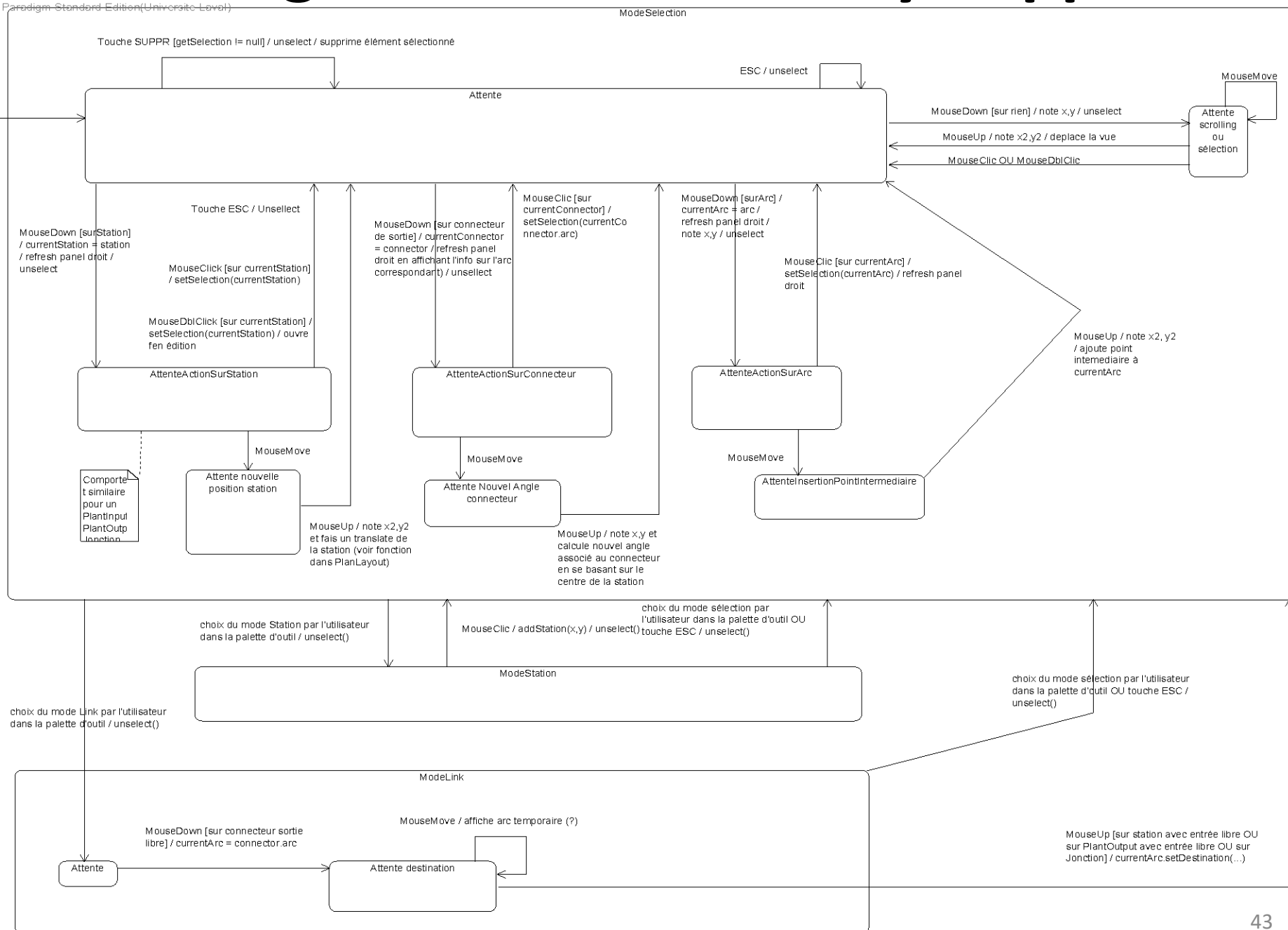


Diagramme d'états RecyclApp

Visual Paradigm Standard Edition (University Level)

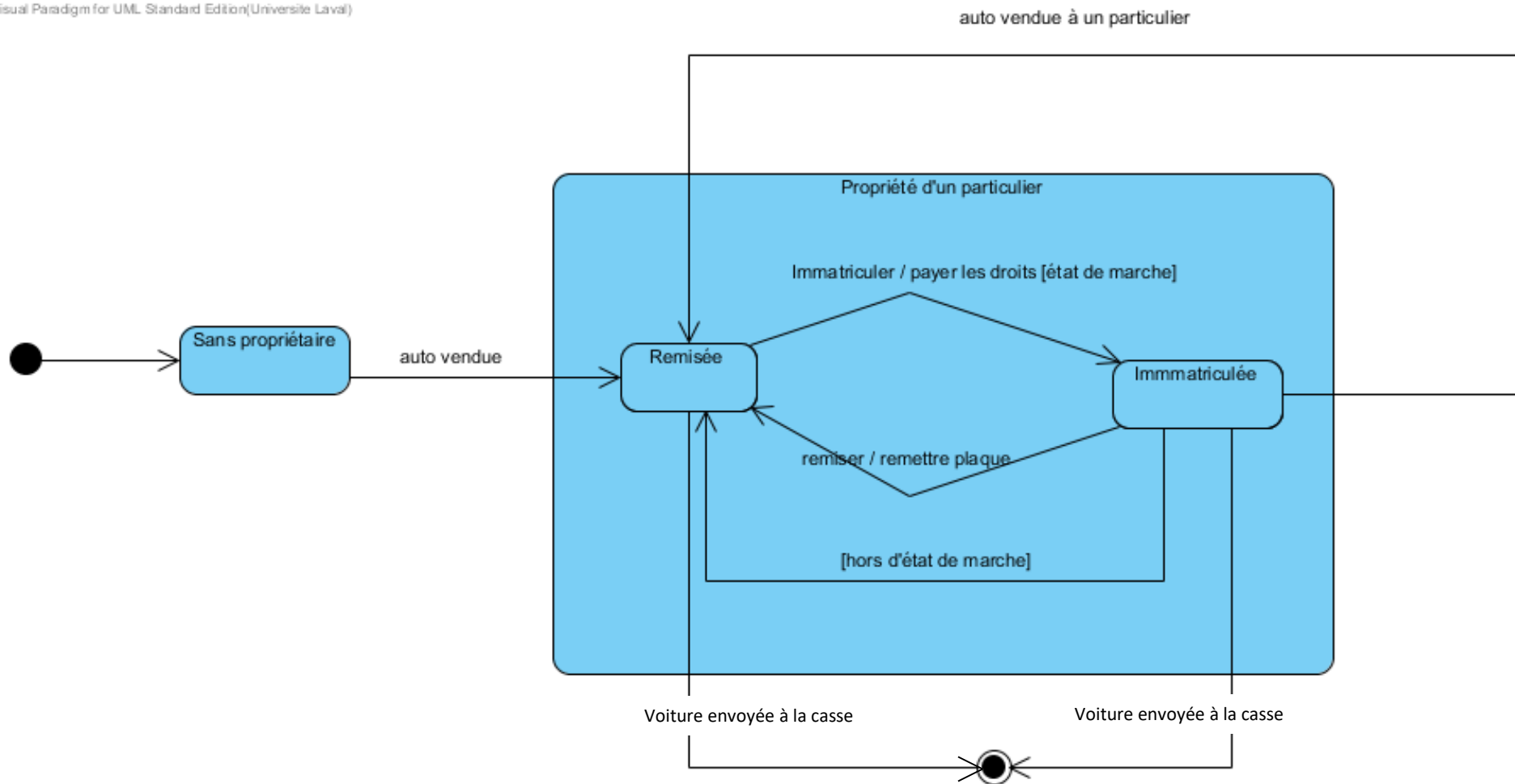


Exercice – Diagramme d'états d'une voiture

- Une voiture peut ne pas encore être la propriété d'un individu (ex : elle se trouve chez le concessionnaire en attendant d'être vendue). Lorsqu'elle est vendue elle devient la propriété d'un particulier.
- Le véhicule ne pourra pas rouler sur les routes tant qu'il ne sera pas immatriculé par son propriétaire (en attendant on dit que le véhicule est remisé). Pour immatriculer le véhicule, il doit être en état de marche. On paie aussi des frais d'immatriculation au moment de l'immatriculation.
- Le propriétaire peut aussi décider de remiser son véhicule (il doit par le fait même remettre la plaque d'immatriculation).
- Un véhicule qui échoue une inspection mécanique peut aussi se voir l'obligation d'être remisé.
- Un véhicule peut être vendu à un autre particulier (il change de propriétaire).
- Finalement, en fin de vie utile un véhicule est recyclé
- Tracer un diagramme d'état :
 - Pistes : les différents états sont soulignés dans le texte (mais vous pouvez changer les noms); des états peuvent être imbriqués dans d'autres états.
 - Identifier les transitions (en y associant les événements, conditions et actions associées
 - Ne pas oublier d'identifier l'état initial et l'état final.

Solution – Diagramme d'états d'une voiture

Visual Paradigm for UML Standard Edition (Université Laval)



Exercise – Design 3 (Robot)

Et dans le projet de session?

- Interface utilisateur (édition diagramme, mode arc, nœud, etc.. Et ce qui est permis dans chaque cas)
- Arcs et éléments
- Transit de l'entité (?)
- Lié à la fonctionnalité de l'optimisation automatique temps-réel.
 - Deux sous-réseaux... est-ce valide...
 - Détection des boucles?

À faire pour ce module

- Lecture des chapitres 29 (version anglaise) ou 25.8 à 25.15 (version française)
- Référence complémentaire sur les diagrammes d'activités et diagrammes d'état: chapitres 20 et 25 de [Booch 2005] (disponible en version électronique à la bibliothèque)
- Comprendre les concepts suivants:
 - Diagramme d'états (état; événement; transition; condition de garde, ...)

