

DIAGRAMME D'ETATS- TRANSITIONS

ELAN

202 avenue de Colmar - 67100 STRASBOURG

☎ 03 88 30 78 30 📠 03 88 28 30 69

✉ elan@elan-formation.fr

www.elan-formation.fr

SAS ELAN au capital de 37 000 € -

RCS Strasbourg B 390758241 – SIRET 39075824100041 – Code APE : 8559A

N° déclaration DRTEFP 42670182967 - Cet enregistrement ne vaut pas agrément de l'Etat

SOMMAIRE

I.	<i>Théorie</i>	3
1.	Principe.....	3
2.	Définition.....	3
II.	<i>Notation graphique des éléments</i>	4
1.	Etats, transitions et évènements.....	4
2.	Etat initial et états finaux.....	5
3.	Etat composite, super-état et historicité	5
a.	Utilisation	5
b.	Exemple d'historicité.....	6
4.	Transitions spéciales.....	6
5.	Les souches	7
III.	<i>Cas concret</i>	8
1.	Diagramme d'états-transitions d'un lecteur DVD.....	8
2.	Fonctionnement en détail.....	9
IV.	<i>Quelques subtilités</i>	10
1.	Les évènements réflexifs	10
2.	Etats concurrents	10
3.	Evènement paramétré.....	11
V.	<i>Pour aller plus loin</i>	11

I. Théorie

1. Principe

Dans le cadre du processus de fonctionnement d'un système (machine, application...), celui-ci est programmé pour exécuter un certain nombre d'actions dans un ordre donné.

Durant ce processus, le système passe par un certain nombre d'états (allumé, éteint, en traitement...) et le passage d'un état à un autre s'effectue de multiples manières : modification d'une valeur, activation d'une fonction, choix effectué par l'utilisateur, résultat d'une condition...

Chaque "passage" représente une transition entre deux états du système. Tout ceci constitue le principe d'un **diagramme d'états-transitions**.

2. Définition

Les diagrammes d'états-transitions permettent de décrire les comportements d'un objet (dans le sens de la Programmation Orientée Objet (POO)) ou d'un composant (une partie du système ou un système au complet), dans le contexte d'un scénario où interviennent d'autres objets/composants ou l'utilisateur lui-même. Un graphe d'états-transitions s'appelle également un **automate**.

Dans le cadre de la réflexion du développement d'un projet, un diagramme d'états-transitions (abrégé "DET" pour la suite de ce support) a pour fonction principale de représenter un schéma d'actions fonctionnel qui orientera le développement du système vers une plus grande exhaustivité.

En général, les DET permettent de caractériser des éléments qui pourraient avoir échappé aux développeurs :

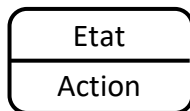
- ✚ **Décrire un processus** (type workflow) pour matérialiser efficacement des classes d'objet sans omettre une fonctionnalité primordiale du système
- ✚ **Aider à concevoir les activités, les IHM** (un état peut, si besoin, se présenter visuellement, et par là entraîner un écran à l'utilisateur)
- ✚ Aider à construire les **scénarii de test**
- ✚ **Déterminer les états** pour gérer les risques de dysfonctionnement du système

II. Notation graphique des éléments

1. Etats, transitions et évènements







Un état se caractérise par sa durée et sa stabilité, il représente une conjonction instantanée des valeurs des attributs d'un objet.



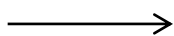
L'état atteint par un élément du système peut entraîner une action (une méthode d'objet, un traitement particulier...). L'action se décrit dans l'état, très brièvement, elle n'est inscrite qu'à des fins de précision.

Les actions peuvent se décomposer en fonction de ce qui devra les déclencher au sein de l'état. Un déclencheur peut être :

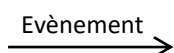
-  **Entry** : à l'entrée dans l'état
-  **Do** : action dans l'état, sans condition d'exécution
-  **Exit** : à la sortie
-  **On event** : à l'arrivée d'un sous-événement interne à l'état

Attention, les actions attachées aux clauses "entry" et "exit" **ne sont pas exécutées** si l'évènement spécifié dans la clause "on" survient.

Les états se succèdent par l'intermédiaire de transitions :



Une **transition** représente le passage instantané d'un état vers un autre.



Une transition peut être déclenchée par un **évènement**. En d'autres termes : c'est l'arrivée d'un évènement qui conditionne la transition. Ainsi, l'évènement déclenche le passage entre deux états, donc la transition associée.

2. Etat initial et états finaux



L'**état initial** est le point de départ du scénario couvert par le DET.



L'**état final** indique la fin du scénario couvert par le DET.

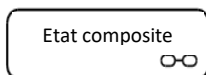


Le **point de connexion** représente un sous-ensemble d'états-transitions non-couvert par le DET. Il indique bien souvent qu'un autre scénario ou processus d'actions se déclenchent à l'arrivée sur ce point de connexion. Utile pour, encore une fois, ne pas surcharger le DET.

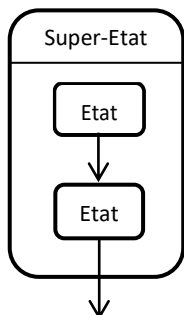
3. Etat composite, super-état et historicité

a. Utilisation

Des états particuliers peuvent également intervenir dans le DET pour aider à la compréhension du schéma, ou tout simplement symboliser un élément qui serait trop complexe à représenter entièrement, ce qui alourdirait le DET.

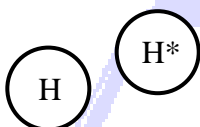


Un état peut également représenter tout un ensemble d'états-transitions sans pour autant devoir le déployer au complet. L'**état composite** sert alors à rassembler une sous-partie implicite du système dont le processus est décrit dans un autre DET.



Un **super-état** regroupe un ensemble d'états et de transitions susceptible d'être sollicité au complet par d'autres événements du scénario.

Par exemple, allumer un ordinateur entraîne de nombreux états successifs à celui-ci (boot, chargement des drivers, exécution de l'OS, chargement du compte utilisateur...). Cet ensemble d'état peut s'englober dans un super-état qu'on pourrait nommer "démarrage du PC".



Un **état historique** est un pseudo-état qui mémorise le dernier sous-état actif d'un super-état ou d'un état composite, pour y revenir directement ultérieurement. Il en existe deux :



l'état historique **plat** (noté H)



l'état historique **profond** (noté H*)

Un état historique profond reprend le sous-état actif au moment même de son interruption. L'état historique plat, lui, reprend le sous-état actif du début, sans mémorisation "profonde" de son état lors de l'interruption.

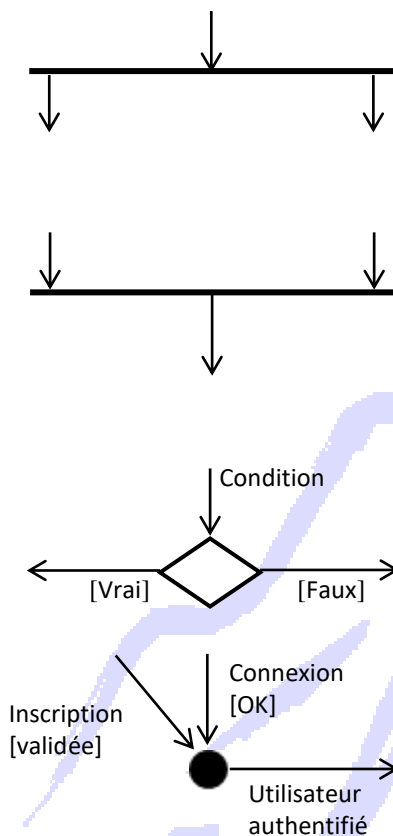
b. Exemple d'historicité

Un exemple pour illustrer l'utilisation des états historiques : lorsque vous visionnez un DVD ou un Blu-Ray décomposé en chapitres, le visionnage du film peut connaître deux états d'interruption particuliers :

- Le bouton pause, qui stoppe le visionnage mais le reprendra au moment même où il a été interrompu (état historique profond).
- Le bouton "chapitre précédent", qui reprend le chapitre du début mais n'aura pas mémorisé le moment exact où le film a été interrompu (état historique plat).

4. Transitions spéciales

Les transitions d'un état à l'autre ne sont pas forcément linéaires : un état peut survenir à la suite d'une condition, ou un évènement peut déclencher deux états s'exécutant en simultané. Ces spécificités se présentent ainsi :



Une transition déclenchant deux états simultanés donne lieu à une **barre de synchronisation (fourche, ou fork en anglais)**. Elle exprime l'obligation pour le système d'enclencher deux états obligatoirement (il faut penser que "ceci" entraîne "ça" **ET** "ça")

De la même manière, deux états exécutés donnant lieu simultanément à un même résultat se présentent avec une **fourche inverse**. Le système oblige l'accomplissement des deux transitions initiales pour atteindre l'état suivant (ici, il faut penser que "ceci" **ET** "cela" entraînent "ça")

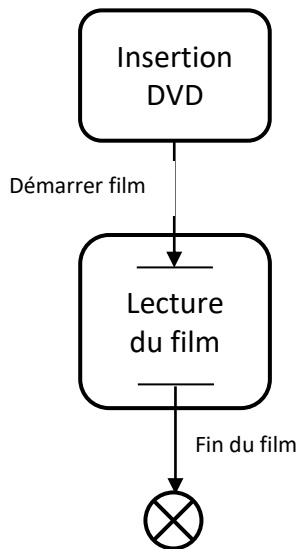
Un évènement peut donner lieu à deux états différents via un **point de décision** (le principe de "Si...Alors...Sinon..."). Les deux états qui en découlent se déclencheront selon le résultat de la transition de départ. Les **arcs sortants** sont appelés des **gardes**, ils se notent entre crochets.

Un **point de jonction** peut être utilisé pour factoriser le résultat de transitions ayant soit la même finalité, soit plusieurs issues. Cette représentation est purement graphique et permet de simplifier l'enchaînement d'évènements ne nécessitant pas d'état intermédiaire.

Attention : les transitions entrantes ou sortantes d'un point de jonction sont forcément évaluées à "vrai".

En d'autres termes, dans l'exemple ci-dessus, seules une connexion réussie (login et mot de passe correspondants) **OU** une inscription validée selon les règles du système déclencheront l'authentification de l'utilisateur en question, et donc donnera lieu à l'état "Bienvenue, cher Monsieur X" d'un site internet, par exemple.

5. Les souches



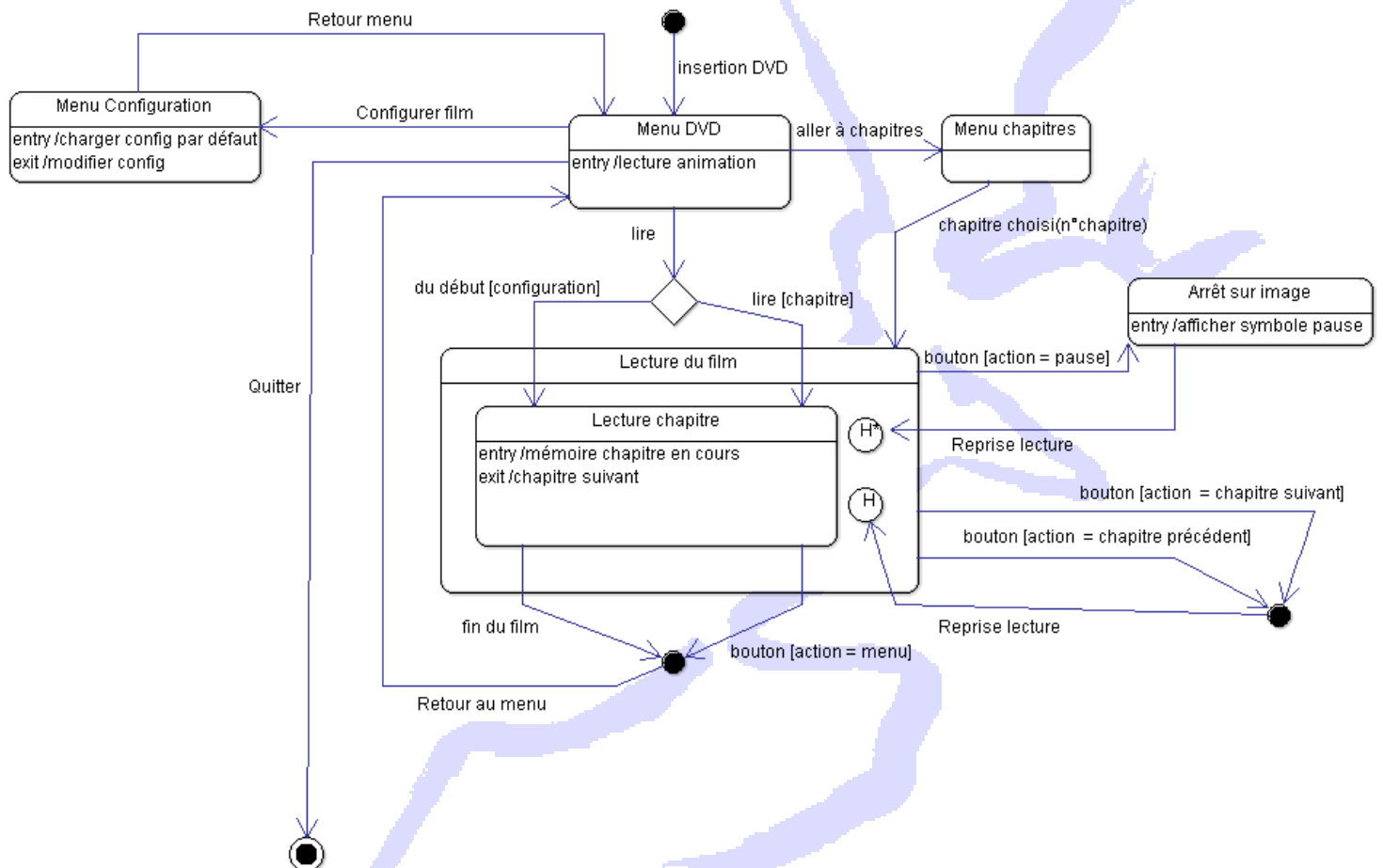
Dans le même souci d'alléger un diagramme qui ne nécessite pas le détail d'une partie de son fonctionnement, il est possible de représenter un ensemble de sous-états atteints ou déclencheurs d'une transition précise par **une souche**.

Dans l'exemple ci-contre, la lecture d'un film en DVD comporte logiquement bien plus d'informations et d'actions pouvant être représentées. Mais si le système à schématiser ne nécessite pas le détail de cet état, si le but du DET est d'illustrer le système sans nécessiter la présence de certains sous-états, alors l'utilisation d'une souche est vivement conseillée.

III. Cas concret

1. Diagramme d'états-transitions d'un lecteur DVD

Le diagramme d'états-transitions ci-dessous présente les différents états par lesquels passe un lecteur DVD dans la prise en charge d'un film à visionner.



2. Fonctionnement en détail

Ce qui suit développe le processus du graphe d'états-transitions du lecteur DVD progressivement :

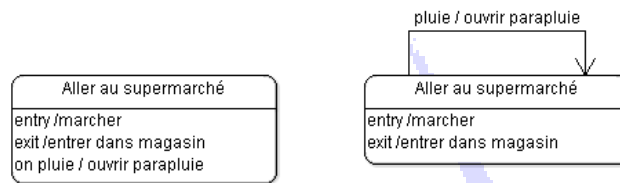
- ✚ L'état initial représente le lecteur allumé avant insertion d'un disque
- ✚ Une fois le disque inséré, le menu du DVD s'affiche à l'écran et attend un choix du spectateur :
 - Soit il désire configurer le film (changer la langue, les sous-titres, etc.)
 - Soit il souhaite aller à un chapitre précis
 - Soit il lance la lecture du film du début (avec prise en charge de sa configuration, si elle a eu lieu)
- ✚ Quels que soient les choix du spectateur, les transitions amènent vers le super-état "Lecture du film", se faisant chapitre par chapitre selon qu'il ait démarré du début ou depuis un chapitre précis.
- ✚ La lecture peut s'interrompre de plusieurs manières :
 - Le spectateur appuie sur le bouton chapitre précédent ou suivant de sa télécommande, ce qui entraîne un état historique plat. Ces deux transitions débouchant sur la même idée de reprise du film, elles sont associées via un point de jonction.
 - Le bouton pause a été enclenché, ce qui entraîne l'état "Arrêt sur image" et ne reprendra qu'à la reprise du film (ce qui impose un état historique profond)
 - Enfin, la fin normale du film ou le bouton Menu de la télécommande occasionnent, via un autre point de jonction, un retour à l'état "Menu DVD".
- ✚ Pour finir, quitter le film depuis le menu du DVD entraîne bien évidemment un état final du système.

Il faut noter qu'il s'agit d'une **possibilité de représentation** de ce processus et qu'elle ne constitue en rien la seule manière de le schématiser.

Il est bien évident qu'une ou plusieurs parties du cheminement du diagramme pourraient être plus explicites ou que certaines actions sont manquantes ou trop simplifiées, le but étant, dans le cadre de ce support, d'illustrer un maximum des éléments décrits précédemment.

IV. Quelques subtilités

1. Les évènements réflexifs



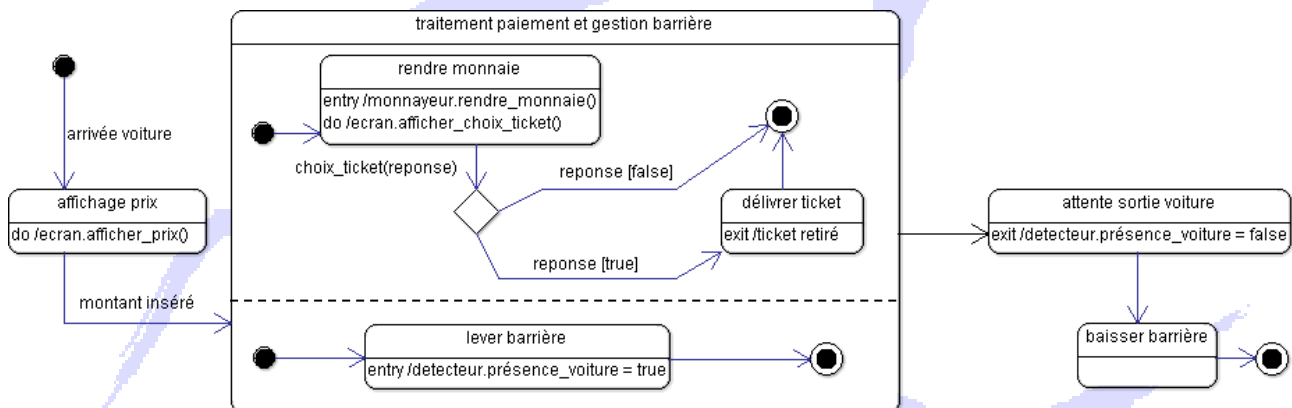
Les diagrammes ci-dessus représentent, au premier coup d'œil, le même système.

Dans le diagramme de gauche et pour rappel : **les actions attachées aux clauses "entry" et "exit" ne seront pas exécutées si l'évènement "pluie" dans la clause "on" survient**. Il faut donc comprendre que s'il pleut avant le démarrage de l'évènement, l'état ne démarrera pas (autrement dit, on n'ira pas au supermarché s'il pleut...)

Étant donné que sur le chemin du supermarché, il est possible qu'il pleuve par intermittences, donc que le parapluie pourrait être ouvert plusieurs fois, tenant compte également que la marche s'interrompra lors de l'ouverture du parapluie, puis redémarrera ensuite, le diagramme de droite est **plus précis** (voire plus juste). L'évènement "pluie" sur le second diagramme renvoie au même état.

Il est ainsi appelé **évènement réflexif**.

2. Etats concurrents

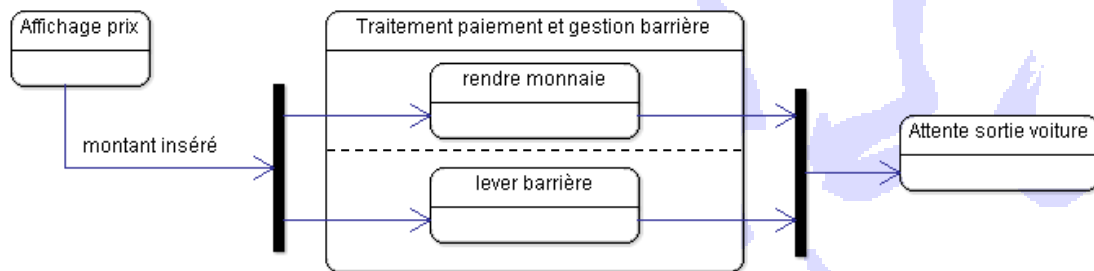


Le diagramme ci-dessus présente l'activité d'une borne de péage à l'arrivée d'un véhicule. Lorsque le conducteur insère le montant correspondant au prix affiché, deux évènements s'exécutent en parallèle : la barrière se lève et le monnayeur rend la monnaie (s'il y en a) et demande si un reçu doit être imprimé.

Le super-état "traitement paiement et gestion barrière" est appelé **état orthogonal** (disposant de plus d'une région de flots d'exécution) et illustre la synchronisation de deux évènements survenant en parallèle. Ces évènements sont dits alors **en concurrence**.

Un trait horizontal en pointillés doit être ajouté à l'état orthogonal pour scinder les flots d'exécution simultanés.

Une autre façon de schématiser l'entrée et la sortie du super-état ci-dessus est d'utiliser des fourches (puisque les états concurrents sont exécutés au même moment et que leurs deux accomplissements uniquement entraînent l'état en sortie). Ci-dessous, une illustration du système simplifié :



3. Évènement paramétré

Le diagramme précédent possède un **évènement paramétré** : choix_ticket(reponse).

Il est tout à fait possible de faire passer des paramètres entre parenthèses dans un évènement pour avoir à l'exploiter dans un état ou le faire passer à travers une partie du système (comme une canette de Cola sur une chaîne de production, par exemple)

V. Pour aller plus loin

<http://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-etats-transitions#L5>

http://pagesperso.lina.univ-nantes.fr/~vailly-a/Enseignement/EMIAGE/Chantier/C352/c5/DET/d350_5_3_2.html