

LE COMPORTEMENT DES SYSTEMES LOGIQUES SEQUENTIELS : Comment décrire un comportement par un graphe d'états ?

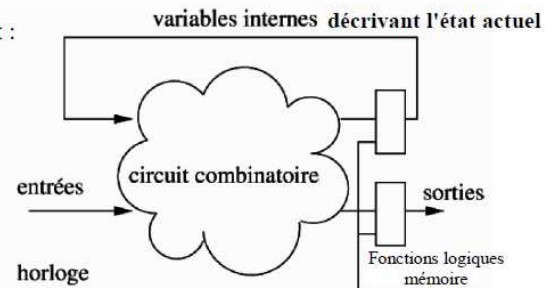
- La synthèse d'un **graphe d'états** permet d'obtenir une **machine à états** réalisant la partie commande d'un système. **Les états représentent toutes les valeurs que peuvent prendre au fil du temps les variables internes du circuit de logique séquentielle.**
- La notion d'évènement intervenant dans ces systèmes pour le passage d'un état à un autre, ils sont également appelés **systèmes à évènements discrets** ("discret" s'oppose à "continu").



- Le schéma de la machine à états générique est le suivant :

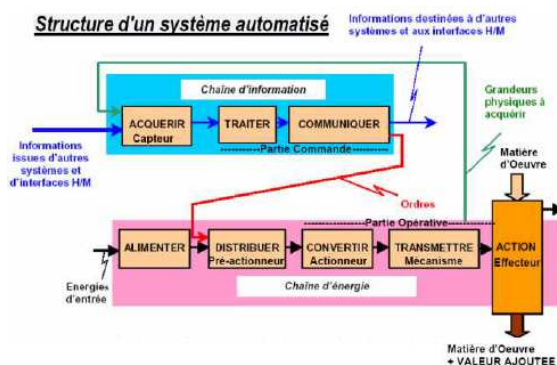
- Par exemple, pour une machine à café, les états peuvent être :

- 1 – Attente de pièce ;
- 2 – Descente du gobelet ;
- 3 – Versement de la poudre à café ;
- 4 – Versement de l'eau chaude ;
- 5 – Affichage "café est prêt".



Ce comportement se déroule selon une **SÉQUENCE**. Un tel système est donc bien **SÉQUENTIEL**.

Structure d'un système automatisé



- Le graphe d'états, comme l'algorithme (cf. cours d'informatique), est essentiellement un outil graphique permettant de **modéliser** le comportement séquentiel, en termes de déroulement d'actions temporelles.

Mais il peut aussi servir à **programmer** les composants réalisant la fonction "Traiter" de la chaîne d'information (microcontrôleur, microprocesseur, automate programmable, ...). Les variables d'entrée de la fonction "Traiter" sont alors les informations fournies par la fonction "Acquérir" (capteurs, ...) et les variables de sortie sont les ordres pour la fonction "Distribuer" de la chaîne d'énergie, éventuellement via la fonction "Communiquer".

Quel que soit l'outil adopté pour modéliser le comportement séquentiel d'un système, il existe souvent plusieurs solutions. La solution la plus simple qui respecte l'ensemble des contraintes est donc à privilégier. C'est le rôle de l'ingénieur de choisir le "bon" outil de description, et la "meilleure solution".

CONSTITUTION D'UN GRAPHE D'ETATS

1/ Etat

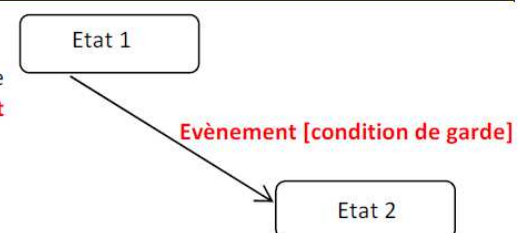
- Dans un graphe d'états, la loi d'évolution des états n'est évidemment pas aléatoire. Cette loi est choisie par le créateur du graphe afin que celui-ci remplisse la fonction précise.
- Chaque état est dessiné sous la forme d'un rectangle aux coins arrondis, contenant son nom.**

- Il existe différentes sortes d'états :

- l'état initial : pseudo-état qui indique un point d'entrée dans un graphe ;
- l'état final : non obligatoire, il indique que le système décrit n'a plus d'état actif ;
- état : un état représente une période de la vie du système. Pendant cette période, le système accomplit une ou plusieurs "actions" (ou "activités"), ou attend un "évènement". Il peut être actif ou non ; plusieurs états peuvent être actifs en même temps dans le même système.

2/ Transition, évènement et condition de garde

- Une transition représente le passage **instantané** d'un état vers un autre. Une transition ne peut donc pas avoir de durée. On appelle **état source** l'état de départ d'une transition et **état destination** l'état d'arrivée.
- Une transition n'est évaluée que si l'état source est actif.



- Une transition est déclenchée par un évènement. En d'autres termes : c'est l'arrivée d'un évènement qui conditionne le franchissement de la transition.
- Une transition peut aussi être automatique, lorsqu'on ne spécifie pas l'évènement qui la déclenche. Elle est alors franchie lorsque les actions de l'état source sont terminées.
- En plus de spécifier un évènement précis, il est aussi possible de conditionner une transition à l'aide d'une "condition de garde" : il s'agit d'une expression booléenne encadrée de crochets, évaluée lorsque l'état précédant la transition est vrai et que l'évènement déclencheur se produit. Si la condition de garde est vraie, la transition est alors franchie, sinon elle ne l'est pas et l'évènement est perdu.

Rq : Différence entre évènement et condition de garde : un évènement est parfaitement daté dans le temps, il correspond par exemple à un passage d'une variable de 0 à 1 à un instant précis (front montant) ; une condition de garde n'est pas datée, elle doit être vraie (niveau logique 1) à l'instant où l'évènement survient pour que la transition soit franchie.

Exemple d'évènement : appui sur un bouton-poussoir, capteur fin de course atteint, etc.

Exemple de condition de garde : vitesse du véhicule non nulle, température > 20°C, etc.

- Il existe 2 sortes d'évènements :

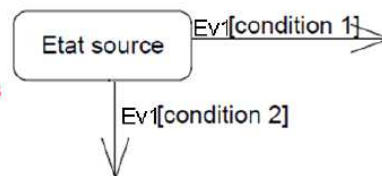
- évènement signal : un signal est émis à destination d'un objet ; cette émission est asynchrone, c'est-à-dire que le destinataire ne l'attend pas, et qu'elle peut survenir n'importe quand. Par exemple : l'appui sur un bouton-poussoir ;

- évènement temporisé : un évènement de ce type fait intervenir le temps. Il nécessite l'utilisation des mots réservés **when(date)** pour spécifier un temps absolu, ou **after(durée)** pour spécifier une durée à partir de l'instant d'activation de l'état précédent.

- Transition conditionnelle :

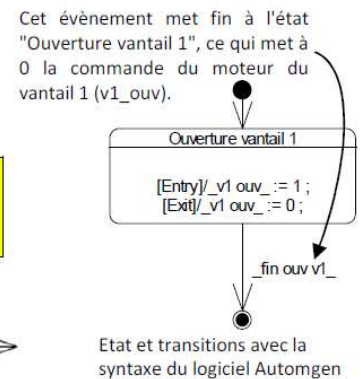
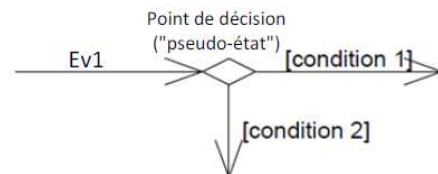
Plusieurs transitions peuvent quitter un même état.

Une seule d'entre elles doit être déclenchée ; les évènements et / ou les conditions de garde doivent donc être exclusives.



Une autre représentation est possible, qui utilise le point de décision :

Il est possible d'utiliser une condition particulière, notée [else], sur un des segments en aval d'un point de décision. Ce segment n'est franchissable que si les conditions des autres segments sont toutes fausses.



3/ Actions (ou activités)

- Le lancement des actions (ou activités) à l'intérieur de l'état actif est organisé selon des mots réservés :

- **entry/** est suivi des actions exécutées lorsque l'état devient actif ;
- **do/** est suivi d'1 ou plusieurs actions exécutées dans l'ordre de leur écriture, à partir de l'instant où l'activité /entry est terminée ;
- **exit/** est suivi des actions qui se déroulent lorsque l'état se désactive ;

• Pendant que l'état est actif, un évènement peut lancer une action avec la syntaxe : évènement[condition de garde]/ suivi de l'action. Cette action est lancée chaque fois que l'évènement survient, tant que l'état est actif, en parallèle avec l'action du do/. On parle alors de **transition interne**.

Rq : On peut aussi ne pas utiliser de mot réservé, auquel cas cela correspond à un do/.

Rq2 : Un état peut ne pas contenir d'action. Il sert alors à attendre le déclenchement de la transition suivante.

- Une action peut être :

- un ordre de mise à 1 d'une sortie ; la syntaxe est alors **SORTIE := 1 ;** ;
- un ordre de mise à 0 d'une sortie ; la syntaxe est alors **SORTIE := 0 ;** ;
- une modification d'une variable numérique interne, par exemple un compteur C ; la syntaxe est alors **C := C + 1 ;** pour incrémenter la valeur du compteur C de 1.

▲ peut être oublié

4/ Lien avec les entrées/sorties de la fonction "Traiter l'information" de la chaîne d'information

On aura compris que :

- Les informations en entrée vont intervenir dans les transitions du graphe d'états.
- Les ordres de commande en sortie vont intervenir dans les actions lancées dans les états actifs.

DEUX EXEMPLES DE CONSTITUTION D'UN GRAPHE D'ETATS



1/ Machine à laver

• On peut ici commencer par déterminer les états : Prélavage, Lavage, Rinçage, Essorage et bien sûr, Arrêt.

• On complète le graphe en plaçant des transitions entre les états, pour indiquer la **loi d'évolution des états en fonction des entrées**. On se sert pour cela du comportement séquentiel souhaité ou observé.

Les variables d'entrée sont ici les informations suivantes :

- **m** : bouton marche/arrêt du lave-linge ; m = 1 indique marche.

- **p** : bouton qui indique si le programme du lavage sélectionné comporte (p = 1) ou non (p = 0) une phase de prélavage.

Les durées des différentes étapes du lavage sont fixées par le constructeur : prélavage : 10 minutes ; lavage : 30 minutes ; rinçage : 10 minutes ; essorage : 5 minutes.

Compléter le graphe des états avec les transitions et les événements déclencheurs :

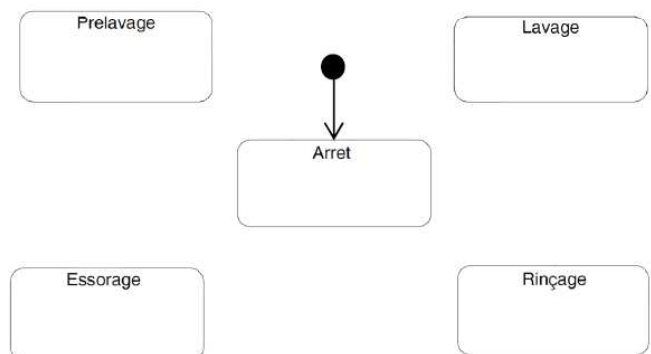
• Il reste ensuite à définir les variables de sorties, pour lancer les actions :

- **Cmd_m** : égale à 1 si le moteur doit tourner, sinon 0 ;

- **Vitesse** : égale à 0 à l'arrêt ; 1 000 tr/min en prélavage, en lavage et en rinçage ; 1 400 tr/min en essorage.

Compléter les états avec les actions à lancer.

Compléter ce graphe d'états afin que l'utilisateur puisse interrompre le cycle du lave-linge quel que soit l'état actif de celui-ci. La variable d'entrée associée à la demande d'arrêt est la variable binaire **stop**.



2/ Porte de garage basculante



Boîtier de commande mural



Bouton-poussoir tel



Fonctionnement souhaité :

Une porte de garage basculante est mise en mouvement par un moteur à 2 sens de rotation : **Ouvrir** ou **Fermer**.

Un capteur **butée** détecte une surintensité moteur, qui correspond à l'atteinte d'une position en butée de la porte : ouverte ou fermée. Une télécommande possède 1 bouton **tel** dont l'appui produit le fonctionnement suivant :

- **Ouvrir** si la porte est fermée ;

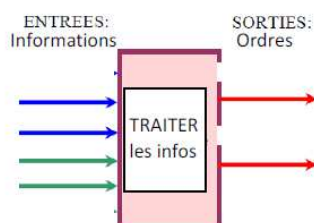
- **Fermer** si elle est ouverte ;

- inverser le sens si la porte est en mouvement.

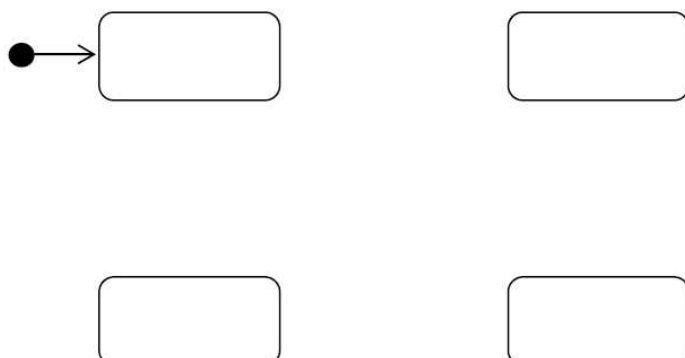
Enfin, un boîtier de commande mural dans le garage possède 2 boutons-poussoirs **cmd_ouv** et **cmd_fer**.

On suppose qu'à la mise sous tension, la porte est en position fermée.

Lister les entrées-sorties :



Proposer le graphe d'états :



Rq : Une alternative à l'utilisation du capteur butée est de se servir de temporisations de 10 s. Modifier le graphe en conséquence.

3/ Synthèse : Démarche de modélisation du comportement séquentiel d'un système par un graphe d'états

- Définir la frontière du système et recenser les variables d'entrée et de sortie ;
- Recenser, nommer et tracer les états du système ;
- Tracer les transitions entre les états en fonction du comportement séquentiel souhaité ou observé ;
- Définir les conditions et événements associés à chaque transition et les actions associées à chaque état.

HIERARCHISATION AVEC LA NOTION D'ETAT COMPOSITE

1/ Etat composite

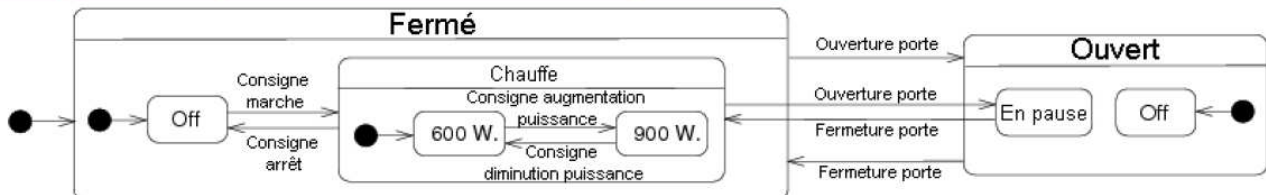
- Lorsque le comportement à décrire :

- risque d'être peu lisible car trop complexe,
- fait apparaître des parties "séparables" de l'ensemble,

alors **on extrait ces parties pour en faire des sous-graphes séparés, hiérarchiquement inférieurs au graphe principal. On englobe ces sous-graphes dans un état appelé "état composite" ou "super-état".**

Un état composite est composé de plusieurs sous-états internes. Un état composite possède un état initial.

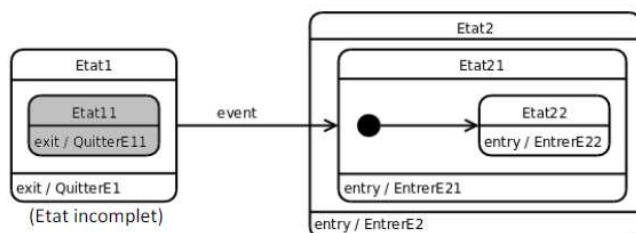
2/ Exemple avec le comportement d'un four à micro-ondes et généralisation



- Il y a 2 états composites "Fermé" et "Ouvré". L'état composite "Fermé" contient lui-même un sous-état composite "Chauffe". Il contient lui-même un sous-graphe formé de 2 sous-états "600 W" et "900 W".

- Il peut y avoir des transitions ayant pour source / cible la frontière d'un sous-état ou la frontière d'un état composite.
- Quand plusieurs transitions sont possibles, on choisit de suivre celle qui part de l'état le plus en bas de la hiérarchie des états actifs, donc ici partant de "Chauffe" et non pas de "Fermé". Autrement dit :
 - si l'état actif est "Chauffe" quand on ouvre la porte, l'état "En pause" s'active ;
 - si l'état actif est "Off" contenu dans "Fermé" quand on ouvre la porte, l'état "Ouvré" s'active et donc l'état "Off" qu'il contient.

- Autre exemple de l'évolution avec des états composites :



Supposons que l'état actif soit l'Etat11. Alors, lorsque l'évènement "event" arrive, l'Etat1 est désactivé et il se produit successivement les actions suivantes :

- Quitter E11
 - Quitter E1
 - EntrerE2
 - EntrerE21
 - EntrerE22,
- et l'état actif est l'Etat22 (les états composites Etat 21 et Etat2 sont aussi actifs).

- Il existe une notation abrégée pour un état composite :



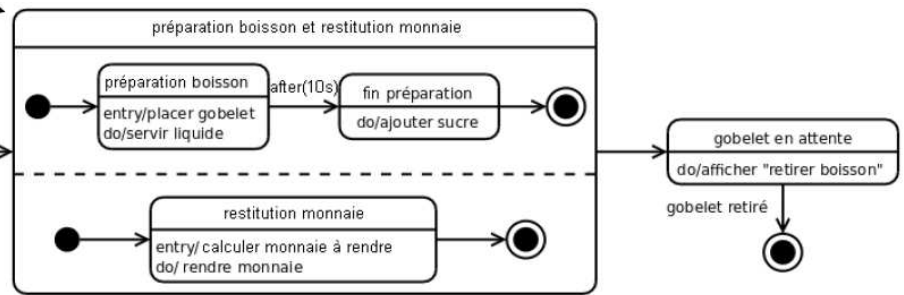
3/ Concurrency et synchronisation

- Dans un état composite, plusieurs graphes d'états peuvent évoluer simultanément (en parallèle). On dit qu'il y a **concurrence de plusieurs états**.

Par exemple pour un distributeur de boissons :

L'état composite est dit **orthogonal** car il comporte **plus d'une région**, chaque région représentant un flot d'exécution.

- Graphiquement, dans un état orthogonal, les différentes régions sont séparées par un trait horizontal ou vertical en pointillés allant d'un bord à l'autre de l'état composite.

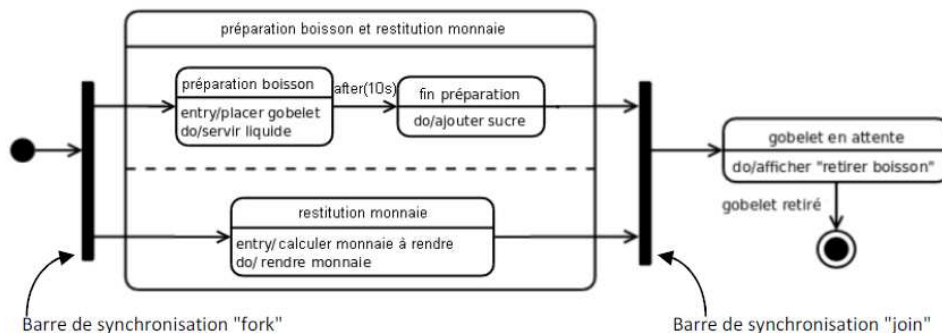


- Chaque région peut posséder un état initial et final. Une transition qui atteint la bordure d'un état composite orthogonal est équivalente à une transition qui atteint les états initiaux de toutes ses régions concurrentes.

Toutes les régions concurrentes d'un état composite orthogonal doivent atteindre leur état final pour que l'état composite soit considéré comme terminé.

La synchronisation est alors automatique et la transition de sortie de l'état composite est déclenchée.

- Il est également possible de représenter ce type de comportement au moyen de **transitions concurrentes constituées de barres de synchronisation "fork" et "join"**. Le graphe ci-dessous est une représentation équivalente à la précédente :

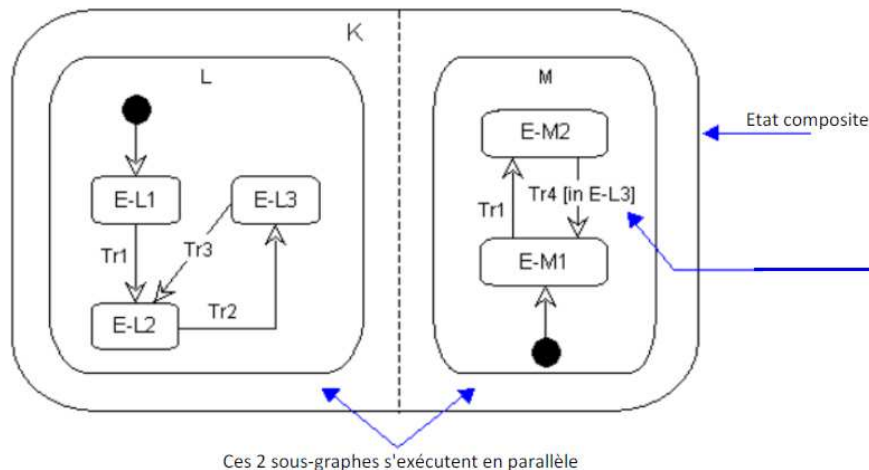


- Les transitions automatiques (sans évènement ou condition de garde) qui partent d'une barre de synchronisation "fork" se déclenchent en même temps. On ne franchit une barre de synchronisation "join" qu'après déclenchement de toutes les transitions qui s'y rattachent.**

- Rq : Sur ce diagramme, l'état orthogonal "préparation boisson et restitution monnaie" peut éventuellement ne pas apparaître (tout en gardant la représentation de ses sous-états) pour alléger la représentation, car la notion de concurrence est clairement apparente de par l'utilisation des barres.

4/ Comment spécifier la synchronisation de 2 sous-graphes ?

- Il est aussi possible, et très utile, de conditionner une transition par l'activité de l'état d'un autre sous-graphe (!). Un exemple éclaircira les choses :



L'état composite orthogonal K est composé des sous-états L et M.

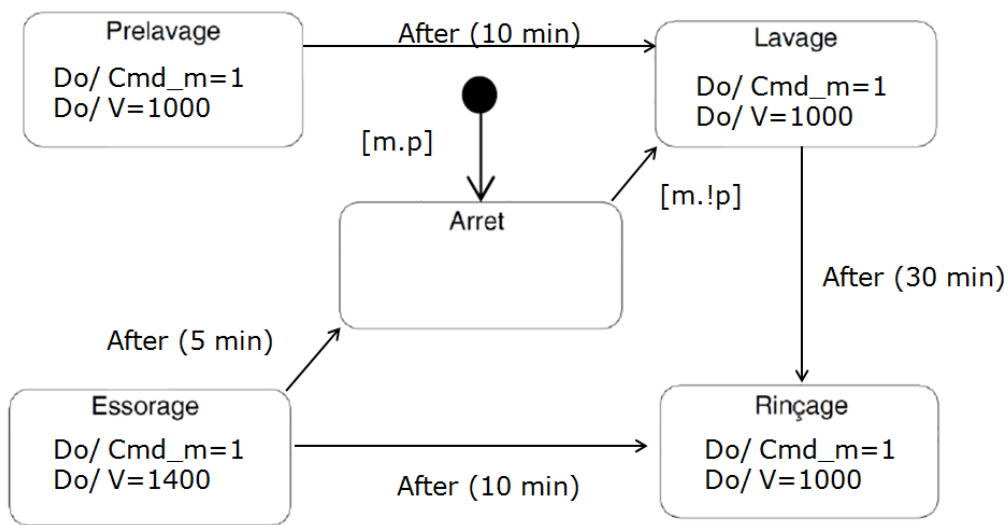
L et M s'activent simultanément, et évoluent indépendamment, en parallèle. Au départ, E-L1 et E-M1 sont actifs.

Pour que M passe d'E-M2 à E-M1, il faut que l'évènement Tr4 survienne, à condition que E-L3 soit actif.

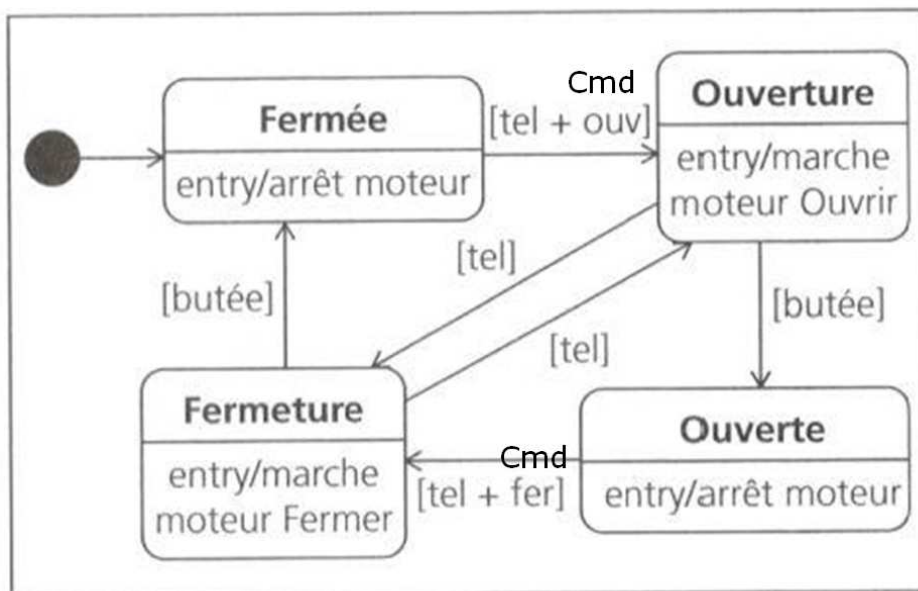
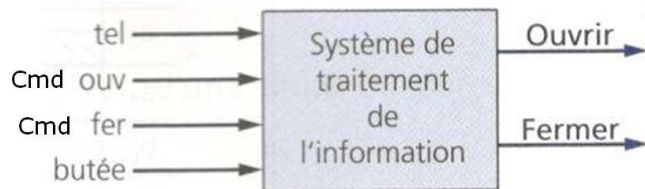
Cet évènement est conditionné par l'activité de l'état E-L3 de l'autre sous-graphe.

- La syntaxe de la condition de garde vérifiant l'activité de ETAT est : [in ETAT].**

Correction machine à laver



Correction porte de garage basculante :



Chronogramme

