

## **Les Grafcet**

### **1- Historique**

En 1975, un groupe d'universitaires et industriels de la section "Systèmes Logiques" de l'**AFCET** (Association Française de Cybernétique Economique et Technique) se sont fixés l'objectif de définir un formalisme adapté à la représentation des évolutions séquentielles d'un système et ayant les caractéristiques suivantes :

- Simple;
- Accepté par tous;
- Intelligible à la fois par les concepteurs et les exploitants;
- Fournissant potentiellement des facilités de passage à une réalisation, à base matérielle et/ou logicielle de l'automatisme ainsi spécifié.

Au début, le travail consista à dresser un état de l'art des différentes approches de modélisation du comportement de tels automatismes. Trois grandes classes d'outils de modélisation furent ainsi recensées :

- les organigrammes ;
- Les Réseaux de Pétri;
- Les graphes d'état.

L'analyse des avantages et inconvénients de ces outils mena, en 1977, à la définition du **GRAFCET**, ainsi nommé pour, à la fois marquer l'origine de ce nouvel outil de modélisation « **AFCET** » et son identité **GRA**phe **F**onctionnel de **C**ommande **E**tapes-**T**ransitions). Les résultats de ces travaux firent l'objet d'une publication officielle dans la revue "**Automatique et Informatique Industrielle**" en décembre 1977, date que la communauté considère aujourd'hui comme correspondant à la date de naissance effective du **GRAFCET**.

### **2 - Définition**

Le **GRAFCET** (**GRA**phe **F**onctionnel de **C**ommande par **E**tapes et **T**ransitions) ou **SFC** (Sequential **F**onction **C**hart) est un outil **graphique** qui décrit les différents comportements de l'évolution d'un **automatisme** et établit une correspondance à caractère séquentiel et combinatoire entre :

- Les **ENTREES**, c'est-à-dire les transferts d'informations de la Partie Opérative vers la Partie Commande,
- Les **SORTIES**, transferts d'informations de la Partie Commande vers la Partie Opérative.

C'est un outil graphique puissant, directement exploitable, car c'est aussi un langage pour la plupart des **API** existants sur le marché. Lorsque le mot **GRAFCET** (*en lettre capitale*) est utilisé, il fait référence à l'outil de **modélisation**. Lorsque le mot **grafcet** est écrit en minuscule, il fait alors référence à **un modèle** obtenu à l'aide des règles du **GRAFCET**. (Exemple : *J'ai utilisé le **GRAFCET** pour concevoir cette machine, jette un coup d'œil sur le **grafcet de sécurité** et dit moi ce que tu pense*)

Le **GRAFCET** comprend :

- des **étapes** associées à des actions ;
- des **transitions** associées à des **réceptivités** ;
- des **liaisons orientées** reliant étapes et transitions.

### 3 - Description du GRAFCET

La description du comportement attendu d'un automate peut se représenter par un **GRAFCET** d'un certain « **niveau** ». La caractérisation du «**niveau**» du GRAFCET nécessite de prendre en compte trois dimensions :

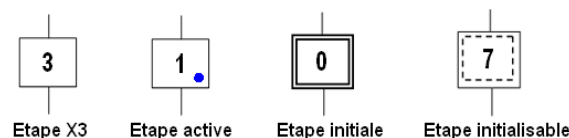
- **Le point de vue**, caractérisant le point de vue selon lequel un observateur s'implique dans le fonctionnement du système pour en donner une description. On distingue trois points de vue :
  - Un point de vue système,
  - Un point de vue Partie Opérative,
  - Un point de vue Partie Commande.
- **La spécifications**, caractérisant la nature des spécifications techniques auxquelles doit satisfaire la Partie Commande. On distingue trois groupes de spécifications :
  - Spécifications fonctionnelles,
  - Spécifications technologiques,
  - Spécifications opérationnelles.
- **La finesse**, caractérisant le niveau de détail dans la description du fonctionnement, d'un niveau global (ou macro-représentation) jusqu'au niveau de détail complet où toutes les actions et informations élémentaires sont prises en compte.

### 4 - Les concepts de base du GRAFCET

#### 4.1 - Etape

Une **étape** symbolise un état ou une partie de l'état du système automatisé. L'étape possède deux états possibles : **active** représentée par un jeton dans l'étape ou **inactive**. L'étape  $i$ , représentée par un carré repéré numériquement, possède ainsi une variable d'état, appelée variable d'étape  $X_i$ . Cette variable est une variable booléenne valant **1** si l'étape est active, **0** sinon.

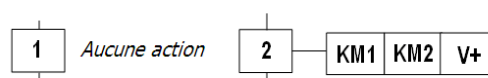
La situation initiale d'un système automatisé est indiquée par une étape dite **étape initiale** et représentée par un carré double.



*Remarque :* Dans un **grafcet** il doit y avoir au moins une étape initiale.

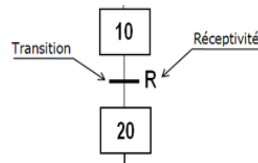
#### 4.2 - Actions associées aux étapes

A chaque étape est associée une **action** ou plusieurs, c'est à dire un ordre vers la partie opérative ou vers d'autres grafkets. Mais on peut rencontrer aussi une même action associée à plusieurs étapes ou une étape **vide** (sans action).



### 4.3 - Transition

Une transition indique la possibilité d'évolution qui existe entre deux étapes et donc la succession de deux activités dans la partie opérative. Lors de son franchissement, elle va permettre l'évolution du système. A chaque transition est associée une condition logique appelée réceptivité qui exprime la condition nécessaire pour passer d'une étape à une autre.



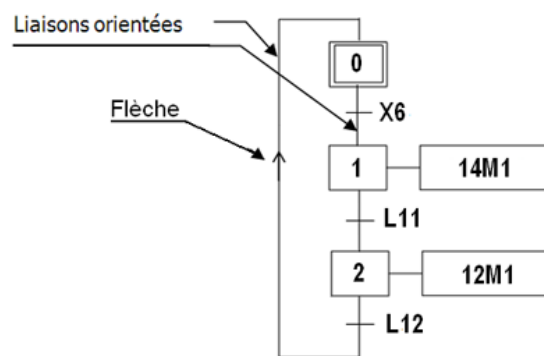
La réceptivité qui est une information d'entrée qui est fournie par :

- l'opérateur : pupitre de commande,
- la partie opérative : états des capteurs,
- du temps, d'un comptage ou toute opération logique, arithmétique...
- du grafcet : d'autres grafcets pour la liaison entre grafcets ou de l'état courant des étapes du grafcet (les  $X_i$ ),
- d'autres systèmes : dialogue entre systèmes,
- .....

Remarque: Si la réceptivité n'est pas précisée, alors cela signifie qu'elle est toujours vraie. (=1)

### 4.4 - Liaisons orientées

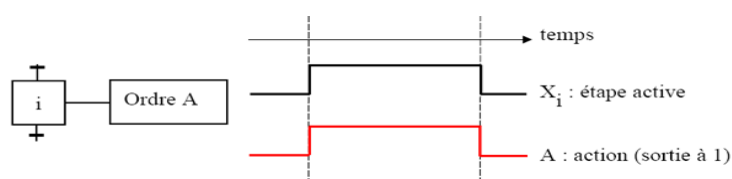
Elles sont de simples traits verticaux qui relient les étapes aux transitions et les transitions aux étapes. Elles sont normalement orientées de haut vers le bas. Une flèche est nécessaire dans le cas contraire.



### 4.3 - Classification des actions associées aux étapes

L'action associée à l'étape peut être de 3 types : **continue**, **conditionnelle** ou **mémorisée**. Les actions peuvent être classées en fonction de leur durée par rapport à celle de l'étape.

#### 4.3.1 - Actions continues :

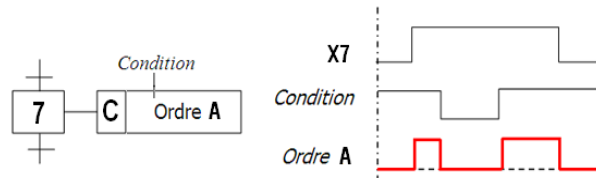


L'ordre est émis, de façon continue, tant que l'étape, à laquelle il est associé, est active.

#### 4.3.2 - Actions conditionnelles:

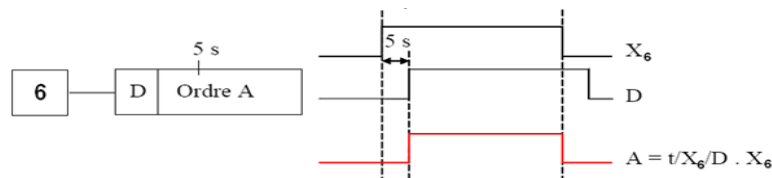
Une action **conditionnelle** n'est exécutée que si l'étape associée est active et si la condition associée est vraie. Elles peuvent être décomposées en 3 cas particuliers:

##### 4.3.2.1 - Action conditionnelle simple : Type C



##### 4.3.2.1 - Action retardée : Type D (delay)

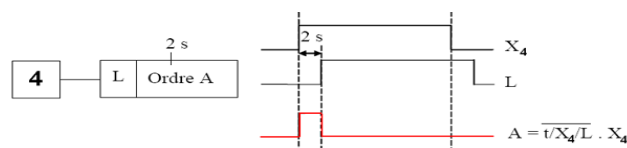
Le temps intervient dans cet ordre conditionnel comme condition logique. L'indication du temps s'effectue par la notation générale " t / xi / q " dans laquelle "xi" indique l'étape prise comme origine du temps et "q" est la durée du retard.



Exemple : "t /x6/ 5s" : prendra la valeur logique 1, 5s après la dernière activation de l'étape 6.

##### 4.3.2.2 - Action de durée limitée: Type L (limited)

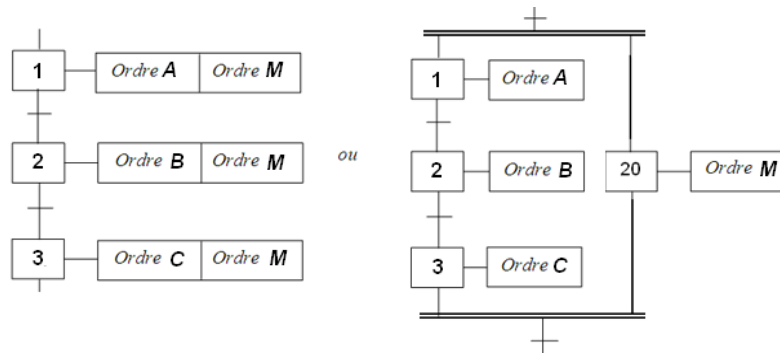
L'ordre est émis dès l'activation de l'étape à laquelle il est associé ; mais la durée de cet ordre sera limitée à une valeur spécifiée.



L'ordre "A" est limité à 2s après l'activation de l'étape 4.

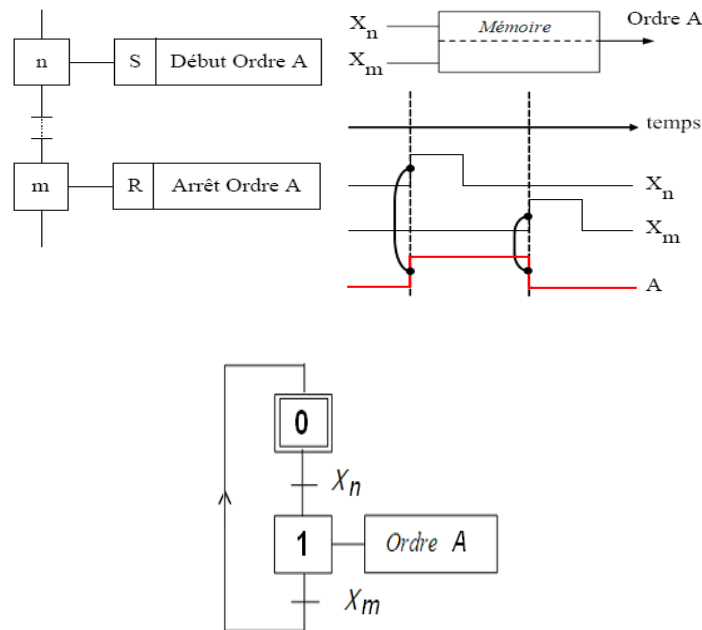
#### 4.3.3 - Action maintenue sur plusieurs étapes:

Afin de maintenir la continuité d'une action sur plusieurs étapes, il est possible de répéter l'ordre continu relatif à cette action, dans toutes les étapes concernées ou d'utiliser une description sous forme de séquences simultanées (*Les séquences simultanées seront traitées ultérieurement*).



#### 4.3.4 - Action mémorisée :

Le maintien d'un ordre, sur la durée d'activation de plusieurs étapes consécutives, peut également être obtenu par la mémorisation de l'action, obtenue par l'utilisation d'une fonction auxiliaire appelée fonction mémoire.



Cette fonction pourra être décrite par un **GRAFCET**

### 5 - Règles d'évolution d'un GRAFCET

#### 5.1 - Règle N°1 : Condition initiale

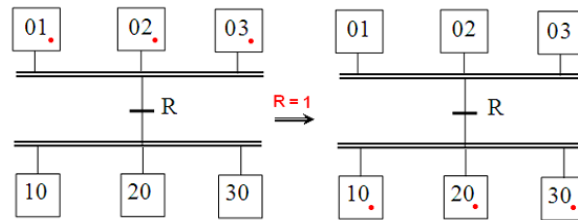
A l'instant initial, seules les étapes initiales sont actives.

#### 5.2 - Règle N°2 : Franchissement d'une transition.

Pour qu'une transition soit validée, il faut que toutes ses étapes amont (immédiatement précédentes reliées à cette transition) soient actives. Le franchissement d'une transition se produit lorsque la transition est validée, **ET seulement si** la réceptivité associée est **vraie**.

#### 5.3 - Règle N°3 : Evolution des étapes actives

Le franchissement d'une transition entraîne obligatoirement l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes.



#### 5.4 - Règle N°4 : Franchissement simultané

Toutes les transitions simultanément franchissables à un instant donné sont simultanément franchies.

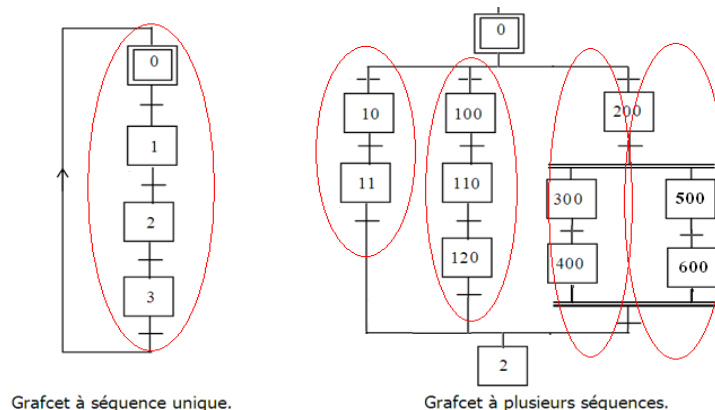
#### 5.5 - Règle N°5 : Conflit d'activation

Si une étape doit être simultanément désactivée par le franchissement d'une transition aval, et activée par le franchissement d'une transition amont, alors elle reste active. On évite ainsi des commandes transitoires (néfastes à la partie opérative).

### 6 - Les structures de base

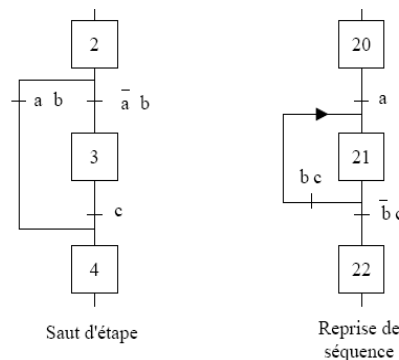
#### 6.1 - Notion de Séquence :

Une séquence, dans un Grafset, est une suite d'étapes à exécuter l'une après l'autre. Autrement dit chaque étape ne possède qu'une seule transition AVAL et une seule transition AMONT.



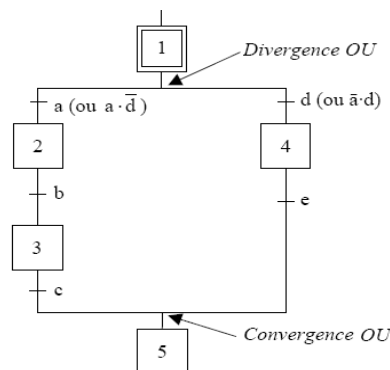
#### 6.2- Saut d'étapes et reprise de séquence

Le saut d'étapes permet de sauter une ou plusieurs étapes lorsque les actions associées sont inutiles à réaliser, La reprise de séquence (ou boucle) permet de reprendre, une ou plusieurs fois, une séquence tant qu'une condition n'est pas obtenue.



### 6.3 - Aiguillage entre deux ou plusieurs séquences (Divergence en OU)

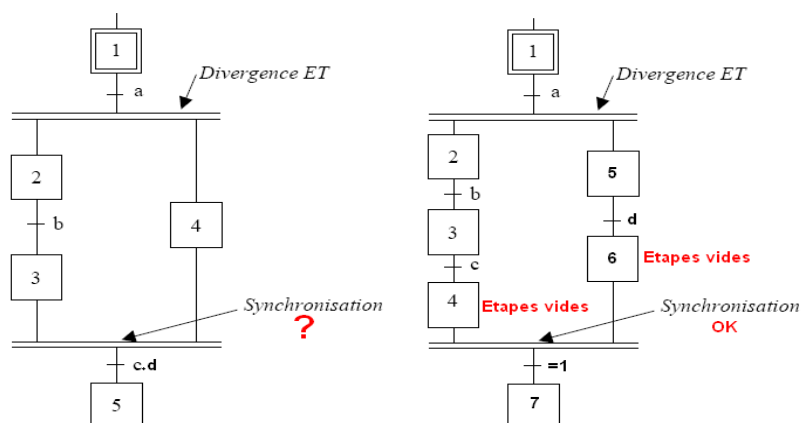
On dit qu'il y a **Aiguillage** ou **divergence en OU** lorsque le grafcet se décompose en deux ou plusieurs séquences selon un choix conditionnel. Comme la divergence en OU on rencontre aussi la convergence en OU. On dit qu'il y a convergence en OU, lorsque deux ou plusieurs séquences du grafcet convergent vers une seule séquence.



Si les deux conditions  $a$  et  $d$  sont à 1 simultanément, les étapes 2 et 4 vont devenir actives simultanément, situation non voulue par le concepteur. Donc elles doivent être des conditions **exclusives**

### 6.4 - Parallélisme entre deux ou plusieurs séquences (ou séquences simultanées ou divergence convergence en ET) :

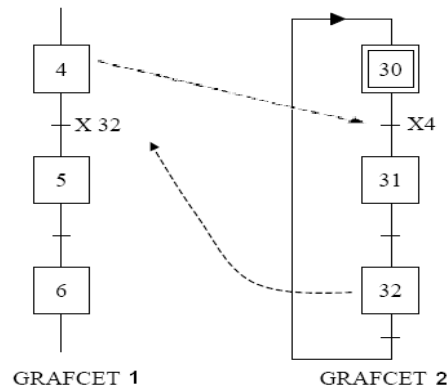
Au contraire de l'aiguillage où ne peut se dérouler qu'une seule activité à la fois, On dit qu'on se trouve en présence d'un parallélisme structurel, si plusieurs activités indépendantes pouvant se dérouler en parallèle. Le début d'une divergence en ET et la fin d'une convergence en ET d'un parallélisme structurel sont représentés par deux traits parallèles.



La synchronisation permet d'attendre la fin de plusieurs activités se déroulant en parallèle, pour continuer par une seule.

### 7 - Liaison entre grafkets :

Une étape dans un grafcet peut servir comme réceptivité à une autre étape d'un autre grafcet. Cette méthode est utilisée aussi pour synchroniser deux grafkets c'est à dire rendre l'évolution de l'un dépendante de l'évolution de l'autre.



## 8 - Mise en équation d'un grafcet :

### 8.1 - Règle générale :

Pour qu'une étape soit activée il faut que :

- L'étape immédiatement précédente soit active ;
- La réceptivité immédiatement précédente soit vraie ;
- L'étape immédiatement suivante soit non active ;
- Après activation l'étape mémorise son état.

