

*République Algérienne Démocratique et Populaire*  
*Ministère de L'Enseignement Supérieur et de la*  
*Recherche Scientifique*

*Université 8 mai 1945 – GUELMA*

*Faculté des Sciences et de la Technologie*

*Département d'Electrotechnique & d'Automatique*



*Polycopié de Cours*

**Automatismes industriels**

*Réalisés par :*

1. Dr: AIDOU MOHAMMED
2. Dr: SEBBAGH ABDENNOUR

Année universitaire : 2019-2020

# SOMMAIRE

---

<b>SOMMAIRE</b> .....	<b>I</b>
<b>LISTE DES FIGURES</b> .....	<b>IV</b>
<b>AVANT-PROPOS</b> .....	<b>VI</b>

## **CHAPITRE 1. INTRODUCTION AUX SYSTÈMES AUTOMATISÉS** ..... **1**

1.1. FONCTION GLOBALE.....	1
1.1.1. MATIÈRE D'ŒUVRE : .....	1
1.1.2. VALEUR AJOUTÉE : .....	2
1.2. AUTOMATISATION.....	2
1.3. STRUCTURE D'UN SYSTÈME AUTOMATISÉ.....	3
1.3.1. LA PARTIE OPÉRATIVES (PO) : .....	4
1.3.2. LA PARTIE COMMANDE (PC) : .....	5
1.3.3. LA PARTIE RELATION (PR).....	5
1.4. CLASSIFICATION DES SYSTÈMES AUTOMATISÉS : .....	6
1.4.1. CLASSIFICATION SUIVANT LE DOMAINE D'APPLICATION : .....	6
1.4.2. CLASSIFICATION SUIVANT LA NATURE DE LA MATIÈRE D'ŒUVRE .....	6
1.4.3. CLASSIFICATION SUIVANT LA NATURE DES FLUX.....	6
1.4.4. CLASSIFICATION SUIVANT LES CRITÈRES TECHNICO-ÉCONOMIQUES.....	7
1.5. SPÉCIFICATION DES NIVEAUX DU CAHIER DES CHARGES : .....	7
1.5.1. DÉFINITION D'UN CAHIER DES CHARGES FONCTIONNEL.....	7
1.5.2. OBJECTIFS DU CAHIER DES CHARGES FONCTIONNEL .....	8
1.5.3. MÉTHODOLOGIE DU CAHIER DES CHARGES FONCTIONNEL .....	8
1.5.3.1. L'étude d'opportunité .....	8
1.5.3.2. L'étude de faisabilité .....	8
1.5.3.3. L'Analyse Fonctionnelle du Besoin .....	9
1.5.3.4. L'Expression Fonctionnelle du Besoin.....	9
1.6. OUTILS DE REPRÉSENTATION DES SPÉCIFICATIONS FONCTIONNELLES : .....	9

## **CHAPITRE 2 : LE GRAFCET**..... **11**

2.1. DÉFINITION ET NOTIONS DE BASES.....	11
2.1.1. DÉFINITION .....	11
2.1.2. DESCRIPTION DU GRAFCET .....	11
2.1.3. LES AVANTAGES : .....	12
2.2. RÈGLES D'ÉTABLISSEMENT DU GRAFCET .....	12
2.2.1. ÉTAPE.....	12
2.2.2. ACTIONS ASSOCIÉES AUX ÉTAPES .....	13
2.2.3. TRANSITION .....	13
2.2.4. LIAISONS ORIENTÉES .....	14
2.2.5. CLASSIFICATION DES ACTIONS ASSOCIÉES AUX ÉTAPES .....	14
2.2.5.1. Actions continues : .....	14
2.2.5.2. Actions conditionnelles: .....	14
2.2.5.3. Action maintenue sur plusieurs étapes: .....	15
2.2.5.4. Action mémorisée : .....	16
2.3. RÈGLES D'ÉVOLUTION D'UN GRAFCET.....	16
2.4. SÉLECTION DE SÉQUENCE ET SÉQUENCES SIMULTANÉES : .....	17
2.4.1. SÉLECTION DE SÉQUENCE .....	17

2.4.2. SÉQUENCES SIMULTANÉES.....	18
2.4.3. SAUT D'ÉTAPES ET REPRISE DE SÉQUENCE.....	18
2.4.4. AIGUILLAGE ENTRE DEUX OU PLUSIEURS SÉQUENCES (DIVERGENCE EN OU).....	19
2.4.5. PARALLÉLISME ENTRE DEUX OU PLUSIEURS SÉQUENCES .....	20
2.4.6. LIAISON ENTRE GRAFCETS : .....	22
2.5. ORGANISATION DES NIVEAUX DE REPRÉSENTATION .....	22
2.5.1. LES DEUX NIVEAUX DE REPRÉSENTATION. ....	23
2.6. MISE EN ÉQUATION D'UN GRAFCET : .....	24
2.6.1. RÈGLE GÉNÉRALE : .....	25
2.6.3. SÉQUENCES PARALLÈLES.....	27
2.6.4. GESTION DES MODES MARCHE/ARRÊT ET ARRÊT D'URGENCE .....	27
2.7. MATÉRIALISATION D'UN GRAFCET .....	28
2.7.1. CAS D'UN GRAFCET LINÉAIRE.....	28
2.7.2. DIVERGENCE SIMPLE EN ET.....	30
2.7.3. DIVERGENCE EN OU .....	30
2.7.4. CONVERGENCE EN ET .....	31
2.7.5. CONVERGENCE EN OU .....	31
2.7.6. EXEMPLE DE GRAFCET AVEC SÉLECTION DE SÉQUENCE .....	31

### **CHAPITRE 3 : AUTOMATE PROGRAMMABLE..... 34**

3.1. DÉFINITION.....	34
3.2. STRUCTURE INTERNE ET DESCRIPTION DES ÉLÉMENTS D'UN A.P.I : .....	34
3.3. CHOIX DE L'UNITÉ DE TRAITEMENT : .....	37
3.4. CHOIX D'UN AUTOMATE PROGRAMMABLE INDUSTRIEL : .....	38
3.5. LES INTERFACES D'ENTRÉES-SORTIES : .....	38
3.6. OUTILS GRAPHIQUES ET TEXTUELS DE PROGRAMMATION : .....	41
3.6.1. OBJETS COMMUNS À TOUS LES LANGAGES .....	41
3.6.2. LANGAGE LD.....	42
3.6.3. LANGAGE IL .....	42
3.6.4. LANGAGE FBD .....	43
3.6.5. LANGAGE SFC.....	43
3.6.6. LANGAGE ST .....	44
3.7. MISE EN ŒUVRE D'UN AUTOMATE PROGRAMMABLE INDUSTRIEL : .....	45

### **CHAPITRE 4 : GUIDE D'ETUDE DES MODES MARCHE ET ARRÊT (G.E.M.M.A)..... 47**

4.1. INTRODUCTION : .....	47
4.2. CONCEPT ET STRUCTURATION DU GEMMA.....	48
4.2.1. PARTIE COMMANDE HORS ÉNERGIE : .....	48
4.2.2. PARTIE COMMANDE EN ÉNERGIE ET ACTIVE : .....	48
4.3. LES RECTANGLES-ÉTATS : .....	49
4.4. PROCÉDURES DE FONCTIONNEMENT (LA FAMILLE F).....	52
4.4.1. PRODUCTION NORMALE (F1) .....	52
4.4.2. MARCHES DE PRÉPARATION (F2).....	52
4.4.3. MARCHES DE CLÔTURE (F3) .....	53
4.4.4. MARCHES DE VÉRIFICATION DANS LE DÉSORDRE (F4) .....	53
4.4.5. MARCHES DE VÉRIFICATION DANS L'ORDRE (F5) .....	54
4.4.6. MARCHES DE TEST (F6).....	55
4.5. PROCÉDURES D'ARRÊT (LA FAMILLE A) .....	55
4.5.1. ARRÊT DANS L'ÉTAT INITIAL (A1) .....	55
4.5.2. ARRÊT DEMANDÉ EN FIN DE CYCLE (A2).....	55
4.5.3. ARRÊT DEMANDÉ DANS UN ÉTAT DÉTERMINÉ (A3).....	56
4.5.4. ARRÊT OBTENU (A4) .....	56
4.5.5. PRÉPARATION POUR REMISE EN ROUTE APRÈS DÉFAILLANCE (A5) .....	56
4.5.6. MISE DE LA P.O. DANS UN ÉTAT INITIAL (A6) .....	56

4.5.7. MISE DE LA P.O. DANS UN ÉTAT DÉTERMINÉ (A7) .....	57
4.6. PROCÉDURES EN DÉFAILLANCES (LA FAMILLE D) , .....	57
4.6.1. ARRÊT D'URGENCE (D1) .....	57
4.6.2. DIAGNOSTIC ET/OU TRAITEMENT DE DÉFAILLANCE (D2) .....	58
4.6.3. PRODUCTION TOUT DE MÊME (D3).....	58
4.7. UTILISATION PRATIQUE DU GEMMA ET APPLICATIONS. ....	59
4.7.1. MISE EN CONTEXTE .....	59
4.7.2. UTILISATION DU GEMMA (SÉLECTION DES MODES ET DES ÉVOLUTIONS). ....	60
4.7.3. UTILISATION DU GEMMA (CONDITIONS D'ÉVOLUTION ENTRE MODES MARCHES / ARRÊTS).....	62
4.7.4. EXEMPLES TYPES DE GEMMA : .....	63
4.8. PASSAGE DU GEMMA À UNE SPÉCIFICATION GRAFCET : .....	68
4.8.1. EFFET DU GEMMA SUR LE GRAFCET : .....	69
4.8.1.1. Elaboration du Grafcet de conduite d'un système : .....	69
4.8.1.2. Enrichissement du Grafcet : .....	69
4.8.1.3. Découpage en tâches .....	70
4.8.1.4. Cas de l'arrêt d'urgence : .....	73
4.8.1.5. Les arrêts d'urgence par inhibition des actions : .....	74
4.8.1.6. Les arrêts d'urgence par figeage : .....	74
4.8.1.7. Les arrêts d'urgence utilisant une tâche spéciale : .....	75
<b>RÉFÉRENCES BIBLIOGRAPHIQUES .....</b>	<b>76</b>

# LISTE DE FIGURE

---

FIGURE 1.1. FONCTION GLOBALE D'UN SYSTÈME .....	1
FIGURE 1.2. PRINCIPE D'AUTOMATISATION.....	2
FIGURE 1.3. STRUCTURE D'UN SYSTÈME AUTOMATISÉ .....	3
FIGURE 1.4. CONTACTEURS OU RELAIS (POUR LES MOTEURS ÉLECTRIQUES) .....	4
FIGURE 1.5. DISTRIBUTEURS (POUR LES VÉRINS) .....	4
FIGURE 1.6. LE SYSTÈME ET LE MILIEU EXTÉRIEUR .....	7
FIGURE 2.1. SITUATION DES ÉTAPES D'UN SYSTÈME AUTOMATISÉ.....	13
FIGURE 2.2. ACTIONS ASSOCIÉES AUX ÉTAPES.....	13
FIGURE 2.3. TRANSITION ASSOCIÉE UNE CONDITION LOGIQUE APPELÉE RÉCEPTIVITÉ .....	13
FIGURE 2.4. LIAISONS ORIENTÉES ENTRE LES ÉTAPES ET LES TRANSITIONS .....	14
FIGURE 2.5. CHRONOGRAMME DE L'ACTION CONTINUE.....	14
FIGURE 2.6. CHRONOGRAMME DE L'ACTION CONDITIONNELLE SIMPLE .....	14
FIGURE 2.6. CHRONOGRAMME DE L'ACTION RETARDÉE .....	15
FIGURE 2.7. CHRONOGRAMME D'ACTION DE DURÉE LIMITÉE .....	15
FIGURE 2.8. ACTION MAINTENUE SUR PLUSIEURS ÉTAPES.....	15
FIGURE 2.9. ACTION À EFFET MAINTENUE PAR UNE ACTION MÉMORISÉE .....	16
FIGURE 2.10. FRANCHISSEMENT D'UNE TRANSITION .....	17
FIGURE 2.12. DÉBUT DE SÉLECTION DE SÉQUENCE .....	17
FIGURE 2.13. FIN DE SÉLECTION DE SÉQUENCE.....	18
FIGURE 2.14. DÉBUT DE SÉQUENCES SIMULTANÉES .....	18
FIGURE 2.15. FIN DE SÉQUENCES SIMULTANÉES.....	18
FIGURE 2.16. SAUT D'ÉTAPES ET REPRISE DE SÉQUENCE.....	19
FIGURE 2.17. AIGUILLAGE ENTRE DEUX OU PLUSIEURS SÉQUENCES.....	19
FIGURE 2.18. PARALLÉLISME ENTRE DEUX OU PLUSIEURS SÉQUENCES .....	20
FIGURE 2.19. LIAISON ENTRE GRAFCETS.....	22
FIGURE 2.20. GRAFCET DE NIVEAU 1 ET 2.....	24
FIGURE 2.21. GRAFCET ET SONT DIAGRAMME ÉCHELLE .....	24
TAB 2.1. TABLE DE VÉRITÉ DE L'ÉTAPE N .....	26
TAB 2.2. TABLEAU DE KARNAUGH ASSOCIÉ.....	26
FIGURE 2.22. GRAFCET ET SONT DIAGRAMME ÉCHELLE .....	29
FIGURE 2.23. MATÉRIALISATION D'UN GRAFCET PAR BASCULE RS .....	29
FIGURE 2.24. MATÉRIALISATION D'UN DIVERGENCE SIMPLE EN ET PAR BASCULES RS .....	30
FIGURE 2.25. MATÉRIALISATION D'UN DIVERGENCE EN OU PAR BASCULES RS .....	31
FIGURE 2.26. MATÉRIALISATION D'UN CONVERGENCE EN ET PAR BASCULES RS .....	31
FIGURE 2.27. MATÉRIALISATION D'UN CONVERGENCE EN OU PAR BASCULES RS .....	31
FIGURE 2.28. SYSTÈME DE DEUX POMPES .....	32
FIGURE 2.29. GRAFCET CORRESPONDANT À CE SYSTÈME DE DEUX POMPES .....	32
FIGURE 2.30. GRAFCET CORRESPONDANT À P1 ET P2 MARCHENT À TOUR DE RÔLE .....	33
FIGURE.3.1. ARCHITECTURE D'UN API .....	34
FIGURE.3.2. ARCHITECTURE RÉELLE D'UN API S7-300.....	35
FIGURE.3.3. INTERFACE D'ENTRÉE D'UN API.....	36
FIGURE.3.4. INTERFACE DE SORTIE D'UN API .....	37
FIGURE.3.5. FONCTIONNEMENT DE L'INTERFACE D'ENTRÉE D'UN API .....	39
FIGURE.3.6. FONCTIONNEMENT DE L'INTERFACE D'ENTRÉE D'UN API .....	40
FIGURE.3.7. SCHÉMA DE PRINCIPE D'UN MODULE TOR INDUSTRIEL .....	40
FIGURE 4.1. LES TROIS FAMILLES DE MODES DE MARCHÉ ET D'ARRÊT .....	47
FIGURE 4.2. LES FAMILLES ET LES SOUS-FAMILLES DE PROCÉDURES. ....	49
FIGURE.4.3. REPRÉSENTATION D'UN RECTANGLE-ÉTAT .....	50
FIGURE.4.5. FEUILLE GEMMA .....	51
FIGURE.4.6. SÉLECTEUR DE MODES (MANUEL/AUTO) .....	53
FIGURE.4.7. LE RECTANGLE-ÉTAT F2 EST RETENU, LE RECTANGLE-ÉTAT F3 N'EST PAS RETENU .....	61
FIGURE.4.8. CONDITIONS D'ÉVOLUTION ENTRE MODES MARCHES / ARRÊTS .....	63
FIGURE.4.9. GEMMA MINIMAL D'UNE MACHINE SEMI-AUTOMATIQUE .....	63
FIGURE.4.10. GEMMA MINIMAL D'UNE MACHINE AUTOMATIQUE.....	63

---

FIGURE.4.11. GEMMA D'UNE MACHINE AUTOMATIQUE OU SEMI-AUTOMATIQUE EXIGEANT UNE MARCHE DE PRÉPARATION.....	64
FIGURE.4.12. GEMMA D'UNE MACHINE EXIGEANT UNE MARCHE DE PRÉPARATION ET UNE MARCHE DE CLÔTURE .....	64
FIGURE.4.13. GEMMA DE PRÉPARATION ET UNE MARCHE DE CLÔTURE ET AYANT UN ARRÊT DANS UN ÉTAT AUTRE QU'INITIAL .....	65
FIGURE.4.14. GEMMA D'UNE MACHINE AVEC ARRÊT D'URGENCE .....	65
FIGURE.4.15. GEMMA D'UNE MACHINE AVEC ARRÊT D'URGENCE ET REMISE EN ROUTE.....	66
FIGURE.4.16. GEMMA D'UNE MACHINE AVEC PRODUCTION TOUT DE MÊME .....	66
FIGURE.4.17. GEMMA D'UNE MACHINE AVEC MARCHES DE VÉRIFICATION DANS L'ORDRE.....	67
FIGURE.4.18. GEMMA D'UNE MACHINE AVEC MARCHES DE VÉRIFICATION DANS LE DÉSORDRE ET DANS L'ORDRE .....	67
FIGURE.4.19. RÉALISATION DU GRAFCET À PARTIR DU GEMMA.....	68
FIGURE.4.20. ELABORATION DU GRAFCET DE CONDUITE .....	69
FIGURE.4.21. GRAFCET DE NIVEAU 2 DE LA PRODUCTION NORMALE .....	70
FIGURE.4.22. LA MARCHE DE CLÔTURE SE FAIT SI UNE CONDITION « LA VARIABLE CLO ».....	70
FIGURE.4.23. LA CONDITION CLO DANS LES DEUX CAS, INACTIF ET ACTIF.....	70
FIGURE.4.24. GRAFCET DE NIVEAU 2 INITIAL .....	71
FIGURE.4.25. GRAFCET DE NIVEAU 2 INITIAL COMBINÉ AVEC LE GEMMA.....	71
FIGURE.4.26. GRAFCET DÉCOUPÉ EN TÂCHES .....	72
FIGURE.4.27. GEMMA GÉNÈRE UN GRAFCET DE NIVEAU SUPÉRIEUR .....	73
FIGURE.4.28. GRAFCET COMPORTE LE CAS DE L'ARRÊT D'URGENCE.....	73
FIGURE.4.29. ARRÊTS D'URGENCE PAR INHIBITION DES ACTIONS .....	74
FIGURE.4.30. ARRÊTS D'URGENCE PAR FIGEAGE.....	74

# Avant-propos

---

Ce polycopié est un support de cours « **Automatismes industriels** » destiné aux étudiants électrotechniciens de troisième année licence. L'objectif de ce cours est de maîtriser les notions et les outils de représentation graphiques de systèmes automatisés (Grafcet), installer et entretenir des éléments d'automatismes industriels et effectuer la programmation et la configuration des automates programmables. Les pré-requis concernant ses aspects nécessitant des connaissances de base en électronique numérique, en langages de programmation informatiques et en automatique des systèmes.

Le contenu de ce polycopié est conforme au programme du module " **Automatismes industriels** " proposé par le comité pédagogique national du domaine sciences et technologies pour l'harmonisation de l'offre de formation de licence académique de la spécialité électrotechnique de la filière électrotechnique.

Ce polycopié est organisé en quatre chapitres :

Le premier chapitre est entièrement dédié à **l'introduction aux systèmes automatisés**. Il présente la fonction globale d'un système, l'automatisation et structure des systèmes automatisés, pré-actionneurs, actionneurs, capteurs, classification des systèmes automatisés, spécification des niveaux du cahier des charges et l'outil de représentation des spécifications fonctionnelles.

Le chapitre 2 porte sur le Grafcet. Il présente les définitions et les notions de bases d'un grafcet, les règles d'établissement du GRAFCET, les transitions et liaisons orientées, les règles d'évolution, la sélection de séquence et séquences simultanées, l'organisation des niveaux de représentation et finalement la matérialisation d'un GRAFCET.

Le chapitre 3 est consacré à l'étude de l'**automate programmable industriel**. Le contenu de ce chapitre est formé par la structure interne et description des éléments d'un A.P.I, Choix de l'unité de traitement, choix d'un automate programmable industriel, les interfaces d'entrées-sorties, outils graphiques et textuels de programmation, mise en œuvre d'un automate programmable industriel et les principes des réseaux d'automates.

Le chapitre 4 est réservé à l'étude de « Guide d'Etude de Modes Marche et Arrêt (G.E.M.M.A) ». Il présente le concept et la structuration du GEMMA, les procédures de fonctionnement, les procédures d'arrêt et les procédures en défaillances, utilisation pratique du GEMMA et ses applications.

# Chapitre 1. Introduction aux systèmes automatisés

---

Le système automatisé est un ensemble d'éléments matériels et/ou humains (opérateur, pilote, conducteur, équipe,...), en interaction dynamique, organisés en fonction d'un but à atteindre. Ce système permet d'apporter une valeur ajoutée à des matières d'œuvre en les faisant passer d'un état initial donné à un état final souhaité, pour satisfaire un certain besoin dans un environnement donné.

## 1.1. Fonction globale

Pour satisfaire à sa fonction, un système automatisé réalise un certain nombre de tâches. Une tâche est une action bien précise qui exécute un travail sur une matière d'œuvre. Ce travail donne à la matière d'œuvre une valeur ajoutée. Dans un système technique, les tâches sont réalisées par la partie opérative et la coordination des tâches est effectuée par la partie commande. Autrement dit, c'est l'ensemble des opérations exécutées par un système et qui permet de faire passer la matière d'œuvre d'un état initial à un état final (Figure 1.1).

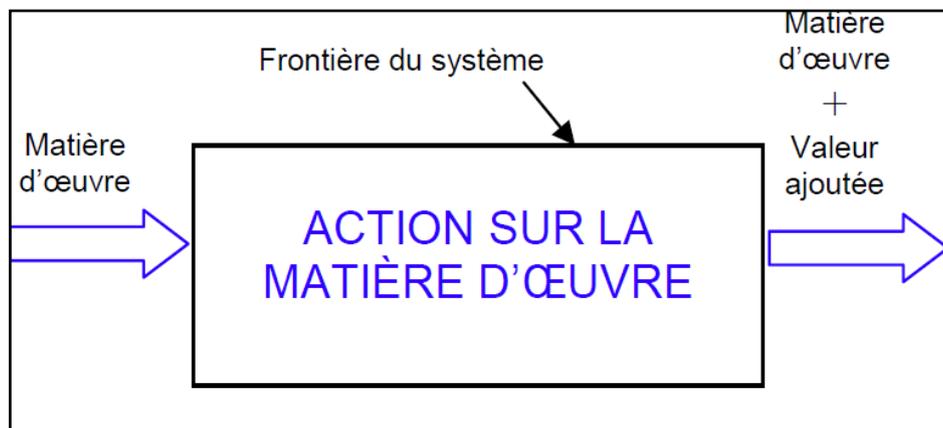


Figure 1.1. Fonction Globale d'un Système

De plus, un système de production est dit « industriel » si l'obtention de cette valeur ajoutée, pour un ensemble de matières d'œuvre donné, a un caractère reproductible et peut être exprimée et quantifiée en termes économiques.

### 1.1.1. Matière d'œuvre :

Une matière d'œuvre peut se présenter sous plusieurs formes. Par exemple :

- Produit : liquide, solide, gazeux,
- Énergie : électrique, thermique, mécaniques, etc.
- Information : physique, audiovisuel, etc.

### 1.1.2. Valeur ajoutée :

La valeur ajoutée à ces matières d'œuvre est l'objectif global pour lequel a été défini, conçu, réalisé puis éventuellement modifié le système. Cette valeur ajoutée est caractérisée par sa nature, sa quantité et sa qualité, à savoir :

- Une modification physique: conversion d'énergie, mécanique, etc.
- Un arrangement particulier: montage, assemblage, etc.
- Un prélèvement d'information : mesure, contrôle, etc.

#### Exemple 1.1

Système	Matières d'œuvre		Valeur ajoutée	Contexte
	Entrantes	Sortantes		
Centre de soins (système de santé)	malades	Individuels soignés	Soins, bonne santé	Sécurité sociale, moyens et personnels de santé

### 1.2. Automatisation

L'automatisation de la production consiste à transférer tout ou une partie des tâches de coordination, auparavant exécutées par des opérateurs humains dans un ensemble d'objets techniques appelé **partie commande**. La partie commande mémorise le savoir faire des opérateurs pour obtenir la suite des actions à effectuer sur les matières d'œuvre afin d'élaborer la valeur ajoutée. Elle exploite un ensemble d'informations prélevées sur la **Partie Opérative** pour élaborer la succession des ordres nécessaires pour obtenir les actions souhaitées.

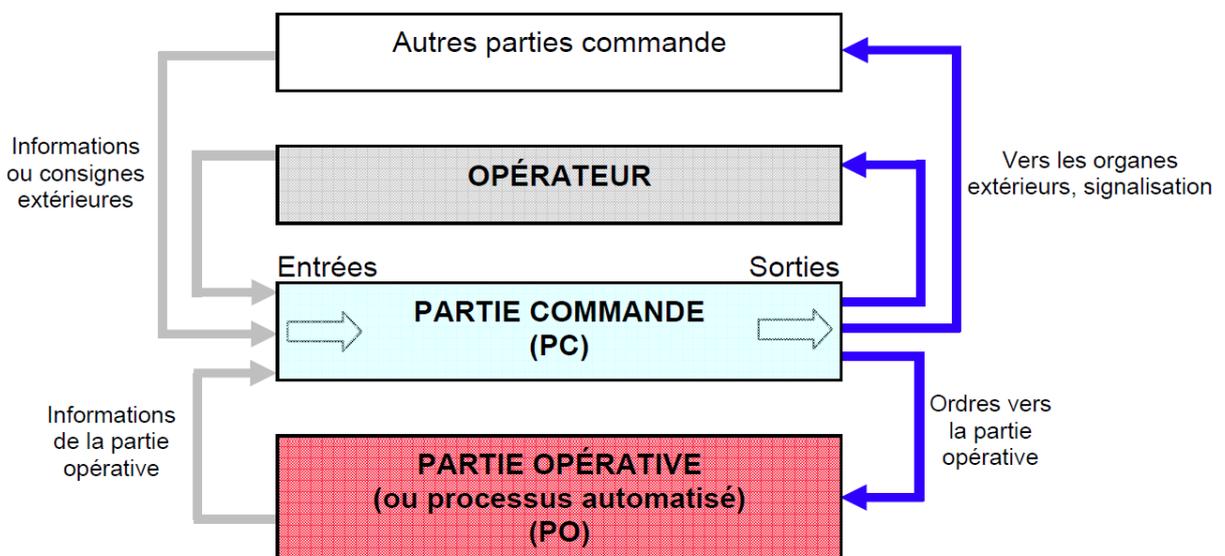


Figure 1.2. Principe d'automatisation

L'automatisation permet d'apporter des éléments supplémentaires à la valeur ajoutée. Ces éléments sont exprimables en termes d'objectifs:

- ✓ La recherche des coûts plus bas pour le produit par la réduction des frais de main d'oeuvre, d'économie d'énergie, d'économie de la matière, etc.
- ✓ La recherche d'une meilleure qualité du produit en limitant le facteur humain et multipliant les contrôles automatisés.
- ✓ L'amélioration de la flexibilité de la production.
- ✓ La suppression des travaux dangereux ou pénibles et l'amélioration des conditions de travail.
- ✓ La réalisation d'opérations impossibles à contrôler manuellement, par exemple des assemblages miniatures, des opérations très rapides, des coordinations complexes.

### 1.3. Structure d'un système automatisé

Un système de production est dit automatisé, lorsqu'il peut gérer de manière autonome un cycle de travail préétabli qui se décompose en séquences ou étapes. Les systèmes automatisés, utilisés dans le secteur industriel, possèdent une structure de base identique. Ils sont constitués de plusieurs parties plus ou moins complexes reliées entre elles que l'on nomme :

- Partie opérative (**PO**);
- Partie commande (**PC**);
- Partie relation (pupitre de dialogue) [**PR**]

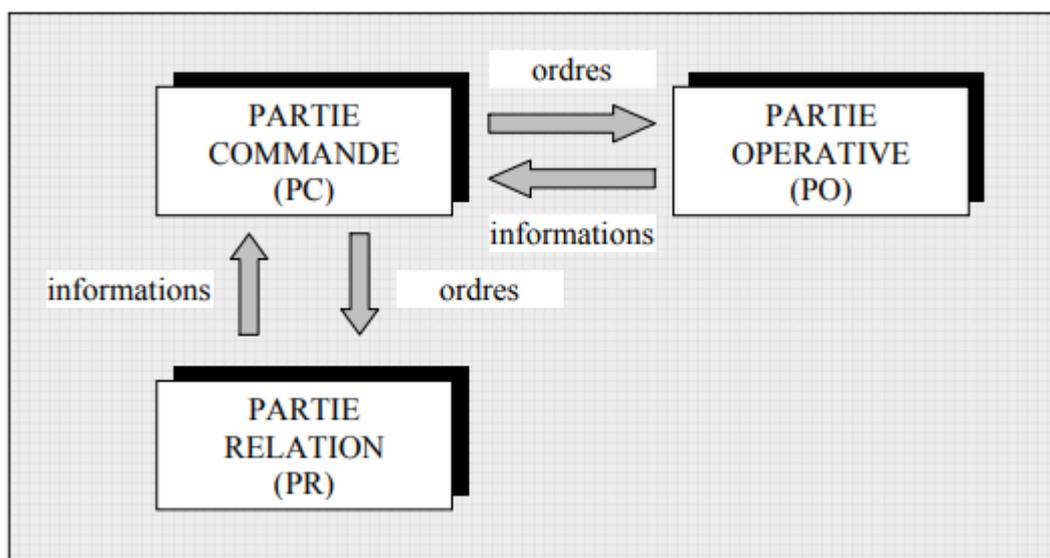


Figure 1.3. Structure d'un système automatisé

### 1.3.1. La partie opératives (PO) :

C'est l'ensemble des moyens techniques qui permettent d'apporter la valeur ajoutée aux matières d'œuvre en effectuant directement le processus de leur transformation. C'est la partie visible du système. Elle comporte les éléments mécaniques du mécanisme, ainsi la fonction « commander la puissance » a pour but d'effectuer une action ou une tâche sur la matière d'œuvre. Elle nécessite l'intervention d'un **actionneur**, et d'un **pré-actionneur**.

Le **pré-actionneur** peut être un contacteur (figure 1.4) ou un distributeur pneumatique (figure 1.5). Il envoie l'énergie aux actionneurs, il représente l'interface entre le traitement des informations et les actionneurs.



Figure 1.4. Contacteurs ou relais (pour les moteurs électriques)



Figure 1.5. Distributeurs (pour les vérins)

- **des actionneurs (vérins-moteurs)** qui ont pour rôle d'exécuter ces ordres.

Un actionneur est un objet technique qui convertit une énergie d'entrée non directement utilisable par les mécanismes agissant sur la matière d'œuvre en une énergie de sortie (très souvent mécanique) utilisable par ces mécanismes pour obtenir une action définie. Les actionneurs les plus courants sont les moteurs électriques et les vérins pneumatiques ou hydrauliques.

- **des capteurs** qui informent la partie de commande de l'exécution du travail. Ils existent sous différents types comme :
  - ✓ Capteurs mécaniques, pneumatiques ou électriques;
  - ✓ Capteurs magnétiques montés sur les vérins,
  - ✓ Capteurs pneumatiques à chute de pression.

Dans un système automatisé de production, ce secteur de détection représente le service de surveillance et renseignement du mécanisme. Il contrôle, mesure, surveille et informe la partie commande sur l'évolution du système.

### **1.3.2. La partie commande (PC) :**

C'est l'ensemble des moyens de traitement de l'information qui assurent le pilotage et la coordination des tâches du processus de transformation de la matière d'oeuvre. Ce secteur de l'automatisme gère dans la suite logique le déroulement ordonné des opérations à réaliser. Il reçoit des informations en provenance des capteurs situés dans la PO, et les restitue vers cette même PO en direction des pré-actionneurs (distributeurs). Sa constitution dépend du type de fonctionnement du système :

- **type séquentiel** : les opérations de transformation de la matière d'oeuvre s'enchaînent en une succession de séquences ; les informations véhiculées par la chaîne d'information sont de nature logique (variation en tout ou rien de leur état). La partie commande est alors constituée de composants logiques qui, regroupés, forment un **système logique de commande** (Automate programmable ou non).
- **type continu** : les informations véhiculées par la chaîne d'information sont de nature analogique (variation continue de leur état). La partie commande est alors constituée d'un système **asservi de commande** (asservissement, régulation).

Dans la suite on s'intéresse seulement à l'outil de description qui s'appelle GRAFCET (Graphe de Commande Étape et Transaction).

### **1.3.3. La partie Relation (PR)**

Ces constituants communiquent l'information. Soit par des bus d'information série ou parallèle dans un protocole machine - machine, soit par des écrans ou voyant. Sa complexité et sa taille dépendent de l'importance du système. Elle regroupe les différentes commandes nécessaires au bon fonctionnement du procédé : marche-arrêt, arrêt d'urgence, marche automatique, marche cycle/cycle,...

L'outil de description s'appelle « GEMMA » (Guide d'Étude des Modes de Marches et Arrêts).

#### **Exemple.1.2 : Gestion de Trafic**

La maquette de gestion de trafic est constituée de:



- Une partie opérative : les feux de carrefour à gérer en fonction du jour et de la nuit, des appels piétons, des choix prioritaires, etc...
- une partie commande : comportant un automate Télémécanique TSX Nano monté et câblé sur support plastique.
- un terminal de réglage et de programmation en langage booléen TFTX117.
- câbles de raccordement.

#### **1.4. Classification des systèmes automatisés :**

Les systèmes automatisés peuvent être classés suivant différents critères tels que :

- le domaine d'application ;
- la nature de la matière d'oeuvre ;
- la nature des flux ;
- les critères technico-économiques.

##### **1.4.1. Classification suivant le domaine d'application :**

Les systèmes automatisés peuvent être classés suivant le domaine d'application, à savoir :

- Production continue (raffineries, centrales, industries papetières,...) ;
- Production discontinue (lavage de vaisselle, fabrication de pièces,...) ;
- Traitement de l'information (chaîne haute fidélité, ordinateur,...).

##### **1.4.2. Classification suivant la nature de la matière d'œuvre**

Les systèmes automatisés peuvent être classés suivant la nature de la matière d'œuvre, à savoir :

- Matière : aliment, passager, pièce, ... ;
- Energie : électrique, pneumatique, hydraulique,... ;
- Information : signal électrique, signal optique.

##### **1.4.3. Classification suivant la nature des flux**

Il s'agit des échanges entre le système et le milieu extérieur:

- flux des matières d'oeuvre ;

- flux des énergies ;
- flux des informations ;
- flux des déchets et nuisances.

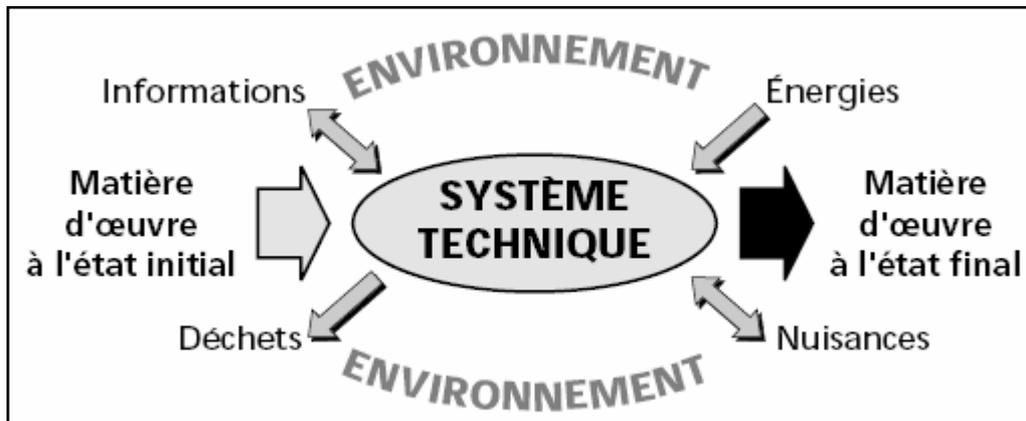


Figure 1.6. le système et le milieu extérieur

#### 1.4.4. Classification suivant les critères technico-économiques

- **Valeur ajoutée** : Résultat de l'évaluation de la différence entre l'état initial et l'état final de la matière d'œuvre.
- **Durée de vie** : Durée pendant laquelle le système a accompli la fonction qui lui a été assignée.
- **Maintenabilité** : Dans des conditions données d'utilisation pour lesquelles il a été conçu, aptitude du système à être maintenu ou rétabli dans un état dans lequel il peut accomplir une fonction requise, lorsque la maintenance est accomplie dans des conditions données, avec des procédures et des moyens prescrits.
- **Coût** : Divers coûts peuvent être considérés, tels que : coût d'achat, coût d'utilisation,...
- **Fiabilité** : Aptitude du système à accomplir une fonction requise dans des conditions données pendant un temps donné.
- **Quantité** : Système unique, ou en quelques exemplaires, ou de diffusion moyenne, grande ou très grande.

#### 1.5. Spécification des niveaux du cahier des charges :

##### 1.5.1. Définition d'un cahier des charges fonctionnel

Un cahier des charges fonctionnel est un document qui présente de manière détaillée et structurée les spécifications, les services à rendre, les contraintes d'un produit (bien qu'il

puisse également s'agir d'un service, d'un processus, d'une prestation intellectuelle, d'un logiciel ou encore d'un système d'information.)

Le cahier des charges fonctionnel permet de cadrer le projet : il est d'ailleurs rédigé à la fin de la phase de lancement du projet, au moment de sa validation.

Il est au centre des négociations et des échanges entre les fournisseurs et le donneur d'ordres : c'est une référence qui permet d'accompagner le projet du début à la fin et d'assurer que les attentes du demandeur soient respectées dans l'élaboration du produit.

### **1.5.2. Objectifs du cahier des charges fonctionnel**

Le cahier des charges fonctionnel présente de nombreux intérêts pour la conduite d'un projet, notamment :

- Il instaure un dialogue structuré et compétitif avec les fournisseurs, car il énonce clairement une question et exige une réponse adaptée.
- Il offre au fournisseur une ample marge de manoeuvre pour qu'il puisse proposer des solutions innovantes et originales pour répondre aux besoins, tout en prenant en compte les contraintes (prix / qualité attendue / délais...)
- Il permet d'évaluer plus rapidement les éventuels écarts entre les solutions proposées et le besoin réel, facilitant l'analyse et la comparaison entre les différentes propositions.
- Il facilite la vérification et l'évaluation des résultats obtenus en se basant sur les références qu'il définit.

### **1.5.3. Méthodologie du cahier des charges fonctionnel**

La rédaction du cahier des charges fonctionnel découle de plusieurs autres processus qui doivent être mis en oeuvre en amont.

#### ***1.5.3.1. L'étude d'opportunité***

L'étude d'opportunité consiste à étudier le contexte du projet et à définir les principaux besoins pour vérifier s'ils sont en phase avec les attentes de l'utilisateur. Elle permet d'évaluer rapidement la viabilité du projet.

#### ***1.5.3.2. L'étude de faisabilité***

L'étude de faisabilité vise à évaluer la viabilité du projet sur plusieurs plans, notamment :

- Économique ;
- Technique ;
- Organisationnel.

Elle permet d'estimer et d'anticiper les coûts, les délais et le ROI (retour sur investissement) probable du projet. On y ajoute généralement des études de scénario afin de prévenir les risques et menaces éventuels.

#### ***1.5.3.3. L'Analyse Fonctionnelle du Besoin***

L'analyse fonctionnelle vise à déterminer les « fonctions de service » du produit, en amont de sa réalisation. Pour ce faire :

- Elle examine les points de vue des différentes parties concernées ;
- Elle se projette dans la durée, prenant en considération les différentes étapes du cycle de vie : mise en place en amont, usage du produit, maintenance, entretien, fin de vie...
- Elle n'exprime pas les moyens à mettre en oeuvre, mais plutôt les résultats recherchés.

#### ***1.5.3.4. L'Expression Fonctionnelle du Besoin***

L'expression fonctionnelle du besoin exploite les résultats de l'AFB afin de :

- Structurer rigoureusement et logiquement l'information pour aider à la prise de décision ;
- Permettre la création d'un produit performant et parfaitement adapté aux emplois et aux services voulus ;
- Servir de référence du besoin du client durant la réalisation du produit ou service.

Elle se structure généralement en quatre grandes parties :

- La définition globale du besoin ;
- La consolidation des besoins et la définition des éléments stratégiques ;
- La définition des principes et concepts retenus ;
- La description des contraintes à respecter et des fonctions de service à assurer.

### **1.6. Outils de représentation des spécifications fonctionnelles :**

Il existe depuis longtemps des outils de représentations graphiques pour la partie opérative, tel le dessin industriel, dont la représentation normalisée est soumise à des règles précises évitant toute interprétation erronée.

Il manquait un outil équivalent pour définir les parties commande : c'est le rôle qui est imparti au GRAFCET, En effet, les différents travaux effectués ces dernières années dans le domaine des automatismes industriels ont conduit à une représentation graphique des spécifications fonctionnelles d'un cahier des charges, dont la formulation est indépendante de toute technologie de réalisation, que celle-ci soit câblée ou programmée. Cette représentation, est fondée sur les notions d'étapes, de transitions et de réceptivités qui simplifient la synthèse

d'un automatisme en tenant compte du fait que, parmi le grand nombre d'informations présentes à un instant donné, peu sont significatives. Le GRAFCET permet de visualiser de façon particulièrement claire toutes les évolutions du système.

# Chapitre 2 : Le Grafcet

---

## 2.1. Définition et notions de bases

Dans ce chapitre, on étudiera un outil de modélisation graphique appelé le GRAFCET (GRaphe Fonctionnel de Commande Étape/Transition). Il est fréquemment utilisé pour la mise en œuvre des automates programmables industriels (API)

### 2.1.1. Définition

Le GRAFCET est un langage graphique pour décrire, étudier, réaliser et exploiter les automatismes. Il est composé d'un ensemble d'étapes et de transitions représentant le déroulement du cycle de l'automatisme.

- 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.

Le GRAFCET comprend :

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

### 2.1.2. Description du GRAFCET

La description du comportement attendu d'un automatisme 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* : C'est un graphe qui décrit le fonctionnement global du système. Il traduit le cahier des charges sans préjuger de la technologie adoptée. Il

permet de dialoguer avec des personnes non spécialistes (fournisseurs, décideurs ...) Son écriture, en langage clair, permet donc sa compréhension par tout le monde.

- *Un point de vue Partie Opérative* : Dans ce type de grafcet on spécifie la technologie de la partie opérative ainsi que le type de ses informations reçues (ordres) et envoyées (comptes-rendus). L'observateur de ce point de vue étant un spécialiste de la partie opérative, la partie commande ne l'intéresse que par ses effets.

- *Un point de vue Partie Commande* : Ce grafcet est établi en spécifiant la technologie des éléments de dialogue :

- ✓ entre PC et PO ;
- ✓ entre PC et opérateur ;
- ✓ entre PC et autre système.

C'est un grafcet établi par un spécialiste, c'est la version qui lui permet d'établir les équations et éventuellement les schémas de réalisation (électrique, pneumatique)

➤ **Les 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.

### **2.1.3. Les avantages :**

- Il est indépendant de la matérialisation technologique,
- Il traduit de façon cohérente le cahier des charges,
- Il est bien adapté aux systèmes automatisés.

## **2.2. Règles d'établissement du GRAFCET**

### **2.2.1. 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<sub>i</sub>*. 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.

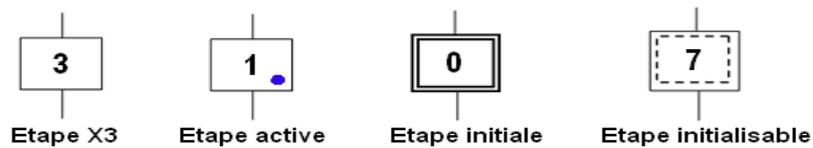


Figure 2.1. Situation des étapes d'un système automatisé

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

### 2.2.2. 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 grafquets. Mais on peut rencontrer aussi une même action associée à plusieurs étapes ou une étape vide (*sans action*).

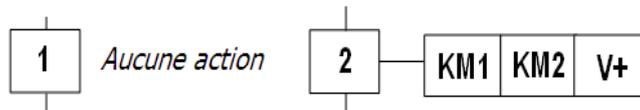


Figure 2.2. Actions associées aux étapes

### 2.2.3. 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.

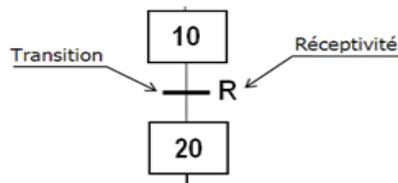


Figure 2.3. Transition associée une condition logique appelée réceptivité

La réceptivité 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 grafquets* : d'autres grafcet pour la liaison entre grafquets ou de l'état courant des étapes du grafcet (les Xi),
- ✓ *d'autres systèmes* : dialoguent entre systèmes, ...

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

## 2.2.4. 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.

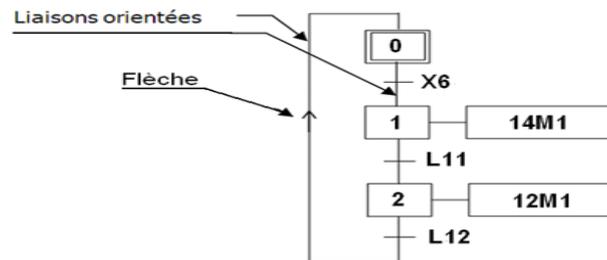


Figure 2.4. Liaisons orientées entre les étapes et les transitions

## 2.2.5. 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.

### 2.2.5.1. Actions continues :

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



Figure 2.5. Chronogramme de l'action continue

### 2.2.5.2. 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:

#### Action conditionnelle simple : Type C

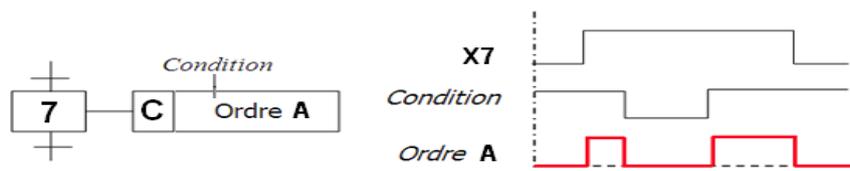


Figure 2.6. Chronogramme de l'action conditionnelle simple

#### 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.

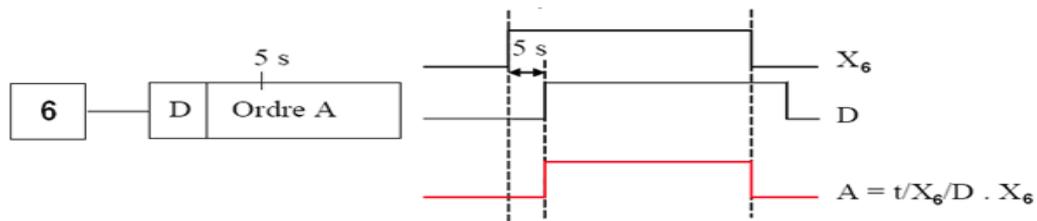


Figure 2.6. Chronogramme de l'action retardée

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

**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.

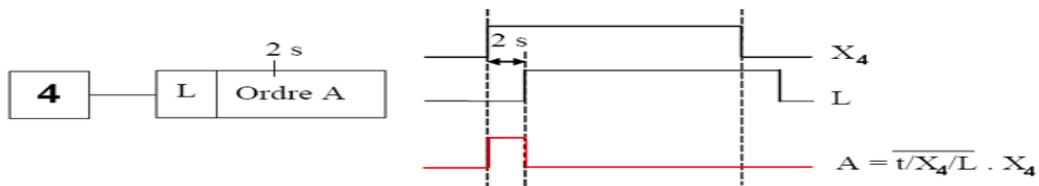


Figure 2.7. Chronogramme d'action de durée limitée

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

**2.2.5.3. 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* ).

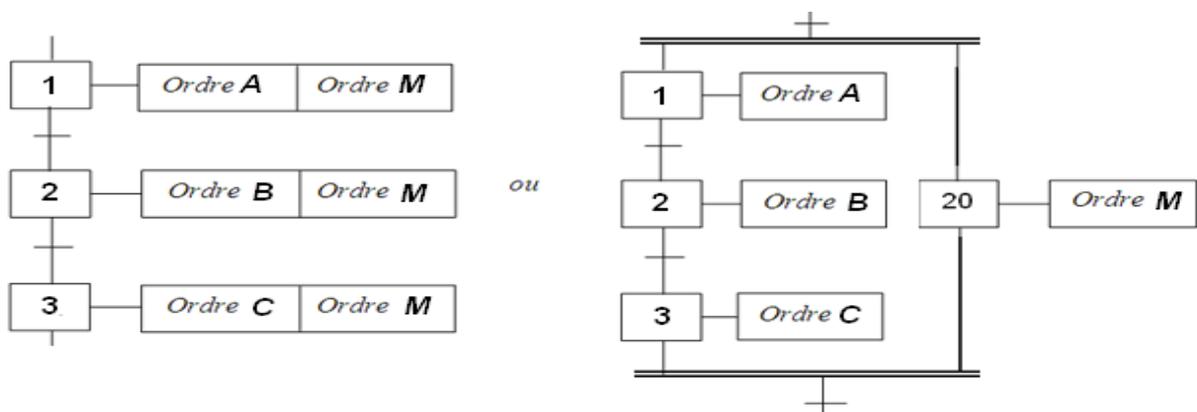


Figure 2.8. Action maintenue sur plusieurs étapes

#### 2.2.5.4. 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.

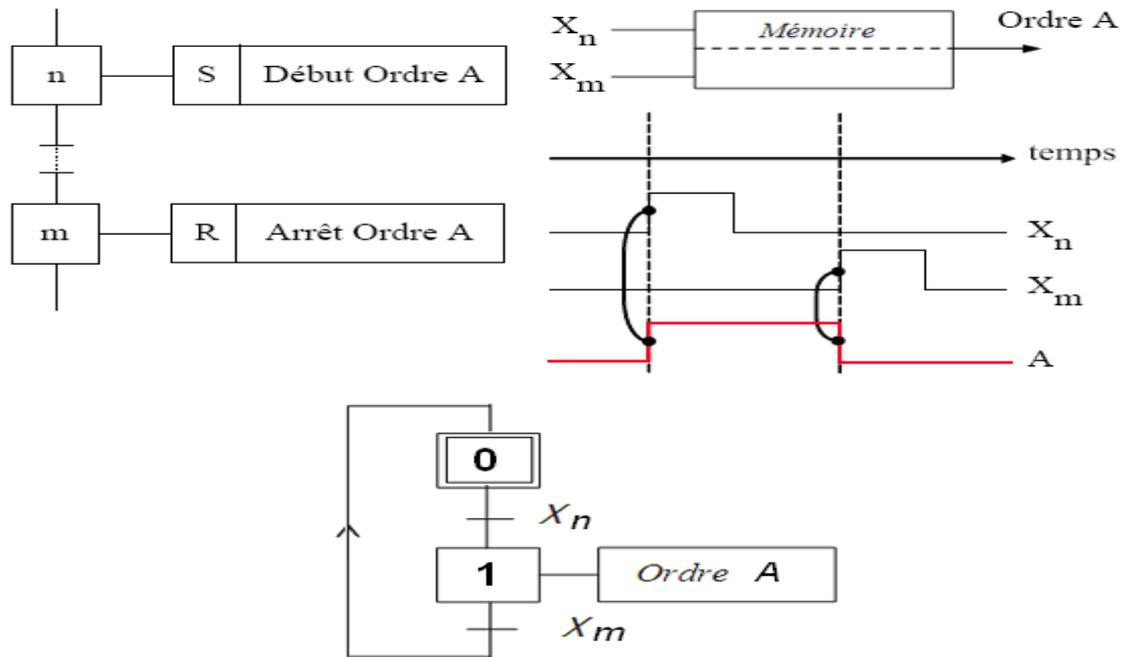


Figure 2.9. Action à effet Maintenue par une Action Mémorisée

### 2.3. Règles d'évolution d'un GRAFCET

#### Règle N°1 : Condition initiale.

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

#### 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.

#### 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.

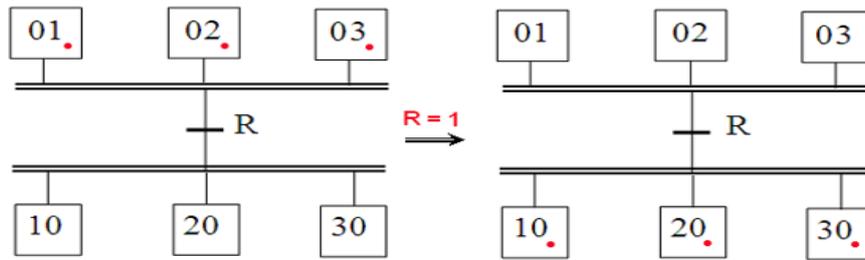


Figure 2.10. franchissement d'une transition

**Règle N°4 : Franchissement simultané.**

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

**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).

**2.4. Sélection de séquence et séquences simultanées :**

Le méta-modèle grafcet présente deux structures particulières, la sélection de séquence et les séquences simultanées.

**2.4.1. Sélection de séquence**

La sélection de séquence dans un grafcet permet de choisir une suite d'étapes plutôt qu'une autre. Cette structure est composée d'une seule étape en amont et de plusieurs transitions en aval qui permettront le choix de la séquence. Elle se représente à l'aide d'un simple trait horizontal. La fin d'une sélection de séquence permet la reprise d'une séquence unique.

*Exemples*

- début de sélection de séquence :

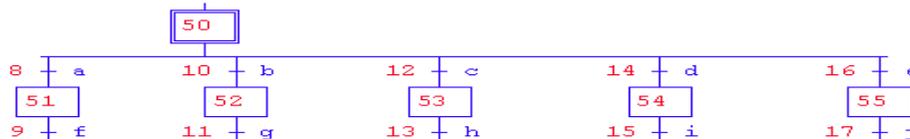


Figure 2.12. début de sélection de séquence

- fin de sélection de séquence :

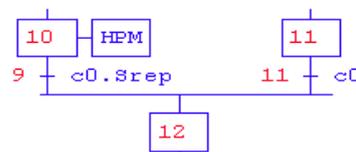


Figure 2.13. fin de sélection de séquence

### 2.4.2. Séquences simultanées

Lorsque l'on souhaite réaliser plusieurs séquences simultanément on utilise cette structure. Elle est composée d'une seule étape et d'une seule transition en amont qui permet de déclencher simultanément plusieurs séquences d'étapes. Elle se représente à l'aide d'un double trait horizontal. A la fin d'une série de séquences simultanées on retrouve, en général, un double trait suivi d'une seule transition. Attention, c'est le déclenchement des étapes situées immédiatement après le double trait qui est simultanément et non l'évolution des séquences. L'évolution de chacune des séquences d'étapes dépendra des conditions d'évolution et des actionneurs utilisés sur la Partie Opérative (compteur, temporisations, états de capteurs,...).

#### Exemples :

Début de séquences simultanées :

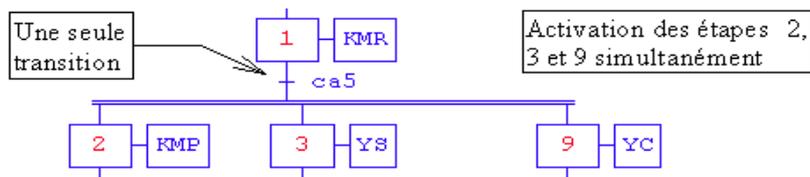


Figure 2.14. Début de séquences simultanées

Fin de séquences simultanées :

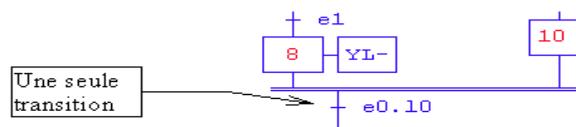


Figure 2.15. Fin de séquences simultanées

### 2.4.3. 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.

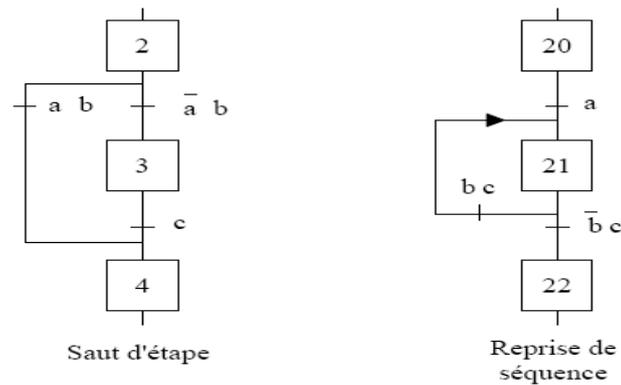


Figure 2.16. Saut d'étapes et reprise de séquence

#### 2.4.4. Aiguillage entre deux ou plusieurs séquences (Divergence en OU)

On dit qu'il y a Aiguillage ou divergence en OU lorsque le grafctet 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 grafctet converge vers une seule séquence.

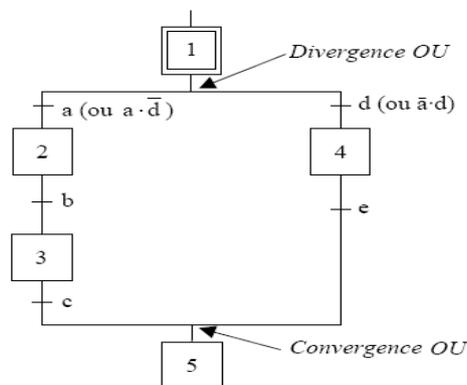


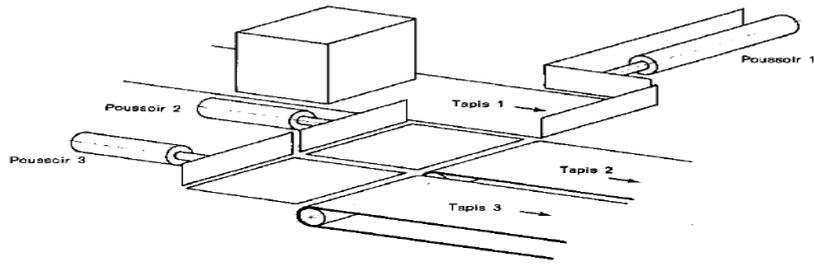
Figure 2.17. Aiguillage entre deux ou plusieurs séquences

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

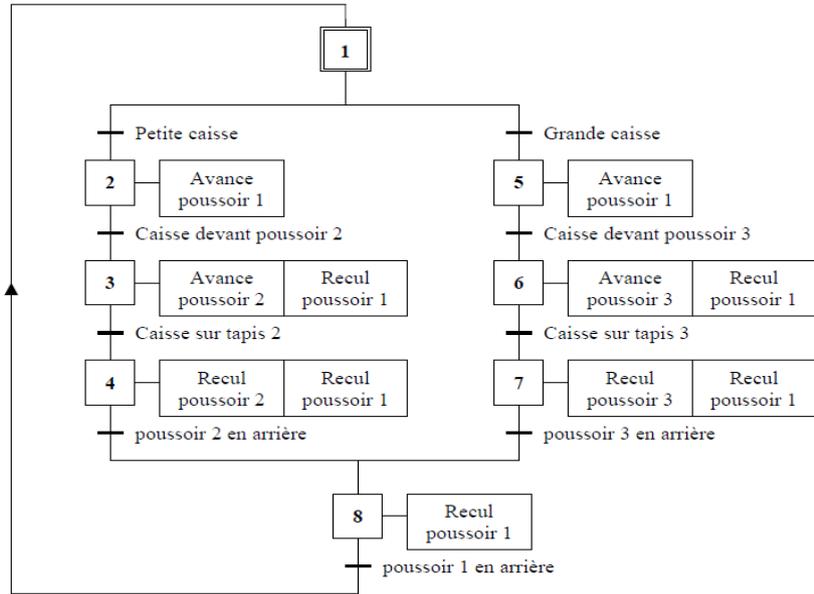
##### Exemple.2.1 : Tri de caisses

Un dispositif automatique destiné à trier des caisses de deux tailles différentes se compose d'un tapis amenant les caisses, de trois poussoirs et de deux tapis d'évacuation.

Le poussoir P1 pousse les petites caisses devant le poussoir P2 qui à son tour les transfère sur le tapis 2, alors que les grandes caisses sont poussées devant le poussoir P3, ce dernier les évacuant sur le tapis 3. Pour effectuer la sélection des caisses, un dispositif de détection placé devant le poussoir P1 permet de reconnaître sans ambiguïté le type de caisse qui se présente.



**Solution :**



**2.4.5. 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.

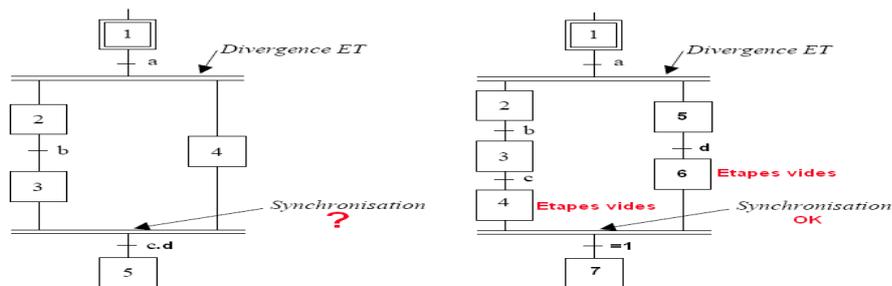
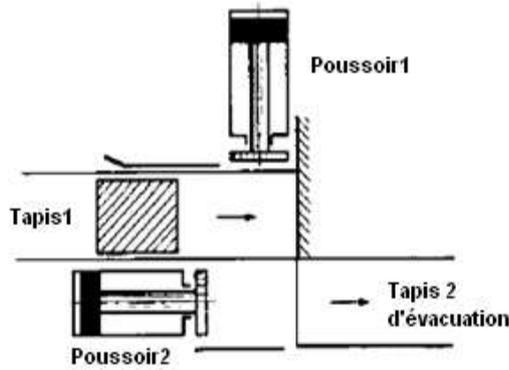


Figure 2.18. Parallélisme entre deux ou plusieurs séquences

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

**Exemple.2.2 : Transfert de pièces**

Un dispositif effectuant le transfert de pièces sur deux tapis différents est représenté par la figure ci- dessous. Dès qu'une pièce se présente devant le poussoir 1, celui-ci la repousse devant le poussoir 2 qui, situé perpendiculairement, transfère cette pièce sur le tapis d'évacuation.



Donner le Grafcet fonctionnel décrivant le fonctionnement du dispositif effectuant le transfert de pièces.

**Solution**

*Capteurs et actionneurs ( i = 1 ou 2 )*

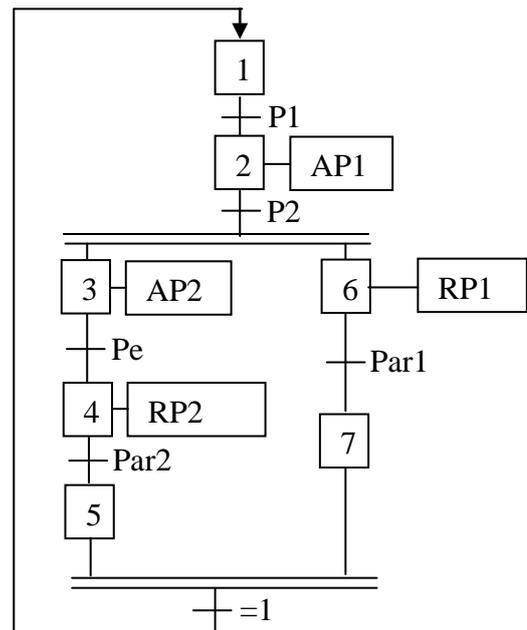
AP<sub>i</sub> : avance poussoir *i*

RP<sub>i</sub> : recul poussoir *i*

P<sub>i</sub> : pièce devant poussoir *i*

Par<sub>i</sub> : poussoir *i* en arrière

Pe : pièce évacuée



#### 2.4.6. Liaison entre grafquets :

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

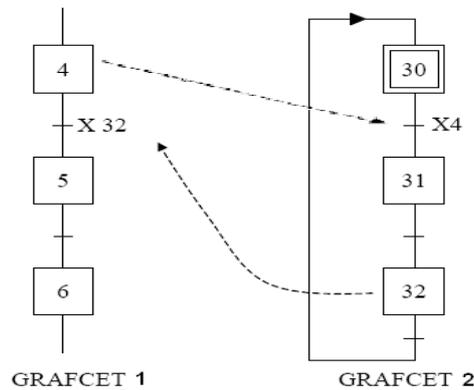


Figure 2.19. Liaison entre grafquets

#### 2.5. Organisation des niveaux de représentation

La description du fonctionnement attendu d'un système automatisé (SA) se fait par succession de trois points de vue. Le point de vue "système" est le premier décrit, il est le plus proche du cahier des charges fonctionnelles. Le point de vue "Partie Opérative" permet de se "rapprocher" du système automatisé et de définir les constituants de la Partie Opérative ainsi que de les interfaçages d'entrée et de sortie. Le troisième est le point de vue "Partie Commande" qui permettra à terme de générer le programme automate.

Les principaux GRAFCETS que l'on peut trouver sont :

**GRAFCET de surveillance :** (de sécurité) ce GRAFCET décrit l'ensemble des procédures de sécurité du système, c'est le GRAFCET hiérarchiquement le plus important. L'arrêt d'urgence et les procédures de mise en route sont décrits dans ce GRAFCET.

**GRAFCET de conduite :** (ou GRAFCET des Modes de Marches) ce GRAFCET décrit l'ensemble des procédures de Marches (auto, Cycle/Cycle, Manuel,...) et des arrêts normaux.

**GRAFCET de maintenance :** Précise les procédures d'intervention de l'opérateur et de réglage de la partie opérative.

**GRAFCET de Production** : ce GRAFCET est le niveau de description du fonctionnement normal de l'automatisme. Ce GRAFCET est en général décomposé en plusieurs tâches représentant les différentes fonctions de l'automatisme.

### **2.5.1. Les deux niveaux de représentation.**

Pour aborder de façon progressive l'étude d'un automatisme, l'analyse GRAFCET est divisée en deux niveaux. Le premier niveau s'attarde aux spécifications fonctionnelles. Le second aux spécifications technologiques.

#### **➤ Le GRAFCET de niveau 1**

Lors de l'analyse des spécifications fonctionnelles, le premier souci de l'automaticien est de comprendre le fonctionnement de l'automatisme. Il faut qu'il soit en mesure d'identifier le comportement de la Partie Commande par rapport à la Partie Opérative.

Pour faciliter ce premier niveau d'analyse, il ne faut pas se soucier de la technologie des actionneurs et des capteurs. Le GRAFCET de niveau 1 permet donc de représenter la séquence de fonctionnement souhaitée. La description des actions et de la séquence de l'automatisme est littérale.

Le GRAFCET de niveau 1 permet d'identifier les fonctions que doit remplir l'automatisme. Pour chacune de ces fonctions, il faut déduire quelles sont les actions à faire, les informations assurant que les actions soient complétées et les précautions à prendre du point de vue sécurité, indépendamment de la matérialisation technologique.

#### **➤ Le GRAFCET de niveau 2**

Lors de l'analyse des spécifications technologiques, l'automaticien utilisera l'analyse faite avec le GRAFCET de niveau 1 pour choisir les actionneurs et les capteurs nécessaires pour générer les actions et obtenir les informations nécessaires pour remplir les fonctions.

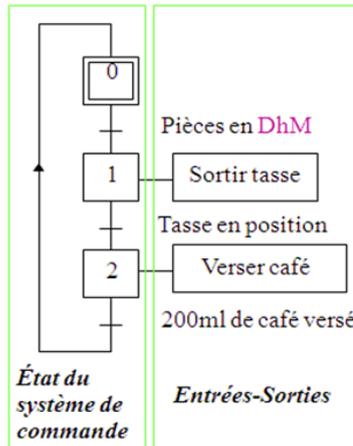
Le choix technologique est donc fait à cette étape. Donc le GRAFCET de niveau 2 est celui qui prend en compte la technologie des capteurs et actionneurs. Il pourrait mener à la programmation d'un automate ou à un séquenceur câblé. En pratique, ce GRAFCET sera ultérieurement modifié pour tenir compte des spécifications opérationnelles.

En effet, les GRAFCET de niveau 1 et de niveau 2 ne s'attardent qu'au fonctionnement normal de l'automatisme. Dans ce fonctionnement normal, il est assumé que l'automatisme ne manquera jamais de matière première, ne subira jamais d'arrêt d'urgence, ne sera jamais

défaillant. Donc les divers modes de marches et d'arrêts ne sont pas pris en compte. Ces modes sont introduits par l'outil méthode « GEMMA ».

**Exemple :**

**GRAFCET de niveau 1**



**GRAFCET de niveau 2**

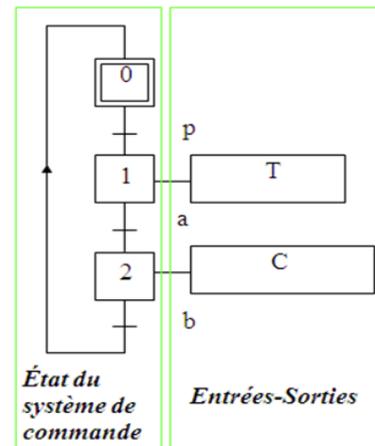


Figure 2.20. GRAFCET de niveau 1 et 2

**2.6. Mise en équation d'un grafcet :**

Malheureusement, ce ne sont pas tous les automates qui se programment en GRAFCET directement. Mais, généralement ils peuvent être programmés en « diagramme échelle » (ou LADDER). Il faut donc pouvoir transformer le GRAFCET qui est la meilleure approche qui existe pour traiter les systèmes séquentiels en « diagramme échelle » qui est le langage le plus utilisé par les automates.

Pour montrer comment le GRAFCET se transforme en diagramme échelle, supposons une suite de trois étapes tel que montré ci-dessous :

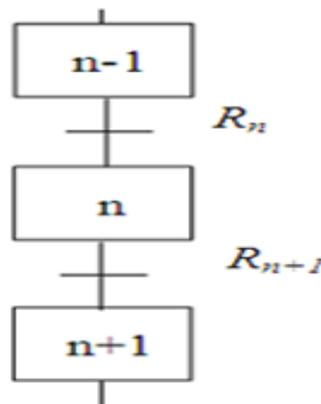


Figure 2.21. GRAFCET et sont diagramme échelle

Pour trouver l'équation logique de  $X_n$ , la variable logique représentant l'étape « n », il faut appliquer les règles du GRAFCET. L'étape « n » s'activera lorsque la réceptivité  $R_n$  sera

franchie. Cette réceptivité est franchie si l'étape «  $n - 1$  » est active et si la condition logique  $R_n$  est vraie. L'étape «  $n$  » s'activera alors et désactivera l'étape «  $n - 1$  ». L'équation logique de la mise à 1 de  $X_n$  sera :  $X_{n-1} R_n$ .

L'étape «  $n$  » se désactivera lorsque la réceptivité  $R_{n+1}$  sera franchie. Lorsque le franchissement se fera, l'étape «  $n + 1$  » s'activera et l'étape «  $n$  » se désactivera. L'équation logique de la mise à 0 de  $X_n$  sera :  $X_{n+1}$

L'équation logique de l'étape  $X_n$  est :

$$X_n = (X_{n-1} R_n + X_n). \bar{X}_{n+1}$$

Avec,

$X_{n-1}$  : Étape ( $n - 1$ ) est active.

$R_n$  : Réceptivité ( $n$ ) est vraie.

$X_n$  : Mémorisation de l'étape ( $n$ ).

$\bar{X}_{n+1}$  : Étape ( $n + 1$ ) est non active.

### 2.6.1. 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.

D'après la règle 2 du grafctet, la condition d'activation de l'étape  $n$  donne :

$$CAX_n = X_{n-1} R_n$$

D'après la règle 3 du grafctet, la condition de désactivation de l'étape  $n$  donne :

$$CDX_n = X_n R_{n+1} = X_{n+1}$$

Si la  $CA$  et la  $CD$  de l'étape  $n$  sont fausses, l'étape  $n$  reste dans son état (effet mémoire).

L'état de  $X_n$  à l'instant ( $t + \delta t$ ) dépend de l'état précédent de  $X_n$  à l'instant  $t$ .

On peut alors écrire la table de vérité de l'étape  $n$  :  $X_n$

$X_n$	$CAX_n$	$CDX_n$	$X_n(t + \delta t)$	Remarque
0	0	0	0	L'étape reste inactive (effet mémoire)
0	0	1	0	L'étape reste inactive
0	1	0	1	Activation de l'étape
0	1	1	1	Activation et désactivation =activation
1	0	0	1	L'étape reste active (effet mémoire)
1	0	1	0	Désactivation de l'étape
1	1	0	1	L'étape reste active
1	1	1	1	Activation et désactivation =activation

Tab 2.1. Table de vérité de l'étape n

Tableau de Karnaugh associé :

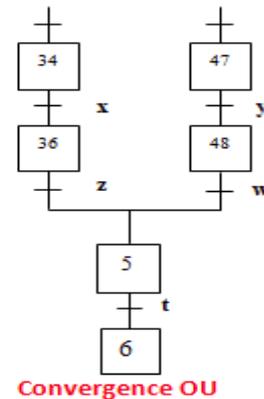
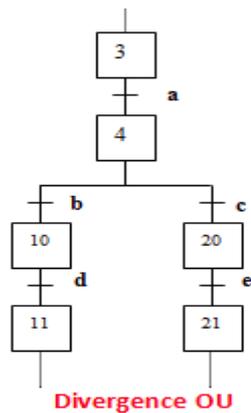
$X_n \backslash CAX_n CDX_n$	00	01	11	10
0	0	0	1	1
1	1	0	1	1

Tab 2.2. Tableau de Karnaugh associé

L'équation de  $X_n$  est :

$$\begin{cases} X_n = CAX_n + \overline{CDX_n} \cdot X_n \\ \text{ou} \\ X_n = X_{n-1}R_n + \overline{X_{n+1}} \cdot X_n \end{cases}$$

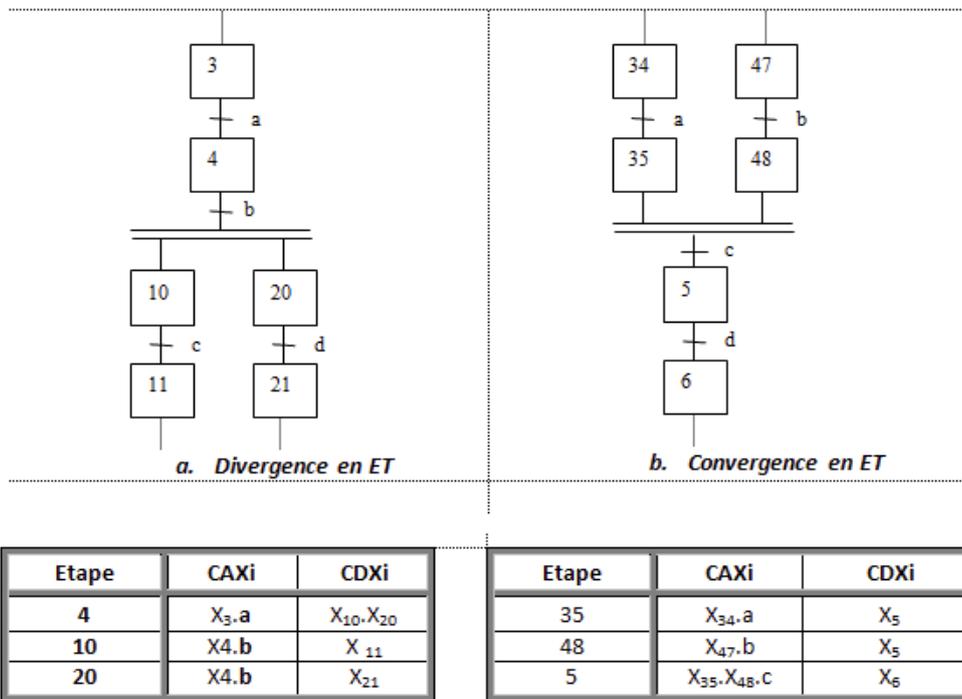
### 2.6.2. Choix de séquence :



Etape	CAXi	CDXi
4	$X_{3.a}$	$X_{10+X_{20}}$
10	$X_{4.b}$	$X_{11}$
20	$X_{4.c}$	$X_{21}$

Etape	CAXi	CDXi
36	$X_{34.X}$	$X_5$
48	$X_{47.Y}$	$X_5$
5	$X_{36.Z+X_{48.W}}$	$X_6$

### 2.6.3. Séquences parallèles



### 2.6.4. Gestion des modes Marche/Arrêt et Arrêt d'Urgence

A l'état initiale du GRAFCET, les étapes initiales sont activées par contre les autres étapes sont désactivées.

On introduit une variable Init telle que :

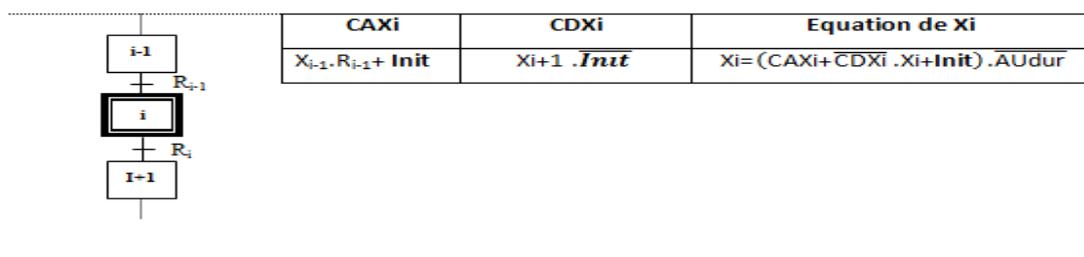
- Init=0 : Mode MARCHE (déroulement du cycle)
- Init= 1 : Mode ARRÊT (initialisation du grafcet)

On introduit deux variables d'Arrêt d'urgence AUdur (Arrêt d'Urgence dur) et AUdoux (Arrêt d'Urgence doux) telles que :

- AUdur= 1 : Désactivation de toutes les étapes.
- Audoux =1 : Désactivation des actions, les étapes restent actives !!

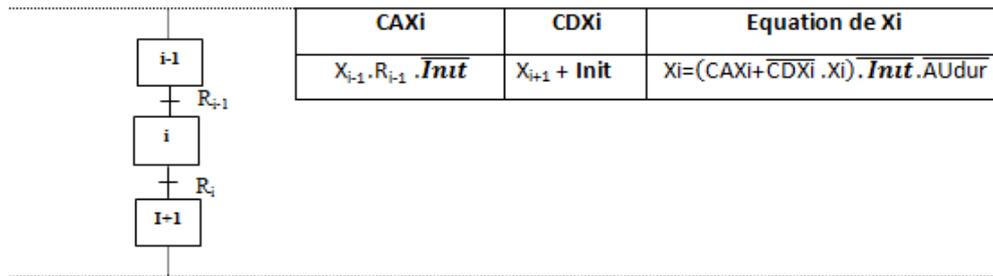
#### ETAPE initiale

L'équation d'une étape initiale devient alors :



## ETAPE NON initiale

L'équation d'une étape NON initiale devient alors :



L'équation des actions :



## 2.7. Matérialisation d'un GRAFCET

Un grafcet est constitué par un ensemble d'étapes ; à chacune de ces étapes, on associe une variable  $X_i$  qui est égale à "1" si l'étape est active et à "0" si l'étape est inactive.

On matérialisera cette variable  $X_i$  par la variable de sortie  $Q_i$  d'une bascule. La synthèse de l'automate est alors le problème de calcul des entrées ( $R_i, S_i$  ou  $J_i, K_i$ ) de ces bascules, et des sorties de l'automate en fonction des  $X_i$ .

### 2.7.1. Cas d'un grafcet linéaire

Il suffit d'utiliser une bascule RS par étape. Une étape est activée si l'étape précédente est active et que la réceptivité d'entrée est vraie. Dans le cas d'un grafcet linéaire, on désactivera une étape quand la suivante est active.

Soit le grafcet de la figure 2.21:

On peut gérer de différentes manières l'étape initiale. Dans la plupart des cas, le plus simple est d'utiliser des bascules se mettant à 0 à la mise sous tension, et d'initialiser l'automatisme à l'aide d'un bouton qu'on notera "Init", qui peut également servir à réinitialiser le grafcet en cours de fonctionnement sans éteindre le système.

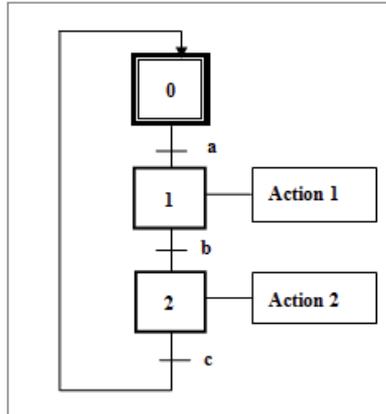


Figure 2.22. GRAFCET et son diagramme échelle

L'étape 1 s'active si l'étape 0 est active et la réceptivité a est vraie  $\rightarrow S_1 = Q_0 \cdot a$ . Tout le temps qu'elle est active, la sortie Action1 est active (égale à 1). Elle est désactivée quand la réceptivité de sortie (b) est vraie, mais il faut attendre que l'étape 2 soit active. Elle peut être également désactivée par Init.

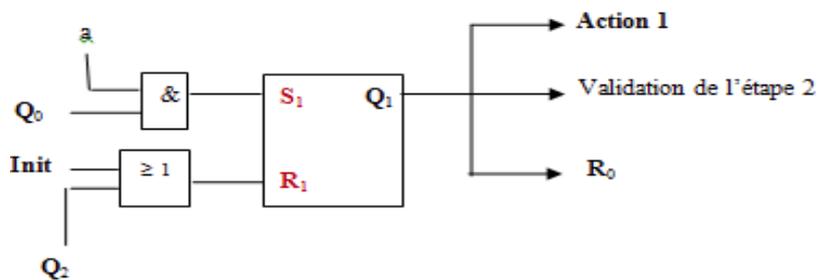


Figure 2.23. Matérialisation d'un GRAFCET par bascule RS

**Équations:**

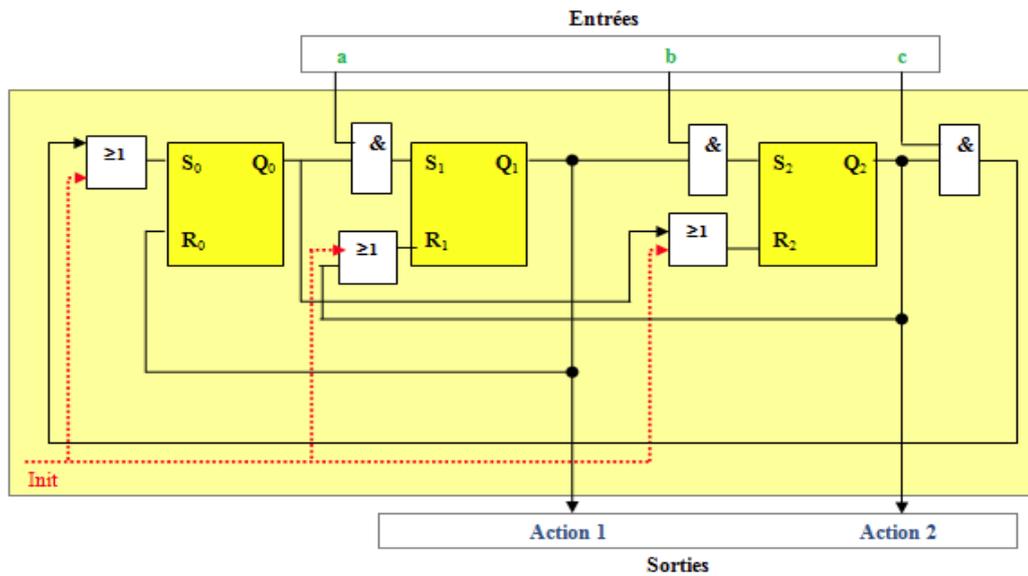
$$S_1 = Q_0 \cdot a$$

$$R_1 = Q_2 + Init$$

Il suffit de répéter cela pour chaque étape et relier le tout. Les équations logiques des commandes des bascules sont :

ACTIVATION	DESACTIVATION
$S_0 = Q_2 \cdot c + Init$	$R_0 = Q_2$
$S_1 = Q_0 \cdot a$	$R_1 = Q_2 + Init$
$S_2 = Q_1 \cdot b$	$R_2 = Q_0 + Init$

Le schéma de câblage du système sera donc:



### 2.7.2. Divergence simple en ET

Quand la transition est franchissable, il suffit d'activer deux étapes au lieu d'une. Le seul problème est la désactivation de l'étape précédente: il faut être sûr que les deux étapes suivantes ont eu le temps de prendre l'information d'activation avant de désactiver la précédente.

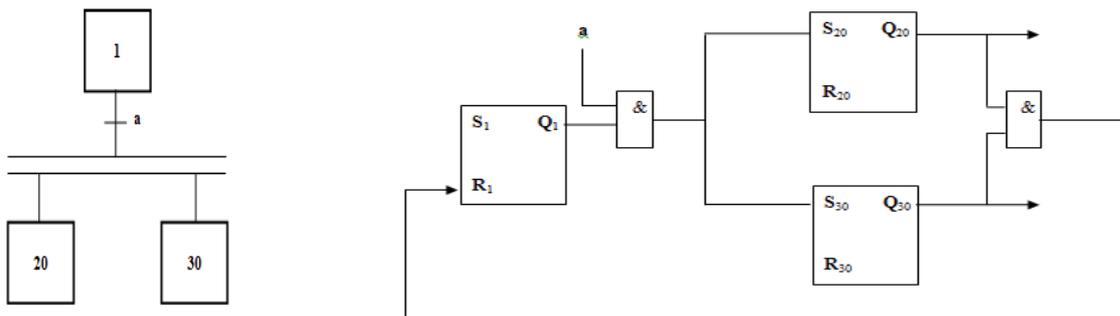


Figure 2.24. Matérialisation d'un divergence simple en ET par bascules RS

### 2.7.3. Divergence en OU

Quand l'étape 1 est active et la réceptivité (a) ou (b) est vraie, l'étape 20 ou 30 devient active et l'étape 1 désactive. Il est possible que l'évolution devienne simultanée si les deux réceptivités sont vraies.

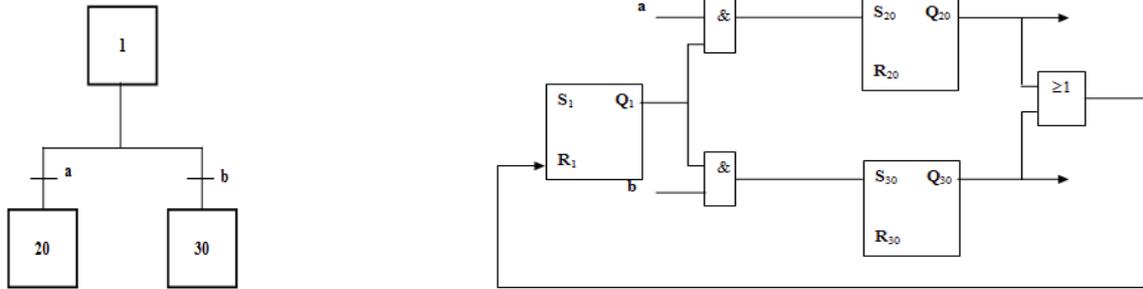


Figure 2.25. Matérialisation d'un divergence en OU par bascules RS

### 2.7.4. Convergence en ET

Quand les deux étapes (3 et 4) sont actives et la réceptivité « c » est vraie, alors l'étape 5 devient active et les deux étapes (3 et 4) désactives.

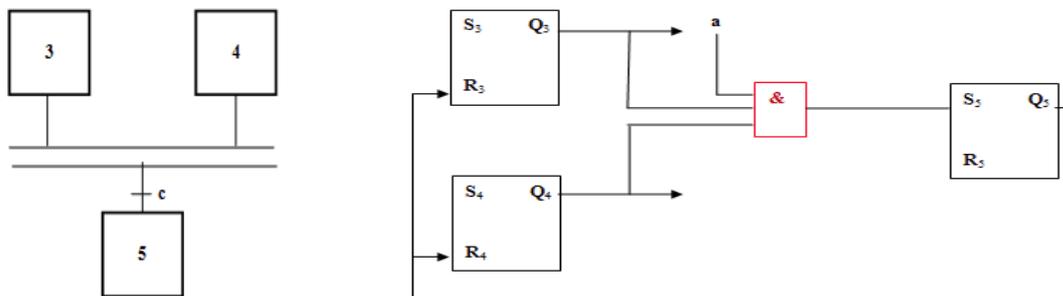


Figure 2.26. Matérialisation d'un convergence en et par bascules RS

### 2.7.5. Convergence en OU

Quand l'étape 3 (ou 4) est active et la réceptivité (a) ou (b) est vraie alors l'étape 5 devient active et l'étape 3 ou 4 désactive.

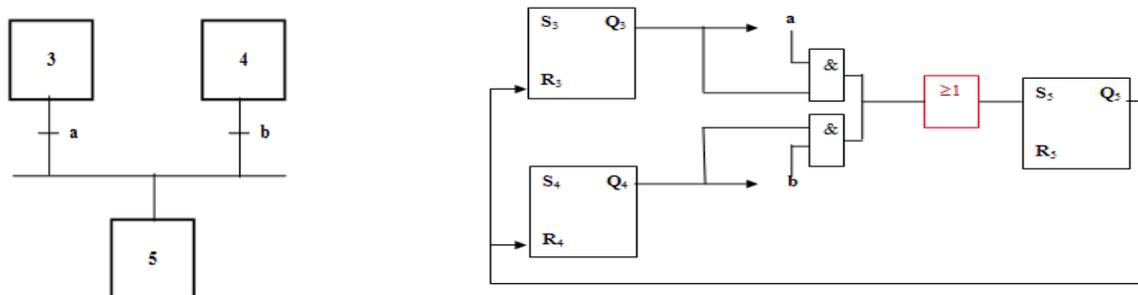


Figure 2.27. Matérialisation d'un convergence en OU par bascules RS

### 2.7.6. Exemple de grafcet avec sélection de séquence

Un puisard sert à collecter les eaux de pluies, celles-ci s'infiltrant peu à peu dans le sol autour de la cavité du puisard. Pour éviter tout débordement d'eau en cas d'afflux trop

important, on a placé deux pompes  $P1$  et  $P2$  et un détecteur de niveau comme indiqué sur la Figure 2.28. Le fonctionnement souhaité est le suivant :

1. Si le niveau d'eau  $N$  est inférieur à  $N1$  (les trois contacts  $N1$ ,  $N2$  et  $N3$  sont relâchés). Aucune des deux pompes ne fonctionne.

2. Supposons que le niveau  $N$  monte ; quand  $N$  atteint  $N2$  la pompe  $P1$  se met en marche :

- ✓ Si le niveau redescend,  $P1$  s'arrête quand  $N$  atteint  $N1$  ;
- ✓ Si le niveau continue de monter,  $P2$  se met en marche lorsque  $N$  atteint  $N3$  ;
- ✓ Lorsque les deux pompes fonctionnent et que le niveau  $N$  atteint  $N2$ , on arrête  $P2$ , mais on laisse fonctionner  $P1$ .

$P2$ , mais on laisse fonctionner  $P1$ .

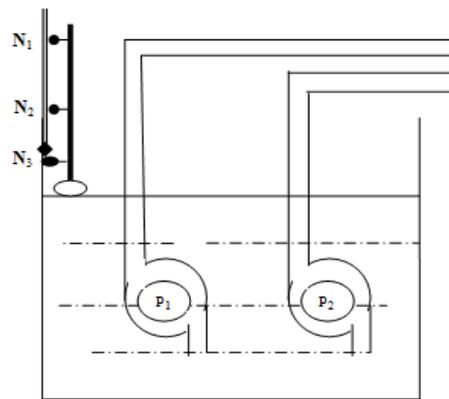


Figure 2.28. Système de deux pompes

La Figure 2.29, donne le grafcet correspondant à ce système :

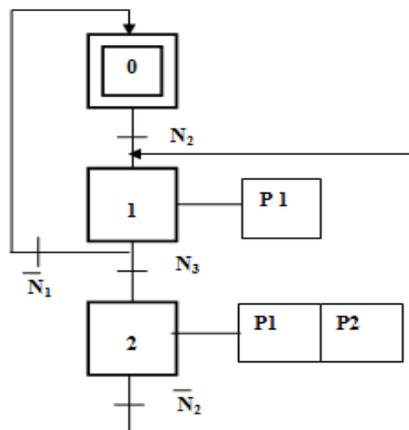
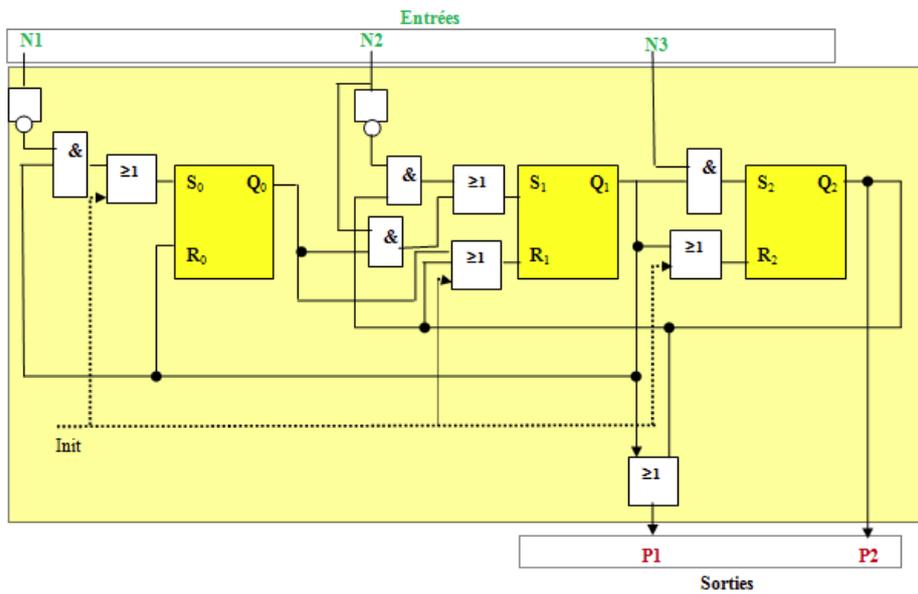


Figure 2.29. Grafcet correspondant à ce système de deux pompes

Équations logiques de commande des bascules :

ACTIVATION	DESACTIVATION
$S_0 = Q_1 \cdot \overline{N_1} +$ <i>Init</i> $S_1 = Q_0 \cdot N_2 +$ $Q_2 \cdot \overline{N_2}$ $S_2 = Q_2 \cdot N_3$	$R_0 = Q_1$ $R_1 = Q_0 + Q_2 +$ <i>Init</i> $R_2 = Q_1 + \text{Init}$



Les pompes P1 et P2 marchent à tour de rôle :

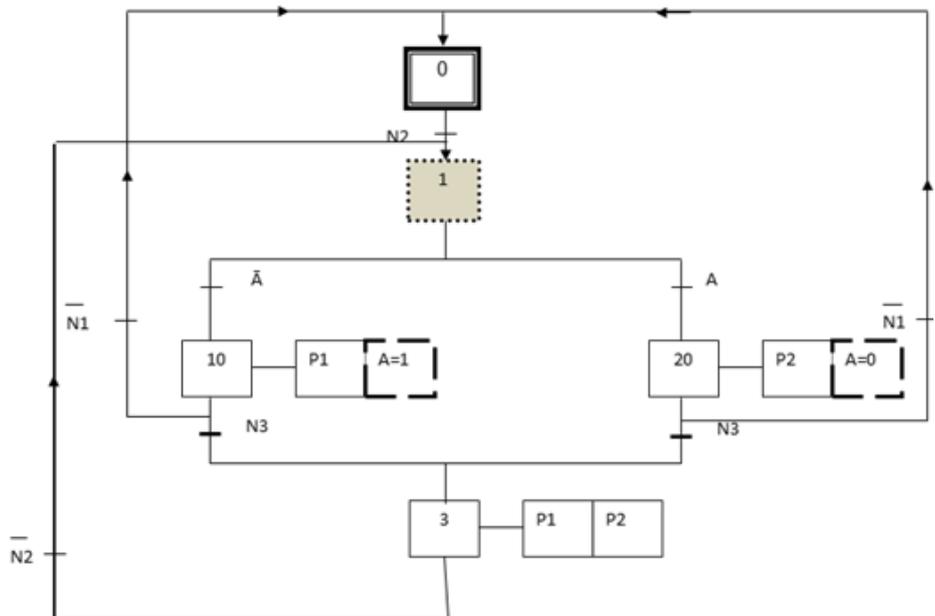


Figure 2.30. Grafcet correspondant à P1 et P2 marchent à tour de rôle

## Chapitre 3 : Automate programmable

### 3.1. Définition

Un automate programmable est un appareil dédié au contrôle d'une machine ou d'un processus industriel, constitué de composants électroniques, comportant une mémoire programmable par un utilisateur non informaticien, à l'aide d'un langage adapté. En d'autres termes, un automate programmable est un calculateur logique, ou ordinateur, au jeu d'instructions volontairement réduit, destiné à la conduite et la surveillance en temps réel de processus industriels. Trois caractéristiques fondamentales distinguent totalement l'Automate Programmable Industriel (API) des ordinateurs (PC industriel ou autres):

- il peut être directement connecté aux capteurs et pré-actionneurs grâce à ses entrées/sorties industrielles,
- il est conçu pour fonctionner dans des ambiances industrielles sévères (température, vibrations, micro-coupures de la tension d'alimentation, parasites, etc.),
- sa programmation à partir de langages spécialement développés pour le traitement de fonctions d'automatisme qui fait en sorte que sa mise en oeuvre et son exploitation ne nécessitent aucune connaissance en informatique.

### 3.2. Structure interne et description des éléments d'un A.P.I :

La structure interne d'un API peut se représenter comme suit :

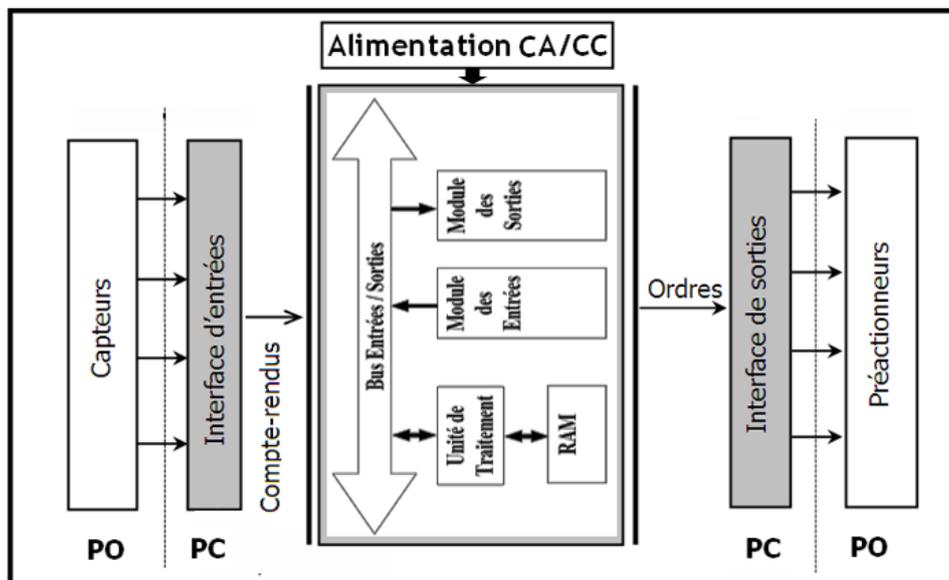
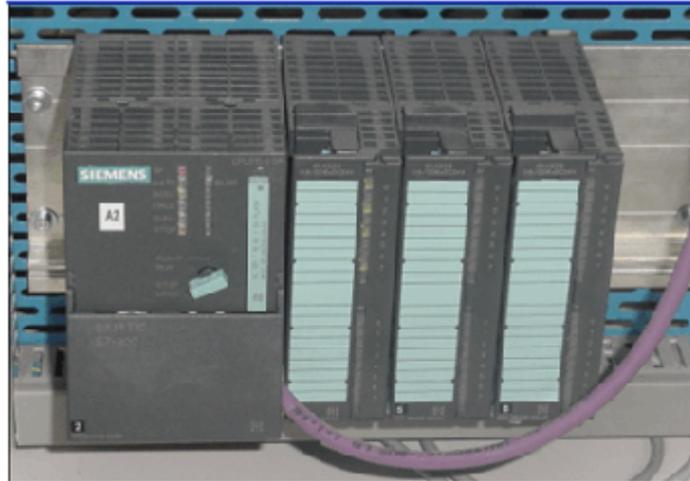


Figure.3.1. Architecture d'un API

L'automate programmable reçoit les informations relatives à l'état du système et puis commande les préactionneurs suivant le programme inscrit dans sa mémoire. Un API se compose donc de quatre grandes parties :

1. Le processeur ;
2. La mémoire ;
3. Les interfaces entrées/sorties
4. Alimentation



*Figure.3.2. Architecture réelle d'un API S7-300.*

Ces quatre parties sont reliées entre elles par des bus (ensemble câblé autorisant le passage de l'information entre ces 4 secteurs de l'API). Ces quatre parties réunies forment un ensemble compact appelé automate. La structure interne d'un automate programmable industriel (API) est assez voisine de celle d'un système informatique simple. L'unité centrale est le regroupement du processeur et de la mémoire centrale. Elle commande l'interprétation et l'exécution des instructions programme. Les instructions sont effectuées les unes après les autres, séquencées par une horloge.

#### ➤ **Le processeur**

Le processeur, ou unité centrale (UC), a pour rôle principal le traitement des instructions qui constituent le programme de fonctionnement de l'application (les fonctions logiques ET, OU, les fonctions de temporisation, de comptage, de calcul PID, etc...). Mais en dehors de cette tâche de base, il réalise également d'autres fonctions :

- Gestion des entrées/sorties.
- Surveillance et diagnostic de l'automate par une série de tests lancés à la mise sous tension ou cycliquement en cours de fonctionnement.

- Dialogue avec le terminal de programmation, aussi bien pour l'écriture et la mise au point du programme qu'en cours d'exploitation pour des réglages ou des vérifications des données.
- Un ou plusieurs processeurs exécutent ces fonctions grâce à un micro logiciel préprogrammé dans une mémoire de commande, ou mémoire système. Cette mémoire morte définit les fonctionnalités de l'automate. Elle n'est pas accessible à l'utilisateur.

➤ **La mémoire**

Elle est conçue pour recevoir, gérer, stocker des informations issues des différents secteurs du système que sont le terminal de programmation (PC ou console) et le processeur, qui lui gère et exécute le programme. Elle reçoit également des informations en provenance des capteurs. Il existe dans les automates plusieurs types de mémoires qui remplissent des fonctions différentes :

- La conception et l'élaboration du programme font appel à la RAM et l'EEPROM.
- La conservation du programme pendant l'exécution de celui-ci fait appel à une EPROM.

➤ **L'alimentation :**

Tous les automates actuels utilisent un bloc d'alimentation alimenté en 240 V alternative ( $V_{ac}$ ) et délivrant une tension de 24 V continu ( $V_{cc}$ )

➤ **Les interfaces :**

Les interfaces d'entrées comportent des adresses d'entrées, une pour chaque capteur relié. Les interfaces de sorties comporte des adresses de sorties, une pour chaque pré-actionneur. Le nombre d'entrées / sorties varie suivant le type d'automate. Les cartes d'entrées / sorties ont une modularité de 8, 16 ou 32 voies. Elles admettent ou délivrent des tensions continues 0 - 24 Vcc.

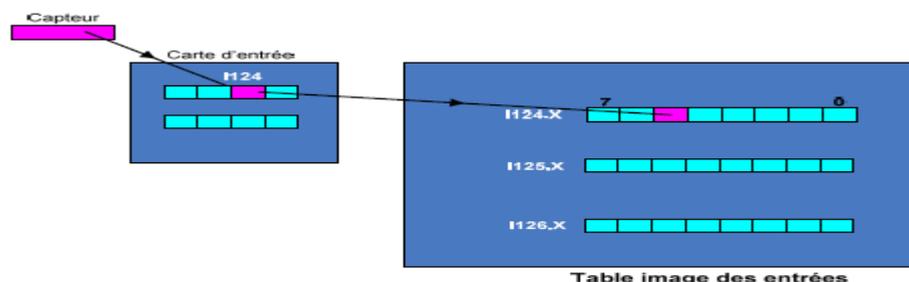


Figure.3.3. Interface d'entrée d'un API

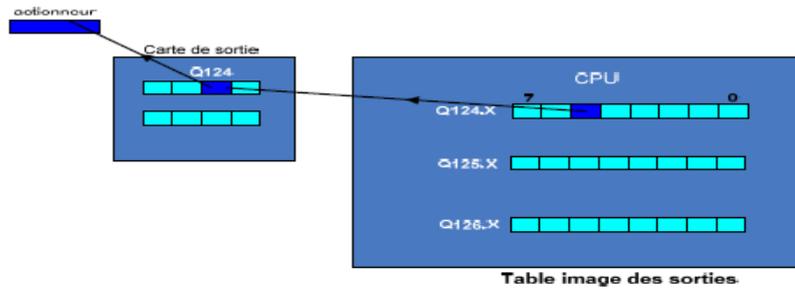


Figure.3.4. Interface de sortie d'un API

L'automate programmable **reçoit** les informations relatives à l'état du système et puis **commande** les pré-actionneurs suivant le programme inscrit dans sa mémoire. Généralement les automates programmables industriels ont un fonctionnement cyclique.

Le **microprocesseur** réalise toutes les fonctions logiques ET, OU, les fonctions de temporisation, de comptage, de calcul,... . Il est connecté aux autres éléments (mémoire et interface E/S) par des liaisons **parallèles** appelées « **BUS** » qui véhiculent les informations sous forme binaire. Lorsque le fonctionnement est dit synchrone par rapport aux entrées et aux sorties, le cycle de traitement commence par la prise en compte des entrées qui sont figées en mémoire pour tout le cycle.

### 3.3. Choix de l'unité de traitement :

Le choix de l'unité de traitement d'un API est conditionné par son prix, ses performances ou son temps de réponse, ses possibilités de connexion à des réseaux, ses capacités de calcul et de la taille de sa mémoire.

#### *a- Le prix de l'UC :*

Le prix de l'UC varie en fonction des capacités qu'elle possède. Le prix intervient dans le choix de l'API, mais il reste très faible en comparaison avec l'ensemble du projet.

#### *b- Le temps de réponse :*

Le temps de réponse est fonction de la vitesse du microprocesseur, du temps d'accès à la RAM et de la taille du programme. Le constructeur donne généralement la valeur du temps de traitement d'un bit ou d'un mot. Par exemple la CPU 215 possède un temps de traitement d'un bit de  $0.18\mu s$  et d'un mot de  $0.78\mu s$ . Le temps de scrutation du programme se déduit par la multiplication du nombre de traitement de bits et de mots par le temps de traitement d'un bit et d'un mot

#### ***c- La connexion à des réseaux :***

Les API modulaires de haute gamme ont la possibilité d'être connectés à des réseaux. Des cartes additionnelles et des coupleurs de communication assurent le dialogue avec d'autres équipements branchés à des réseaux PROFIBUS et ETHERNET.

#### ***d- Les possibilités de calculs des UC :***

Les API de bas de gamme intègrent des fonctions de calcul simples. Les API de haute gamme offrent la possibilité de faire des calculs complexes sur des variables réelles, de traiter des chaînes de caractères, d'utiliser des fonctions de régulation (PID), de dialoguer avec d'autres équipements.

#### ***e- La taille mémoire des UC :***

La mémoire des API est encore très chère. Elle doit être nécessaire et suffisante pour accepter le programme. En général, la taille mémoire utilisée pour le programme utilisateur évolue de 48KB à 512KB. Certains UC sont dotées de mémoire flash. Qui sauvegarde le programme. A la mise sous tension de l'API, le programme résidant dans la mémoire flash est transféré dans la RAM de l'API. Cette dernière peut servir à stocker des données de production.

### **3.4. Choix d'un automate programmable industriel :**

Le choix d'un API est adapté aux besoins après l'établissement du cahier des charges. On doit tenir compte de plusieurs critères à savoir :

- Le nombre et la nature d'entrées/sorties intégrés;
- La nature du traitement (temporisation, comptage, ...);
- Les moyens de dialogue et le langage de programmation;
- La communication avec les autres systèmes;
- Les moyens de sauvegarde du programme ;
- La fiabilité, robustesse et immunité aux parasites ;
- Capacité de la mémoire ;
- La documentation, le service après vente, durée de la garantie et la formation ;

### **3.5. Les interfaces d'entrées-sorties :**

Les entrées/sorties TOR (Tout ou Rien) assurent l'intégration directe de l'automate dans son environnement industriel en réalisant la liaison entre le processeur et le processus. Elles ont une double fonction :

- Une fonction d'interface pour la réception et la mise en forme de signaux provenant de l'extérieur (capteurs, boutons poussoirs, etc.) et pour l'émission de signaux vers

l'extérieur (commande de pré-actionneurs, de voyants de signalisation, etc.). La conception de ces interfaces avec un isolement galvanique ou un découplage opto-électronique assure la protection de l'automate contre les signaux parasites.

- Une fonction de communication pour l'échange des signaux avec l'unité centrale par l'intermédiaire du bus d'entrées/sorties.

Le fonctionnement de l'interface d'entrée (figure 3.5) peut être résumé comme suit :

Lors de la fermeture du capteur ;

- ✓ La « **Led 1** » signale que l'entrée de l'API est actionnée.
- ✓ La « **Led D'** » de l'optocoupleur « **Opto 1** » s'éclaire.
- ✓ Le phototransistor « **T'** » de l'optocoupleur « **Opto 1** » devient passant.
- ✓ La tension  $V_s=0V$ .

Donc lors de l'activation d'une entrée de l'automate, l'interface d'entrée envoie un « 0 » logique à l'unité de traitement et un « 1 » logique lors de l'ouverture du contact du capteur (entrée non actionnée).

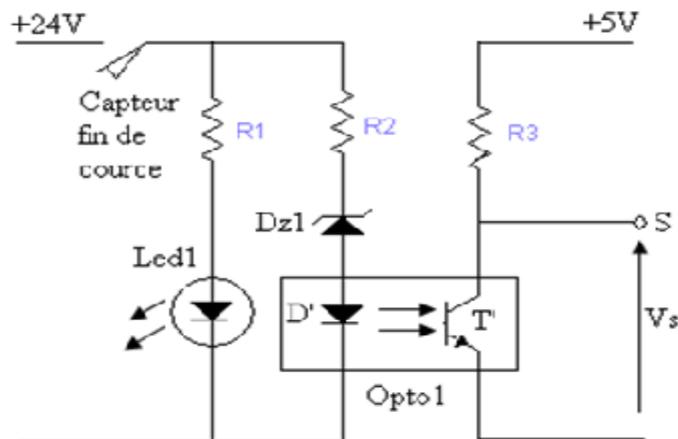


Figure.3.5. fonctionnement de l'interface d'entrée d'un API

Le fonctionnement de l'interface de sortie (figure 3.6) peut être résumé comme suit :

Lors de commande d'une sortie automate ;

- ✓ L'unité de commande envoie un « 1 » logique (5V).
- ✓ « T1 » devient passant, donc la « Led D' » s'éclaire
- ✓ Le photo-transistor « T' » de l'optocoupleur « Opto1 » devient passant.
- ✓ La « Led1 » s'éclaire.
- ✓ « T2 » devient passant.
- ✓ La bobine « RL1 » devient sous tension et commande la fermeture du contact de la sortie « Q0.1 ».

Donc pour commander un API, l'unité de commande doit envoyer :

- ✓ Un « 1 » logique pour actionner une sortie API
- ✓ Un « 0 » logique pour stopper la commande d'une sortie API

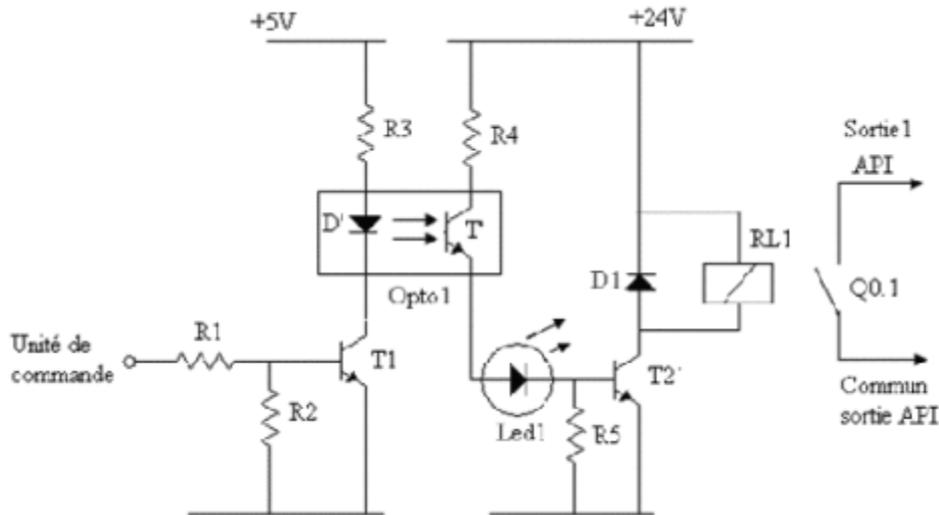


Figure.3.6. fonctionnement de l'interface d'entrée d'un API

La figure 3.7 donne une idée concrète sur un module TOR industriel.

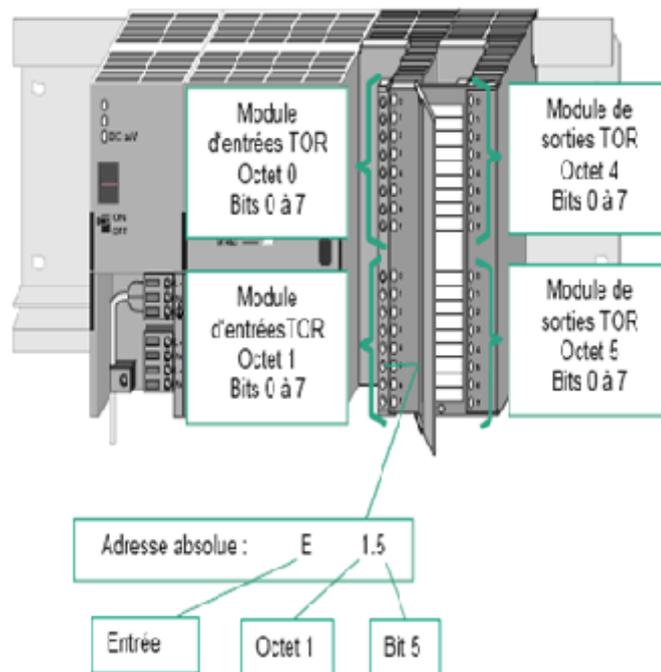


Figure.3.7. Schéma de principe d'un module TOR industriel

L'ensemble de connecteurs qui connectent différents éléments de l'API est appelé bus. Dans un automate modulaire, il se présente sous forme d'un circuit imprimé situé au fond du bac et supporte des connecteurs sur lesquels viennent s'enficher les différents modules : processeur, extension mémoire, interfaces et coupleurs.

Le bus est organisé en plusieurs sous ensembles destinés chacun à véhiculer un type bien défini d'informations :

- Bus de données.
- Bus d'adresses.
- Bus de contrôle pour les signaux de service tels que tops de synchronisation, sens des échanges, contrôle de validité des échanges, etc...
- Bus de distribution des tensions issues du bloc d'alimentation.

### 3.6. Outils graphiques et textuels de programmation :

La norme IEC 1131-3 définit cinq langages qui peuvent être utilisés pour la programmation des automates programmables industriels. Ces cinq langages sont :

- **LD** (« Ladder Diagram », ou schéma à relais): Ce langage graphique est essentiellement dédié à la programmation d'équations booléennes (vraie/faux).
- **IL** (« Instruction List », ou liste d'instructions): ce langage textuel de bas niveau est un langage à une instruction par ligne. Il peut être comparé au langage assembleur.
- **FBD** (« Function Block Diagram », ou schéma par blocs): ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions. Il permet de manipuler tous les types de variables.
- **SFC** (« Sequential Function Char »): issu du langage GRAFCET, ce langage, de haut niveau, permet la programmation aisée de tous les procédés séquentiels.
- **ST** («Structured Text » ou texte structuré): ce langage est un langage textuel de haut niveau. Il permet la programmation de tout type d'algorithme plus ou moins complexe.

#### 3.6.1. Objets communs à tous les langages

Toute expression, constante ou variable, utilisée dans un programme doit être caractérisée par un type, les types de base sont :

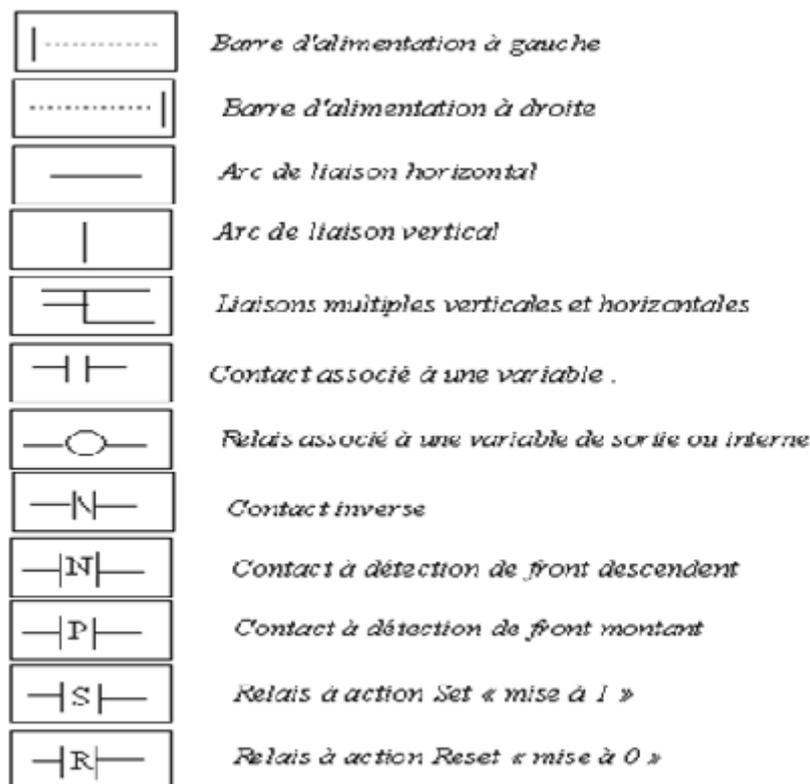
- Booléen : **BOOL** (Vraie ou Faux qui sont équivalent à 1 ou 0).
- Entier : **DINT** (c'est un nombre signé entre -2147483647 et +2147483647. Il est exprimé dans l'une des bases suivantes : décimale, hexadécimale, octale ou binaire).
- Réel : **REAL** (il prend 1 bit de signe +23 bits de mantisse +8 bits d'exposant compris entre -37 et +37).
- Temporisation : **TIME** (elle ne peut jamais être négative et commencer par **T#** ou **TIME#**).

- Chaîne : **STRING** (elle doit être précédée et suivie par une apostrophe, et ne doit jamais excéder 255 caractères). Le caractère spécial ('\$') est utilisée pour insérer des caractères non imprimables.

### 3.6.2. Langage LD

Le langage LD (ladder diagram) est une représentation graphique d'équations booléennes combinant des contacts (en entrée) et des relais (en sortie). Il permet la manipulation de données booléennes, à l'aide de symboles graphiques organisés dans un diagramme comme les éléments d'un schéma électrique à contacts. Les diagrammes LD sont limités à gauche et à droite par des barres d'alimentation.

Les composants graphiques élémentaires d'un diagramme LD sont :



### 3.6.3. Langage IL

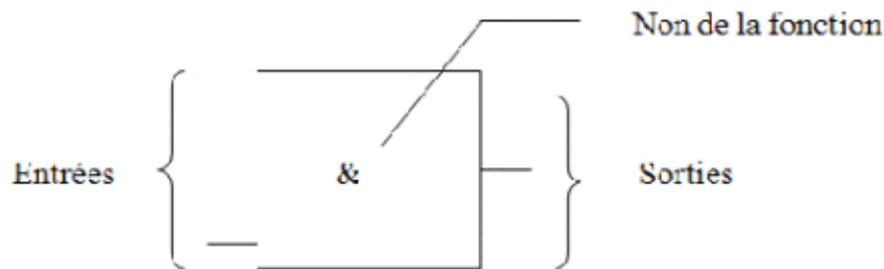
Le langage IL (instruction list), est un langage textuel de bas niveau. Il est particulièrement adapté aux applications de petite taille. Les instructions opèrent toujours sur un résultat courant (ou registre IL). L'opérateur indique le type d'opération à effectuer entre le résultat courant et l'opérande. Le résultat de l'opération est stocké à son tour dans le résultat courant.

Un programme IL est une liste d'instructions. Chaque instruction doit commencer par une nouvelle ligne, et doit contenir un opérateur, complété éventuellement par des modificateurs et si c'est nécessaire pour l'opération, un ou plusieurs opérandes séparés par des virgules (',').

Une étiquette suivie de deux points (':') peut précéder l'instruction. Si un commentaire est attaché à l'instruction, il doit être le dernier élément de la ligne. Des lignes vides peuvent être insérées entre des instructions. Un commentaire peut être posé sur une ligne sans instruction.

### 3.6.4. Langage FBD

Le langage FBD (function block diagram) est un langage graphique. Il permet la construction d'équations complexes à partir des opérateurs standards, de fonctions ou de blocs fonctionnels.



Les principales fonctions sont :

- L'énoncé **RETURN** (peut apparaître comme une sortie du diagramme, si la liaison connectée prend l'état booléen **TRUE**, la fin du diagramme n'est pas interprétée.
- Les étiquettes et les sauts conditionnels sont utilisés pour contrôler l'exécution du diagramme. Aucune connexion ne peut être réalisée à droite d'un symbole d'étiquette ou de saut.
- Saut à une étiquette (le nom de l'étiquette est « **LAB** »).

Si la liaison à gauche du symbole de saut prend l'état booléen **TRUE**, l'exécution du programme est déournée après l'étiquette correspondante. L'inversion booléenne est représentée par un petit cercle.

### 3.6.5. Langage SFC

Le langage **SFC** (Sequential Function Chart), ou **GRAFCET**, est un langage graphique utilisé pour décrire les opérations séquentielles. Le procédé est représenté comme une suite connue d'étapes (états stables), reliées entre elles par des transitions, une condition booléenne est attachée à chaque transition. Les actions dans les étapes sont décrites avec les langages **ST**, **IL**, **LD** ou **FBD**.

Les principales règles graphiques sont :

- un programme **SFC** doit contenir au moins une étape initiale.
- une étape ne peut pas être suivie d'une autre étape.

- une transition ne peut pas être suivie d'une autre transition.
- Les composants de base (symboles graphiques) du graphique SFC sont :
  - étapes et étapes initiales.
  - transitions.
  - liaisons orientées.
  - renvoi à une étape.

Les différents types d'action sont :

- action booléenne (Elle est forcée à chaque fois que le signal d'activité de l'étape change d'état.)
- action impulsionnelle programmée en **ST**, **LD** ou **IL** (c'est une liste d'instructions **ST**, **IL** ou **LD**, exécutée à chaque cycle pendant toute la durée d'activité de l'étape).
- action normale programmée en **ST**, **LD** ou **IL** ;
- action **SFC** (Une action **SFC** est une séquence fille **SFC**, lancée ou tuée selon les évolutions du signal d'activité de l'étape. Elle peut être décrite avec les qualificatifs d'action **N** (non mémorisée), **S** (set), ou **R** (reset).)

Plusieurs actions (de même type ou de types différents) peuvent être décrites dans la même étape. Un appel de fonctions ou de blocs fonctionnels permet d'intégrer des traitements décrits dans d'autres langages (**FBD**, **LD**, **ST** ou **IL**).

### 3.6.6. Langage ST

Le langage **ST** (Structured Text) est un langage textuel de haut niveau dédié aux applications d'automatisation. Ce langage est principalement utilisé pour décrire les procédures complexes, difficilement modélisables avec les langages graphiques. C'est le langage par défaut pour la programmation des actions dans les étapes et des conditions associées aux transitions du langage **SFC**.

Un programme **ST** est une suite d'énoncés. Chaque énoncé est terminé par un point virgule (« ; »). Les noms utilisés dans le code source (identificateurs de variables, constantes, mots clés du langage...) sont délimités par des séparateurs passifs ou des séparateurs actifs, qui ont un rôle d'opérateur. Des commentaires peuvent être librement insérés dans la programmation.

Les types d'énoncés standard sont :

- assignation (variable := expression);
- appel de fonction ;
- appel de bloc fonctionnel ;

- énoncés de sélection (**IF, THEN, ELSE, CASE**) ;
- énoncés d'itération (**FOR, WHILE, REPEAT**) ;
- énoncés de contrôle (**RETURN, EXIT**) ;
- opérateurs booléens (**NOT, AND, OR, XOR**) ;
- énoncés spéciaux pour le lien avec le langage **SFC**.

Il est recommandé de respecter les règles suivantes quand on utilise les séparateurs passifs, pour assurer une bonne lisibilité du code source :

- ne pas écrire plusieurs énoncés sur la même ligne ;
- utiliser les tabulations pour indenter les structures de contrôle ;
- insérer des commentaires.

### **3.7. Mise en œuvre d'un automate programmable industriel :**

La réalisation de la partie commande ou d'une partie de celle-ci en logique programmée nécessite la traduction du modèle concerné (GRAFCET, schémas, équations, ...) en programme exécutable par la machine. L'élaboration d'un tel programme vise donc à écrire les équations d'activation de sorties de l'API et les conditions associées. Elle constitue la phase logicielle de l'application.

#### **a- Affectations :**

En l'absence de normes universelles, l'automaticien doit adapter les modèles aux langages des matériels retenus. Cela nécessite au préalable l'affectation des E/S et des variables auxiliaires (variables internes), c'est-à-dire l'identification avec le repérage à caractère informatique (adressage). Ce repérage est fonction du matériel et du type de programmation retenus. Les raccordements et leurs caractéristiques essentielles (tension, courant,...) sont souvent précisés dans la documentation constructive. En général, les objets techniques raccordés en entrée sont symbolisés par un contact et ceux raccordés en sortie par un rectangle. Des liaisons peuvent être précâblées à la construction (communs des 0V) imposant les modes de branchements extérieurs.

#### **b- Adressage des E/S :**

C'est l'association aux différentes affectations les repérages adéquats : par exemple pour le **SIMATIC S7-300**, 0 à 7 pour les entrées et 0 à 7 pour les sorties. Exemple : machine de conditionnement commandée par le **SIMATIC S7-300** :

<b>OBJET</b>	<b>ADRESSE</b>
<b>Entrées</b>	
présence palette au poste 1 (S4)	E0,5 (entrée 5)
présence flacon au poste 1 (S5)	E0,6 (entrée 6)
<b>Sorties</b>	
commande vibreur (M2)	A4,0 (sortie 0)
commande sole tournante (M3)	A4,1 (sortie 1)

**c- Variables internes :**

L'affectation consiste également à identifier ces variables destinées à mémoriser les états et valeurs intermédiaires durant l'exécution du programme.

# Chapitre 4 : Guide d'Etude des Modes Marche et Arrêt (G.E.M.M.A)

## 4.1. INTRODUCTION :

Un système automatisé est conçu dans le but de produire la valeur ajoutée. Il est alors dans une situation dite de « production normale ». Pendant son cycle de vie, un système ne produit pas en continu ; il tombe en panne, il doit subir des opérations de maintenance ou de réglage, etc. On définit alors 3 grandes familles de modes de marche et d'arrêt.

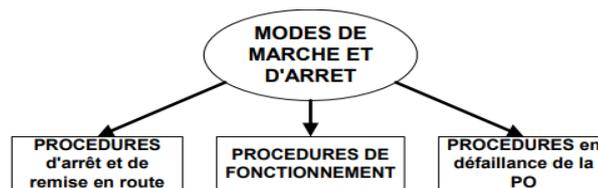


Figure 4.1. Les trois familles de modes de marche et d'arrêt

Cette formalisation des modes de marche et d'arrêt (caractérisant l'état de fonctionnement d'un système automatisé) a abouti en 1981, sous l'impulsion de l'ADEPA, à un guide graphique : le GEMMA (Guide d'Etude des Modes de Marche et d'arrêt). Ces modes de marche et d'arrêt concernent :

- Les défauts de production de l'équipement automatisé
- Les différentes phases d'exploitation de la machine
- Les états et procédures de mise en sécurité sur défaut, sur défaillance ou sur consigne de sécurité.

L'étude faite avec le GEMMA est très importante et son objectif est le suivant:

- Le GEMMA (acronyme de Guide d'Etude des Modes de Marche et d'Arrêt) est un guide graphique (document graphique) permettant de sélectionner et de décrire simplement les différents états de marches et d'arrêts, ainsi que les possibilités d'évoluer d'un état à un autre. Il a été créé parce qu'il y avait un grand besoin d'avoir un vocabulaire commun et précis.
- Le GEMMA permet d'avoir une approche guidée de l'analyse des modes de marches et d'arrêts.
- Le GEMMA permet le recensement et la description des différents états du système automatisé, de la mise en route à la production normale. Il précise les procédures à mettre en œuvre après analyse d'une anomalie ou un défaut de fonctionnement. Pour

une machine donnée, les modes de marches et d'arrêts doivent être choisis et compris de toutes les personnes chargées d'intervenir.

- Un GEMMA est établi pour chaque machine lors de sa conception, puis utilisé tout au long de sa vie : réalisation, mise au point, maintenance, modifications, réglages... Dans ses principes et dans sa mise en oeuvre, le GEMMA doit donc être connu de toutes les personnes concernées par les automatismes, depuis leur conception jusqu'à leur exploitation.

## **4.2. Concept et structuration du GEMMA**

Le GEMMA est composé de deux grandes parties :

### **4.2.1. Partie commande hors énergie :**

Cette zone du GEMMA, située à l'extrême gauche, correspond à l'état inopérant de la partie commande. Elle ne figure que pour la forme. Dans cet état, la partie opérative n'est pas sous le contrôle de la partie commande. La partie opérative peut être en énergie ou hors énergie. La sécurité est garantie par les choix technologiques et la procédure de mise en énergie de la partie opérative.

### **4.2.2. Partie commande en énergie et active :**

Cette zone permet de décrire ce qui se passe lorsque la partie commande (P.C.) fonctionne normalement. Elle couvre la quasi-totalité du guide graphique. Cette partie va nous permettre de définir les différents modes de marches et d'arrêt de notre machine ainsi que les conditions de passage d'un mode à l'autre. Elle est subdivisée en trois zones ou en trois familles de procédures :

#### **✓ Famille A :**

Procédures d'arrêt et de remise en route de la partie opérative : une installation ne peut fonctionner indéfiniment. Il s'avère nécessaire de l'arrêter (à partir du pupitre de commande) de temps à autre pour des raisons normales indépendantes du système :

- Fin de journée,
- Pause,
- Manque d'approvisionnement ...

#### **✓ Famille D :**

Procédures de défaillance de la partie opérative : il est rare qu'un système fonctionne sans incident durant toute sa vie. Il faut envisager qu'il aura des défaillances prévisibles ou imprévisibles. On regroupera dans cette famille tous les modes conduisant à un état d'arrêt du

système pour des raisons intérieures au système, autrement dit, à cause de défaillances de la partie opérative. Les procédures de la famille D ont pour objectif de limiter au maximum les conséquences ou risques pour le personnel ou matériel.

✓ **Famille F :**

Procédures de fonctionnement : on regroupe dans cette famille tous les modes ou états qui sont indispensables à l'obtention de la valeur ajoutée. Notons que l'on ne produit pas forcément dans tous les modes de cette famille. Il peut s'agir de :

- Procédures préparatoire à la production de la valeur ajoutée
- réglages et tests qui sont néanmoins indispensables à la production de la valeur ajoutée

Comme le montre le schéma ci-dessous,

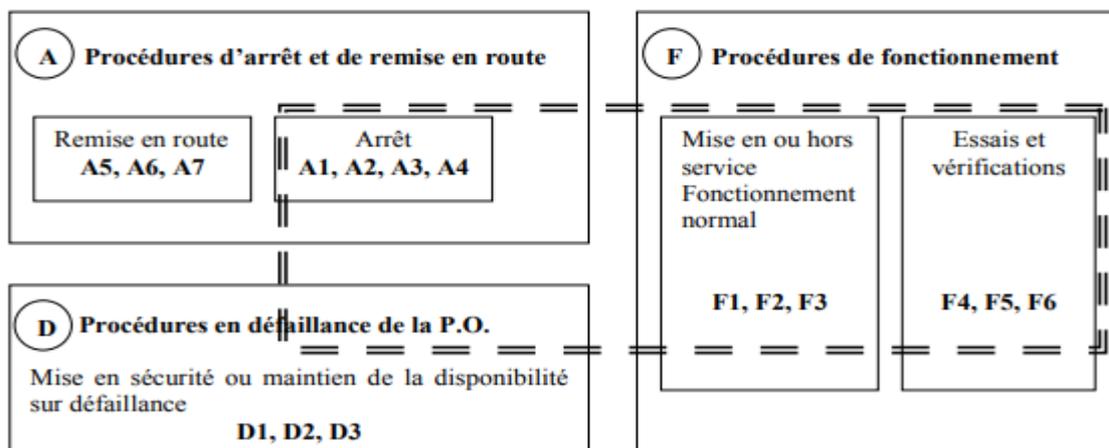


Figure 4.2. Les familles et les sous-familles de procédures.

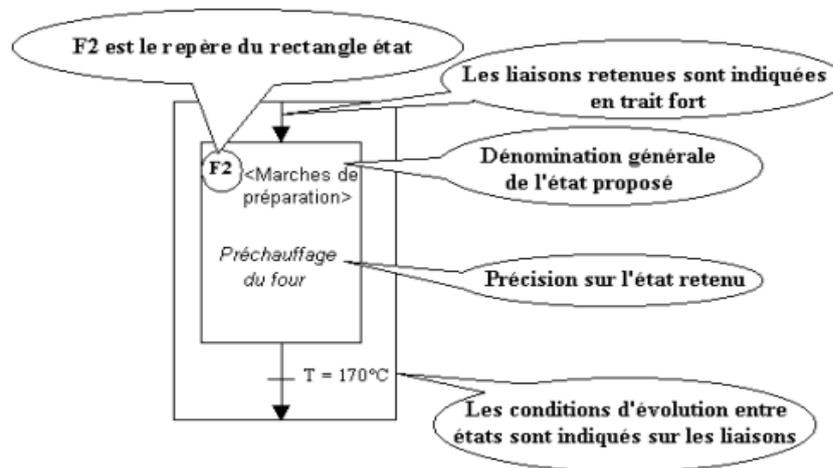
Une distinction supplémentaire est faite parmi ces trois familles de procédures. On distingue la zone de production de la zone hors production par un double encadrement de la zone de production en pointillés. La zone de production se trouve à cheval sur les trois types de procédures.

### 4.3. Les rectangles-états :

Chacun des différents états ou modes dans lequel un automate peut être représenté par un rectangle-état. La figure.4.3 montre le détail de l'un de ces rectangles-états. Chaque rectangle-état appartient à une des trois familles A, F ou D. Le symbole d'appartenance commence par une lettre identifiant la famille à laquelle appartient le rectangle-état.

Un rectangle-état est identifié par un «titre» désignant le mode selon le *vocabulaire général* accepté par l'AFCEC. Dans le rectangle-état, il est inscrit par l'utilisateur (dans la place réservée) une description claire de ce qui se passe dans ce mode. Toute précision jugée nécessaire doit y être inscrite. On y indique aussi l'*appellation «maison»* utilisée pour identifier ce mode. Si le rectangle-état n'est pas assez gros pour y inscrire tous les détails, on

devrait le décrire sur des pages en annexe auquel le rectangle-état devra faire référence. Enfin, si un rectangle-état n'est pas utilisé, il doit être barré par un «X». Les rectangles-états sont reliés entre eux par des lignes pointillées indiquant les évolutions les plus probables. Les évolutions retenues seront représentées par des traits forts continus tracés par l'utilisateur. Une condition d'évolution sera alors ajoutée près de ce trait. Seul un rectangle-état est actif dans un GEMMA. Ce principe dit «**d'unicité de mode**» est imposé et est fort logique. Il est impossible qu'une machine soit en production normale et en arrêt en même temps. Il faut choisir, où la machine produit, où la machine est arrêtée.



*Figure.4.3. Représentation d'un rectangle-état*

D'autre part, le but d'une machine étant de produire de la valeur ajoutée, on distingue, à l'intérieur de ces trois familles de procédures, la zone de production de la zone de non production. Cette zone est à cheval sur les trois familles de procédures. Chaque zone est subdivisée en rectangles d'états ou procédures (figure.4.4).



## **4.4. Procédures de fonctionnement (la famille F)**

### **4.4.1. Production normale (F1)**

Ce rectangle-état est obligatoire dans tout GEMMA puisque la production normale est le but essentiel de tout automatisme. Il correspond au GRAFCET de niveau 2 (GRAFCET de base) sans étapes initiales (qui correspondent au rectangle-état A1). Il est d'ailleurs mis en évidence dans le GEMMA puisque le rectangle-état est tracé avec un trait large. Ce rectangle-état correspond souvent à un mode identifié « *mode AUTOMATIQUE* ». Pour y accéder à partir de l'arrêt en condition initiale (A1), il faut que le mode automatique soit sélectionné et que l'opérateur appui sur un bouton poussoir nommé « *DCY* », abréviation pour « *départ de cycle* ». Il faut aussi que la machine soit en condition initiale. Si nécessaire, certaines préparations doivent être faites avant l'entrée en production. Donc de l'arrêt en condition initiale, il est possible d'évoluer vers la production normale selon les deux séquences suivantes :

A1 → F1

Démarrage sans marche de préparation;

A1 → F2 → F1

Démarrage avec marche de préparation.

### **4.4.2. Marches de préparation (F2)**

Ce rectangle-état est utilisé lorsque l'automatisme exige certaines préparations avant de passer en production normale. Ces préparations peuvent être faites automatiquement ou manuellement. Pendant cette phase de préparation, il peut y avoir production ou non, ce qui explique que ce rectangle-état chevauche la frontière entre la zone «en production» et la zone «hors production».

Par exemple, si l'automatisme est une unité de scellage, il faut préchauffer la tête de scellage à une température adéquate avant d'entrer en production. Pendant le préchauffage la machine n'est pas en mesure de produire.

Dans une marche de préparation, il est possible que les opérations à faire soient manuelles. Ainsi, on peut demander à l'opérateur de remplir les bols vibrants avec une certaine quantité de pièces. Le GEMMA agit alors à titre d'aide mémoire pour l'opérateur (ou pour celui qui aura à rédiger les directives d'utilisation de la machine).

Le remplissage d'un convoyeur et la mise en marche progressive des postes d'une machine est une marche de préparation dans laquelle il y a production. L'exemple de la machine à remplir et à boucher montrera plus loin ce qu'il en est.

#### 4.4.3. Marches de clôture (F3)

Lorsque la machine exige de faire certaines opérations (en plus de la fin de cycle normal) pour retourner en condition initiale, il faut prévoir une marche dite de clôture. Cette marche peut être faite automatiquement ou manuellement. Selon les actions à faire, il y a production ou non, ce qui explique que ce rectangle-état chevauche la frontière entre la zone «en production» et la zone «hors production».

Par exemple, on peut prévoir une marche de clôture pour indiquer à l'opérateur de nettoyer la machine. On peut aussi y prévoir la vidange d'un convoyeur et l'arrêt progressif des postes d'une machine (voir machine à remplir et à boucher).

#### 4.4.4. Marches de vérification dans le désordre (F4)

Ce mode souvent appelé «mode MANUEL» permet de tester et de vérifier des mouvements ou même des fonctions séparées sans respecter l'ordre des actions du cycle de fonctionnement normal. Si le mode manuel est utilisé, il faudrait prévoir sur l'automatisme toutes les sécurités nécessaires pour éviter des incidents. Des inter-verrouillages (interlocks) devraient être ajoutés sur les actionneurs présentant des risques de collisions ou de bris. Les inter-verrouillages doivent être câblés. Le mode manuel est habituellement accessible depuis l'arrêt en état initial (A1) ou de la marche de vérification dans l'ordre (F5). Les deux évolutions possibles sont donc :

A1 → F4

Accès au mode MANUEL de l'arrêt en condition initiale ;

F5 → F4

Accès au mode MANUEL depuis la marche de vérification dans l'ordre.

Le mode manuel ne devrait pas être accessible directement du mode automatique. Si ces deux modes sont les seuls d'un automatisme donné, le sélecteur de mode devrait avoir trois positions, la position centrale étant une position neutre.

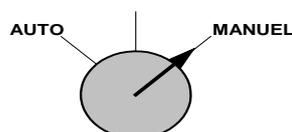


Figure.4.6. Sélecteur de modes (MANUEL/AUTO)

Du mode manuel, il faut obligatoirement passer par la remise de la machine en condition initiale (A6) avant de retourner à l'arrêt en condition initiale (A1). L'évolution à partir de F4 est :

F4 → A6 → A1.

#### 4.4.5. Marches de vérification dans l'ordre (F5)

Ce mode existe pour que le technicien effectuant la mise au point puisse voir le fonctionnement normal du cycle de production à une cadence lente. Dans certaines applications, la machine peut produire pour faciliter la mise au point. Dans d'autres applications, elle peut ne pas produire. Les modes de marches associées à ce rectangle-état sont :

- mode étape par étape :
  - ❖ Accessible si le sélecteur de mode est sur «EE» ;
  - ❖ Exécution d'une étape à la fois suite à l'appui sur un bouton poussoir «BPee».
- mode séquence par séquence :
  - ❖ Accessible si le sélecteur de mode est sur «SS» ;
  - ❖ Exécution d'une série d'étapes (séquence) à la fois suite à l'appui sur un bouton poussoir «BPss».
- mode cycle par cycle :
  - ❖ Accessible si le sélecteur de mode est sur «CC» ;
  - ❖ Exécution d'un cycle à la fois suite à l'appui sur un bouton poussoir «BPcc» ;
  - ❖ Appelé parfois «mode semi-automatique».

Ces différents modes sont sujets à certaines variantes qui doivent être spécifiées. Le mode peut s'appliquer sur l'ensemble du système (mode étape par étape de l'ensemble du système), sur un ou des postes (mode séquence par séquence, poste par poste), sur une ou des tâches.

Ce mode F5 est habituellement accessible depuis l'arrêt en condition initiale (A1) ou de la production normale (F1). Les deux évolutions possibles sont donc :

A1 → F5 :

Accès à la marche de vérification dans l'ordre à partir de l'arrêt en condition initiale ;

F1 → F5 :

Accès à la marche de vérification dans l'ordre depuis la production normale.

Ce mode peut être abandonné soit pour retourner à l'arrêt en état initial (F5 → A1), soit pour passer au mode automatique (F5 → F1), soit pour le mode manuel (F5 → F4).

#### 4.4.6. Marches de test (F6)

Pour des besoins de maintenance, d'étalonnage et de réglages périodiques, les marches de test sont utiles. En mettant le sélecteur de mode en position «*TEST*», il est alors possible de calibrer des capteurs de température et de débit. Il est aussi possible d'ajuster des débits, de modifier des paramètres.

### 4.5. Procédures d'arrêt (la famille A)

#### 4.5.1. Arrêt dans l'état initial (A1)

Ce rectangle-état est obligatoire dans tout GEMMA puisque lorsque l'automatisme est arrêté, il est habituellement en condition initiale. Il correspond aux étapes initiales des GRAFCET de niveau 2 (GRAFCET de base). Le tracé de ce rectangle ressemble à celui des étapes initiales d'un GRAFCET.

La Partie Opérative de la machine doit être dans son état initial, tel que définit par les étapes initiales du GRAFCET de niveau 2. Lors de la mise en énergie (ou de l'activation pour un automate) de la Partie Commande on, accède directement à A1 si la condition initiale est remplie, sinon il faudra passer par le rectangle état de mise en condition initiale (A6) au préalable. Pour indiquer qu'une machine est en condition initiale, on utilise souvent la variable « *ci* » qui est la résultante d'une équation logique vérifiant que la Partie Opérative est en condition sur le GEMMA, car l'équation logique vérifiant cette condition devient très grosse dans les machines complexes. Ainsi, si on considère l'exemple de la machine à remplir et à boucher présentée au chapitre précédent, on aurait pour la variable « *ci* » :  $ci = a_0 \cdot b_0 \cdot c_0 \cdot d_0 \cdot e_0$ . Si on utilise la variable « *ci* », il faudrait que l'équation correspondante apparaisse clairement en annexe du GEMMA.

De la production normale, les évolutions possibles vers l'arrêt dans l'état initial sont :

**F1 → A2 → A1 :**

Arrêt en fin de cycle sans marche de clôture ;

**F1 → F3 → A1 :**

Arrêt avec une marche de clôture

La Partie Opérative d'une machine devrait être schématisée dans sa position initiale. Ces dessins sont des éléments important dans les dossiers techniques de la machine.

#### 4.5.2. Arrêt demandé en fin de cycle (A2)

Ce mode est un état transitoire vers l'état A1. Il peut être considéré comme une mémorisation de la demande d'arrêt en fin de cycle. Pour y accéder (à partir de **F1** ou **D3**). Il

faut appuyer sur un bouton poussoir « *ACY* », (abréviation de « *arrêt en fin de cycle* »). Le passage au rectangle A1 exige que le cycle en cours se soit terminé et que la machine soit revenue en condition initiale.

#### **4.5.3. Arrêt demandé dans un état déterminé (A3)**

Ce mode est un état transitoire vers l'état A4. C'est une mémorisation de la demande d'arrêt de la machine dans un état autre que l'état initial. On accède à cet état à partir des rectangles-états **F1** et **D3**. Dans plusieurs applications, cet arrêt consiste simplement en un figeage de l'automatisme.

#### **4.5.4. Arrêt obtenu (A4)**

Si la machine doit s'arrêter dans un état différent de la condition initiale, ce rectangle-état est requis. La machine s'arrêtera selon une des méthodes suivantes :

- ❖ Arrêt à une (ou des) étape donnée ;
- ❖ Arrêt à l'étape en cours en terminant les mouvements en cours ;
- ❖ Arrêt à l'étape en cours en bloquant les mouvements en cours.

Une condition de redémarrage sera nécessaire pour retourner en production normale (F1).

#### **4.5.5. Préparation pour remise en route après défaillance (A5)**

Suite à une panne, il peut être nécessaire de procéder à diverses opérations qui sont souvent manuelles :

- ❖ Dégagement de pièces ;
- ❖ Dépannage des actionneurs ;
- ❖ Nettoyage ;

On accède à ce rectangle-état depuis les rectangles-états D1 ou D2. La Partie Commande ne doit pas générer d'action, sauf si le fonctionnement en mode manuel est envisagé. La Partie Opérative peut être en ou hors énergie, selon les besoins.

#### **4.5.6. Mise de la P.O. dans un état initial (A6)**

Lors de la mise en énergie de l'automatisme, il peut se produire qu'il ne soit pas dans son état initial. Auquel cas, la machine ne peut pas aller directement en A1 lors de la mise en énergie. Le rectangle-état A6 est prévu pour déclarer la façon dont se fera l'initialisation de la machine.

Dans ce rectangle-état, on indique comment l'automatisme doit être remis en condition initiale. Cela peut être effectué comme suit :

- Mise en référence manuelle :
  - ❖ Passage en mode manuel ;
  - ❖ Un voyant «Référence» indiquera l'atteinte de la condition initiale ;
- Mise en référence automatique
  - ❖ Par actions simultanées sur tous les actionneurs ;
  - ❖ Par actions séquentielles ;
  - ❖ Lancement de la mise en référence par appui sur le bouton poussoir nommé «Bref»

Ce rectangle-état doit être inséré dans toutes les évolutions vers A1 qui impliquent que la machine n'est pas en référence :

**A5 → A6 → A1 :**

Initialisation nécessaire suite à la préparation pour remise en route après défaillance ;

**F4 → A6 → A1 :**

Initialisation suite à un mode manuel ;

**Mise en énergie → A6 → A1 :**

Si la machine n'est pas en condition initiale lors de la mise en énergie

#### **4.5.7. Mise de la P.O. dans un état déterminé (A7)**

Il est possible de placer certains automatismes dans un état autre que l'état initial avant de redémarrer la production. Par exemple, certains automatismes voir leurs actionneurs doivent être remis dans le même état qu'ils étaient lors de l'arrêt d'urgence.

L'évolution à partir de A7 pour retourner en production normale (D1) sera :

**A7 → A4 → F1.**

### **4.6. Procédures en défaillances (la famille D) ,**

#### **4.6.1. Arrêt d'urgence (D1)**

Ce rectangle état correspond à une situation critique ou l'automatisme doit réagir immédiatement. On peut (et on doit) y accéder de tout autre état suite à l'appui d'un bouton «coup de poing» identifié «AU» pour arrêt d'urgence, ou de façon automatique suite à la détection d'une situation critique ou d'un défaut dans la Partie Opérative.

Les procédures envisagées ici sont :

- Arrêt immédiat ou en fin de mouvement des actionneurs par figeage ;

- Arrêt immédiat des actionneurs par inhibition des actions ;
- Procédures limitant les dégâts ;
- Dégagements ;

Simultanément, il faudrait avertir l'opérateur en enclenchant un klaxon et/ou un gyrophare. On affiche un message sur l'interface opérateur indiquant la nature et la gravité de la défaillance.

#### **4.6.2. Diagnostic et/ou traitement de défaillance (D2)**

Lors d'une défaillance, il peut être nécessaire de procéder à un diagnostic pour trouver l'origine de la défaillance. Cela permet de réduire le temps requis pour dépanner l'automatisme et par conséquent d'en augmenter la disponibilité. Suite à un diagnostic, le technicien est en mesure de traiter la défaillance et de réparer l'automatisme.

Ce rectangle-état s'applique lorsque la machine est relativement complexe ou lorsqu'il est possible que l'origine de la défaillance soit difficile à trouver. L'automatisme peut éventuellement programmer des fonctions de diagnostic pour aider le technicien à trouver l'origine de la panne. Selon la nature de l'incident, le retour à la production sera plus ou moins fastidieux. Voici deux façons possibles de faire.

- En cas de panne légère, il est possible d'évoluer selon la séquence suivante :

**D2 → A5 → A7 → A4 → F1.**

Donc, on fait le diagnostic et on traite la panne, l'opérateur nettoie les dégâts, la machine est remise dans les conditions quelle avait lors de l'incident, puis elle attend un ordre de redémarrage avant d'aller en production normale.

- En cas de panne lourde ou grave, il est fortement recommandé de retourner en condition initiale selon l'évolution suivante :

**D2 → A5 → A6 → A1.**

Une fois arrêtée en condition initiale, la machine peut être redémarrée normalement, avec ou sans marche de préparation.

Il est important de noter que la Partie Commande ne doit envoyer aucune action. La Partie Opérative peut être en/ou hors énergie, selon la nature de la panne.

#### **4.6.3. Production tout de même (D3)**

Pour certaines défaillances très bénignes, par exemple un poste dans une chaîne de production, il est possible de produire quand même. Bien sûr, il ne faut surtout pas que la

sécurité des personnes et des biens soit diminuée. La production sera alors dégradé et exigera l'ajout de opérateurs non prévus pour remplacer le poste défaillant.

Par exemple, si le poste de bouchage de la machine à remplir et à boucher tombe en panne, on peut le désactiver et demander à un ou deux opérateurs de boucher manuellement les bouteilles.

Généralement, on prévoit des sélecteurs pour désactiver le poste défaillant.

Il est aussi possible qu'un technicien puisse réparer le poste alors que le reste de la machine fonctionne.

De façon générale, l'évolution vers D3 est :

**F1 → D3.**

Si l'opérateur à fait un arrêt d'urgence, il peut arriver qu'il puisse repartir l'automatisme, mais sans un poste qui est défectueux.

L'évolution pourra donc ressembler à :

**D1 → D2 → A5 → A7 → A4 → D3** ou encore **D1 → A5 → A6 → A1.**

Du rectangle-état D3, il n'est pas recommandé de passer directement au rectangle-état de production normale (F1). Si tout redevient normal on recommande de retourner à F1 via un arrêt en condition initiale (A1) :

**D3 → A2 → A1** ou via un arrêt dans un état quelconque (A4) : **D3 → A3 → A4.**

## **4.7. Utilisation pratique du GEMMA et applications.**

### **4.7.1. Mise en contexte**

Dans la pratique courante, la sélection des modes de marches, d'arrêts et de défaillances se fait souvent de façon non méthodique. Il se produit souvent que des gens oublient de penser à certains modes et ne les intègrent pas dans le GRAFCET final de la machine. Lorsque la machine est installée chez le client, les surprises apparaissent et on découvre rapidement que certains modes ont été oubliés. Cela entraîne un mauvais vécu de la machine par le client et des modifications qui peuvent être longues et pénibles.

Pour nous éviter de se retrouver dans une telle situation, le GRAFCET a recommandé une approche méthodique de l'analyse des spécifications d'un automatisme.

#### **Étape 1:**

- Étude du processus d'action et définition du cycle de production normal.
- ❖ Définition des fonctions, des informations et des actions nécessaires au bon fonctionnement de l'automatisme.

- Cela mène au GRAFCET de niveau 1.

### **Étape 2:**

- Définition de la Partie Opérative et des capteurs.
  - ❖ Choix de la technologie des actionneurs et des capteurs.
- Cela mène au GRAFCET de niveau 2.
  - ❖ Ce GRAFCET est le *GRAFCET opérationnel de base*.

### **Étape 3:**

- Mise en œuvre du GEMMA pour la sélection des modes de marches, d'arrêt et les procédures de défaillance avec la mise en évidence de l'évolution entre ces modes.

### **Étape 4:**

- Définition à l'aide du GEMMA des conditions d'évolution entre les états de marches et d'arrêts.
- Définition des fonctions du pupitre de commande.
- Établissement du GRAFCET final.

### **Étape 5**

- Choix de la technologie de la Partie Commande.
  - ❖ Mise en œuvre des Technoguides.

### **Étape 6:**

- Conception du schéma ou du programme de commande dans la technologie choisie.

Donc la mise en œuvre du GEMMA a lieu après que les spécifications fonctionnelles ont été analysées par les GRAFCET. Ces spécifications fonctionnelles ne concernent que le fonctionnement normal. Le GEMMA s'attarde sur les spécifications opérationnelles qui envisagent les différents modes de marches et d'arrêts, ainsi que les divers cas de défaillance.

#### **4.7.2. Utilisation du GEMMA (Sélection des modes et des évolutions).**

Pour utiliser le GEMMA, il faut commencer dans un premier temps de regarder chacun des rectangles-états et se demander quels sont ceux qui s'appliquent à l'automatisme qui est analysé. Il faut donc envisager tous les états possibles. Si le mode proposé est retenu, il sera précisé en langage littéral de fonctionnement propre à la machine (langage machine) dans le rectangle-état. Si au contraire le mode proposé n'est pas nécessaire pour la machine, une croix est portée dans le rectangle-état, pour bien signifier qu'il n'est pas retenu.

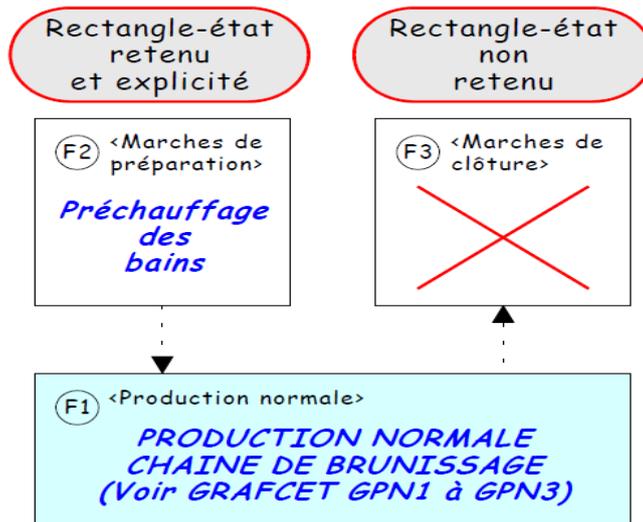


Figure.4.7. Le rectangle-état F2 est retenu, le rectangle-état F3 n'est pas retenu

Deux états essentiels, définis dès le début de l'étude, se retrouvent sur toutes les machines :

- L'état A1, dit <Arrêt dans état initial>, ou <état repos> de la machine.
- L'état F1, mode de <Production normale> pour lequel la machine a été conçue.

En partant de chacun des deux états essentiels, A1 et F1, on recherche les évolutions vers les autres états. On se demande d'abord quelle évolution suivre lors du démarrage. Le choix est :

- A1 → F1 : démarrage sans marche de préparation ;
- A1 → F2 → F1: démarrage avec marche de préparation.

Ce qui permet de répondre à la question suivante : « une marche de préparation est-elle nécessaire ? »

Ensuite, on se demande quelle évolution suivre lors de l'arrêt normal de production. Le choix est alors :

- F1 → A2 → A1 : arrêt en fin de cycle sans marche de clôture ;
- F1 → F3 → A1 : arrêt avec une marche de clôture ;
- F1 → A3 → A4 : arrêt dans un état autre que la condition initiale.

Ce qui permet de répondre aux questions suivantes :

- Une marche de clôture est elle nécessaire ?
- Un arrêt dans un état autre que la condition initial est-il nécessaire?

Puis, on se demande quelle évolution suivre lors d'une défaillance de l'automatisme. Un grand nombre de choix est disponible :

- F1 → D3 : défaillance légère permettant une marche de production tout de même ;

- D1 → D2 → A5 → A7 → A4 → D3 : arrêt d'urgence puis évolution pour production tout de même (défaillance légère impliquant l'arrêt d'un poste) ;
- D1 → A5 → A6 → A1 : arrêt d'urgence puis évolution pour un arrêt en condition initiale (défaillance majeure) ;
- D1 → A5 → A7 → A4 : arrêt d'urgence puis évolution pour un arrêt dans le même état que lors de l'apparition de l'arrêt d'urgence (défaillance mineure) ;
- D1 → D2 → A5 → A6 → A1 : arrêt d'urgence avec diagnostic et traitement, puis évolution pour un arrêt en condition initiale (défaillance majeure) ;
- D1 → D2 → A5 → A7 → A4 : arrêt d'urgence avec diagnostic et traitement, puis évolution pour un arrêt dans le même état que lors de l'arrêt d'urgence (défaillance mineure).

Ce qui permet de répondre aux questions suivantes :

- quel genre de défaillance peut affecter l'automatisme ?
- y-a-t-il des défaillances pour lesquelles on peut réussir de produire tout de même de façon sécuritaire ?
- qu'elle doit être le comportement de la machine en arrêt d'urgence ?
- la machine est elle suffisamment compliquée pour exiger un diagnostic élaboré ?
- suite à une défaillance, la machine doit elle être retournée en condition initiale ?
- suite à une défaillance, la machine peut elle être remise en marche en repartant de l'état ou elle était lors de l'apparition de l'arrêt d'urgence ?

#### **4.7.3. Utilisation du GEMMA (Conditions d'évolution entre modes marches / arrêts)**

Les modes de marches et d'arrêts ayant été sélectionnés et explicités, il convient de préciser le passage d'un état à l'autre.

L'élaboration de ces conditions de passage rend possible la conception du pupitre de commande (action de l'opérateur) et de prévoir éventuellement des capteurs supplémentaires sur la machine, de compléter le grafset, etc.

Le passage d'un état à un autre s'effectue de deux façons :

- *soit avec une condition d'évolution* : la condition peut être liée à l'action sur un bouton du pupitre de commande, ou à l'actionnement d'un capteur situé sur la machine.
- *soit sans condition d'évolution* : dans certains évolutions entre états, l'écriture d'une condition n'apporterait aucune information utile : c'est le cas lorsque celle-ci est

évidente (exemple, passage de A2 à A1), ou parce que l'état atteint dépend de l'intervenant.

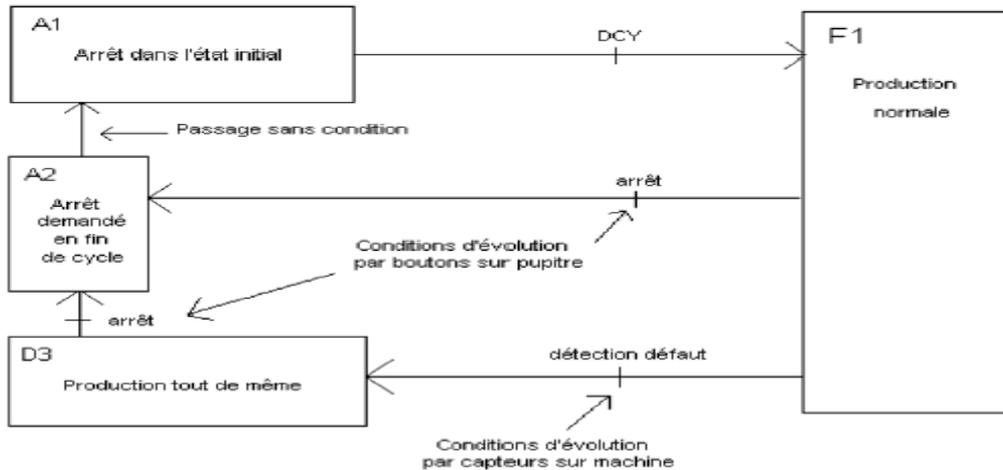


Figure.4.8. Conditions d'évolution entre modes marches / arrêts

#### 4.7.4. Exemples types de GEMMA :

Pour monter la mise en œuvre du GEMMA, nous montrons ici diverses configurations typiques. Un GEMMA réel peut être la combinaison de plusieurs de ces configurations.

##### ➤ Cas 1 : GEMMA minimal d'une machine semi-automatique :

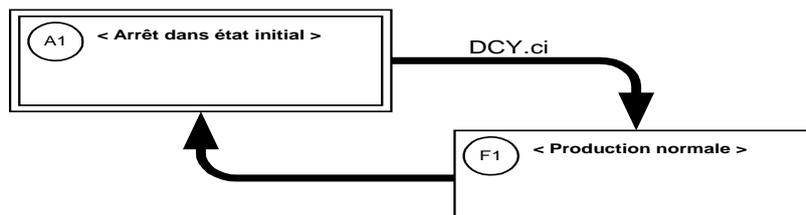


Figure.4.9. GEMMA minimal d'une machine semi-automatique

Ce GEMMA correspond au cas où la machine produit uniquement à la pièce. Ce mode de fonctionnement est appelé mode semi-automatique. Le GRAFCET résultant est directement le GRAFCET de base. La condition de mise en marche, c'est que la machine soit en condition initiale (variable « *ci* ») et que l'opérateur appui sur le bouton de départ de cycle.

##### ➤ Cas 2 : GEMMA minimal d'une machine automatique :

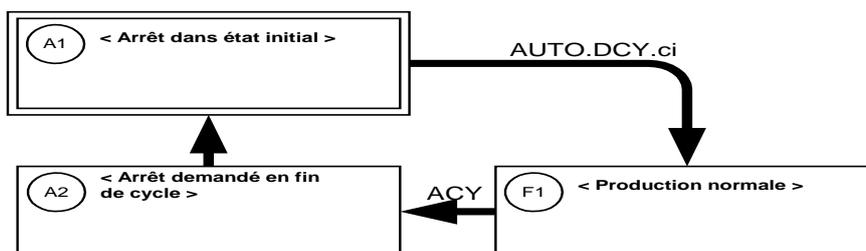


Figure.4.10. GEMMA minimal d'une machine automatique

Ce GEMMA correspond au cas où la machine produit en mode automatique. La condition de mise en marche est que le sélecteur de mode soit à automatique, que le bouton de départ de cycle soit appuyé et que la machine soit en condition initiale.

Lorsque la machine produit, elle peut s'arrêter en fin de cycle suite à l'appui du bouton d'arrêt de cycle (ACY).

➤ **Cas 3 : GEMMA d'une machine automatique ou semi-automatique exigeant une marche de préparation :**

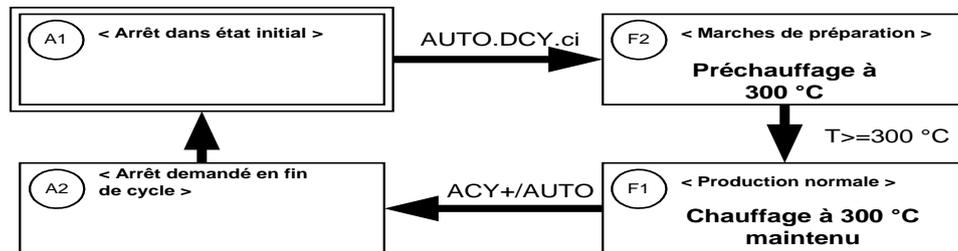


Figure.4.11. GEMMA d'une machine automatique ou semi-automatique exigeant une marche de préparation

Dans le cas où une marche de préparation est nécessaire, l'évolution doit passer par le rectangle-état F2 avant de passer en production normale.

Le schéma ci-dessus correspond à une machine automatique. Pour un *automatisme semi-automatique* ou l'opérateur doit intervenir à chaque cycle, la condition de démarrage de l'automatisme serait simplement : « *MARCHE.ci* ». Le signal DCY serait à l'intérieur du rectangle-état F1. Le signal d'arrêt de l'automatisme serait simplement identifié : ARRÊT.

➤ **Cas 4 : GEMMA d'une machine exigeant une marche de préparation et une marche de clôture:**

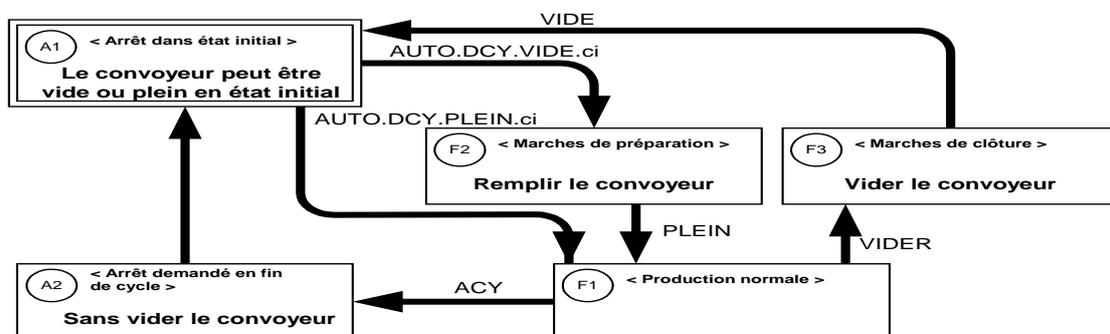


Figure.4.12. GEMMA d'une machine exigeant une marche de préparation et une marche de clôture

Si une machine nécessite une marche de clôture, par exemple pour vider un convoyeur à la fin de la journée, il faut prévoir un signal demandant l'exécution de la marche de clôture. C'est d'autant plus vrai que la machine ayant le GEMMA ci-haut peut être arrêtée en fin de

cycle pour une courte période (sans vider le convoyeur) avec le signal ACY. Donc la condition initiale de ce GEMMA c'est que la machine soit en condition initiale, avec un convoyeur vide ou plein. Cela explique que pour la mise en route, il faut vérifier si le convoyeur est plein ou vide pour savoir si la marche de préparation est nécessaire.

➤ **Cas 5 : GEMMA d'une machine exigeant une marche de préparation et une marche de clôture et ayant un arrêt dans un état autre qu'initial:**

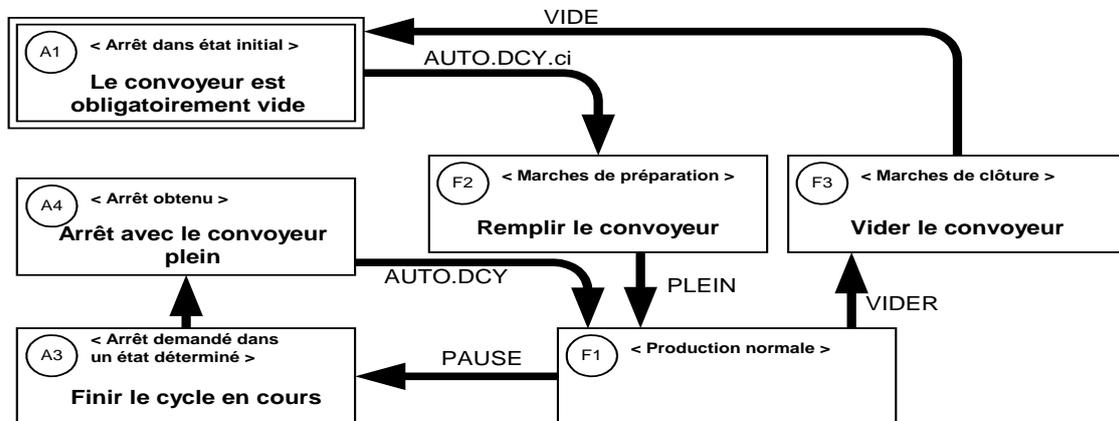


Figure.4.13. GEMMA d'une machine exigeant une marche de préparation et une marche de clôture et ayant un arrêt dans un état autre qu'initial

Si la machine exige d'être arrêtée dans un état autre que l'état initial, les rectangles-états A3 et A4 sont requis. Dans l'exemple ci-haut, on considère que la machine en condition initiale implique obligatoirement que le convoyeur est vide.

Si on désire faire une pause (un arrêt de production) pour une courte période sans vider le convoyeur (comme dans le cas #4), on ne peut retourner en condition initiale. Ainsi, l'arrêt de la machine aura lieu en A4, car le convoyeur ne sera pas vidé. La seule façon de retourner vers A1 à partir de F1, c'est de vider le convoyeur, donc de passer via la marche de clôture (F3).

➤ **Cas 6 : GEMMA d'une machine avec arrêt d'urgence**

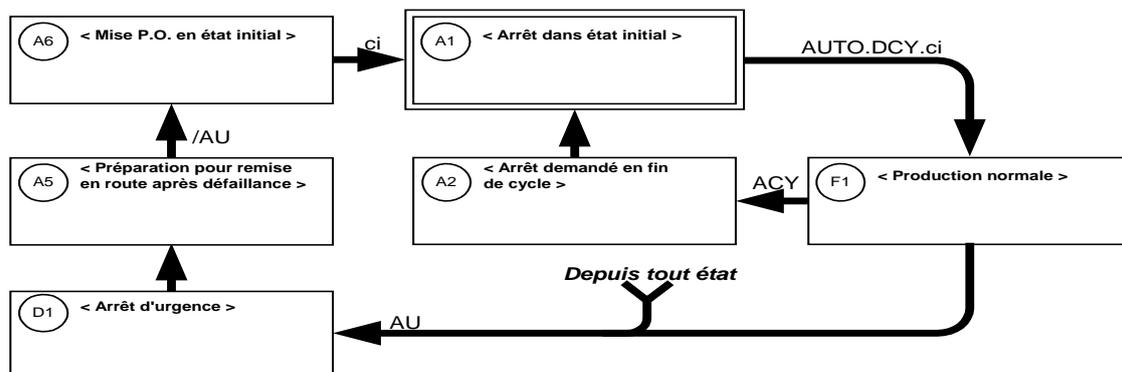


Figure.4.14. GEMMA d'une machine avec arrêt d'urgence

Dans le cas ou un arrêt d'urgence causé par une défaillance grave est envisagé, il faut pouvoir aller au rectangle-état D1 lorsque cet arrêt d'urgence se produit. Et cela, quelque soit le rectangle-état ou la machine se situe. Pour éviter d'encombrer le GEMMA, on met simplement en évidence l'évolution de F1 vers D1 et on ajoute un symbole de regroupement avec la mention «Depuis tout état».

Après l'arrêt d'urgence, il faut préparer la machine à sa remise en route en la nettoyant ou en dégageant les pièces coincées (F5), puis remettre la partie opérative en condition initiale de façon manuelle ou par une initialisation automatisée (F6). La machine sera alors prête à être redémarrée.

➤ **Cas 7 : GEMMA d'une machine avec arrêt d'urgence et remise en route de l'état ou la machine était lors de l'arrêt d'urgence.**

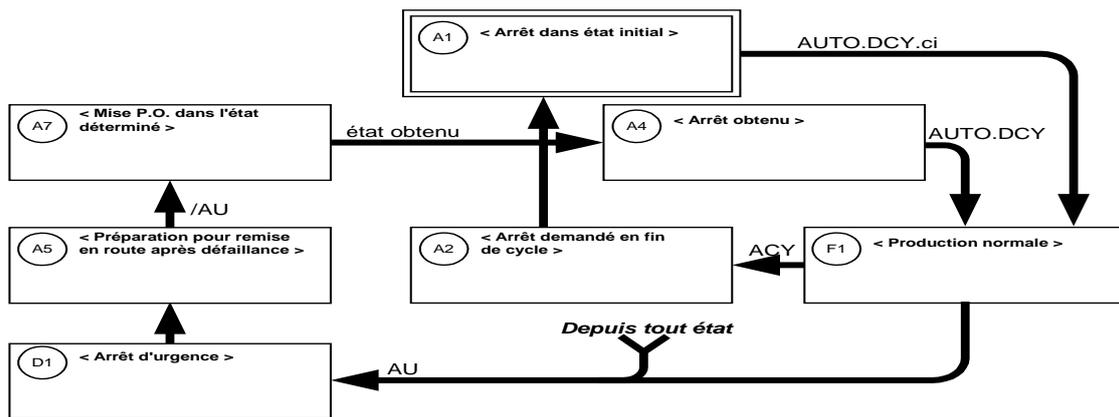


Figure.4.15. GEMMA d'une machine avec arrêt d'urgence et remise en route

Si un arrêt d'urgence causé par une défaillance légère se produit, il peut être possible de repartir l'automatisme de l'étape ou il était lors de l'arrêt d'urgence.

Auquel cas, il faut passer par la remise de la machine dans l'état qu'elle avait avant l'arrêt d'urgence, puis un redémarrage de cet état, d'où l'évolution via les états A7 et A4 pour retourner vers F1.

➤ **Cas 8 : GEMMA d'une machine avec production tout de même**

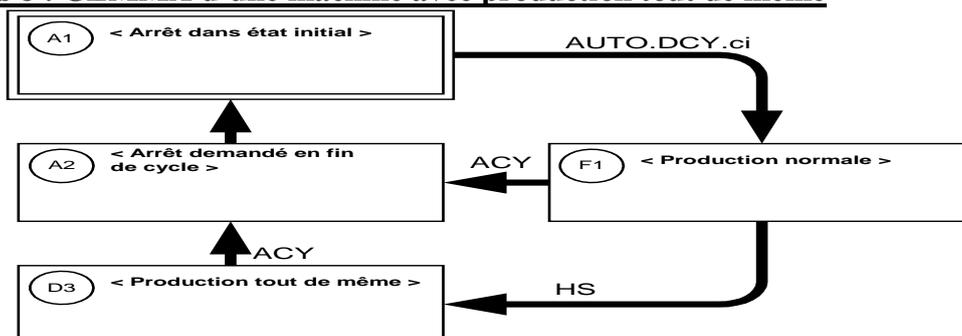


Figure.4.16. GEMMA d'une machine avec production tout de même

Si la machine peut continuer à produire malgré une défaillance, il faut passer par le rectangle-état D3. Un signal (par exemple «*HS*») doit permettre de bloquer le fonctionnement de l'élément défaillant.

Généralement, il est préférable de stopper la machine plutôt que de retourner directement de D3 vers F1. Cela vient du fait que lors de la réparation de la partie défaillante, le technicien peut avoir à actionner certains éléments en mode «manuel».

➤ **Cas 9 : GEMMA d'une machine avec marches de vérification dans l'ordre**

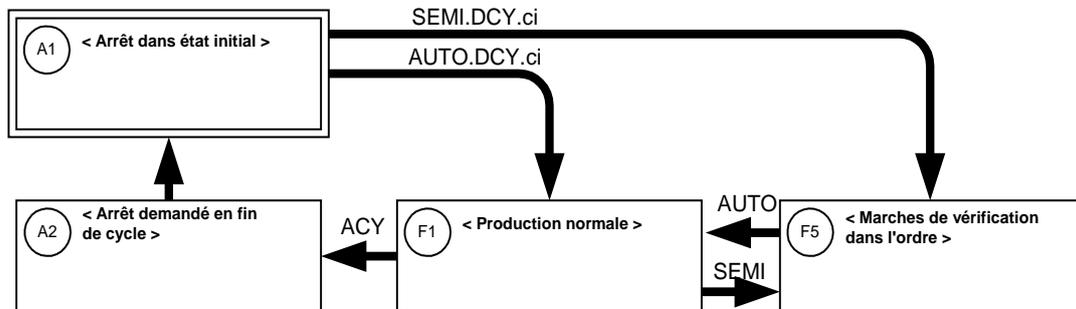


Figure.4.17. GEMMA d'une machine avec marches de vérification dans l'ordre

Si une marche de vérification dans l'ordre est prévue, on parle parfois de mode semi-automatique. Cette marche couvre les cas : marche étape par étape, marche séquence par séquence ou marche cycle par cycle. Lors de la marche de vérification dans l'ordre, il est possible de passer au mode de production normale en tournant le sélecteur en position «*AUTO*».

➤ **Cas 10 : GEMMA d'une machine avec marches de vérification dans le désordre et dans l'ordre**

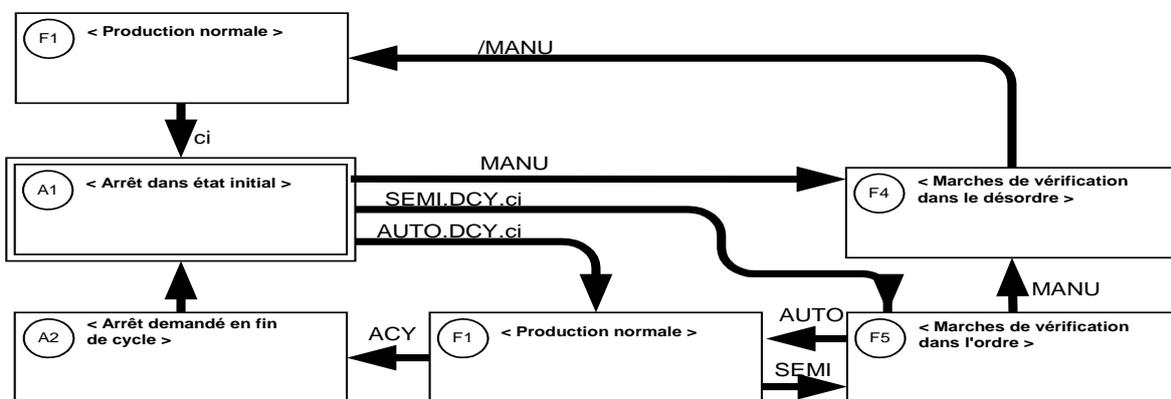


Figure.4.18. GEMMA d'une machine avec marches de vérification dans le désordre et dans l'ordre

Le mode «*MANUEL*» correspond au rectangle-état F4. On peut y accéder du mode «*SEMI-AUTOMATIQUE*» (F5) ou de l'arrêt complet dans l'état initial (A1).

Dans ce mode, l'opérateur peut actionner les actionneurs dans l'ordre qu'il le désire. Par contre, il existe toujours un risque que l'opérateur ne remette pas la machine en condition

initiale. Donc du rectangle-état F4, il faut repasser *obligatoirement* par le rectangle-état A6 pour que cette remise en condition initiale ait lieu.

#### 4.8. Passage du GEMMA à une spécification GRAFCET :

Le passage du GEMMA vers le Grafcet conduit à une structure multi-grafcets hiérarchisés en utilisant les moyens de structuration définis dans la norme IEC 60848 (Macro-étape, Grafcet partiel, Structuration par forçage, Structuration par encapsulation) :

- **GRAFCET DE CONDUITE (GC)** : Grafcet gérant tous les modes marches et d'arrêts normaux, il peut faire appel à des grafkets spécifiques (initialisation, préparation, clôture...). Le dialogue entre les grafkets est réalisé généralement par les variables internes d'étapes Xi.
- **GRAFCET DE PRODUCTION NORMALE (GPN)** : Grafcet décrivant le fonctionnement normal de production du système. Il peut faire appel à des grafkets de tâche ou de procédure. Le dialogue entre les grafkets est réalisé par les variables Xi.
- **GRAFCET DE SURETE (GS)** : Grafcet gérant les procédures de défaillance. Il est *Hiérarchiquement supérieur* à tous les autres grafkets. Cette hiérarchie est réalisée par des ordres de **Forçage**.

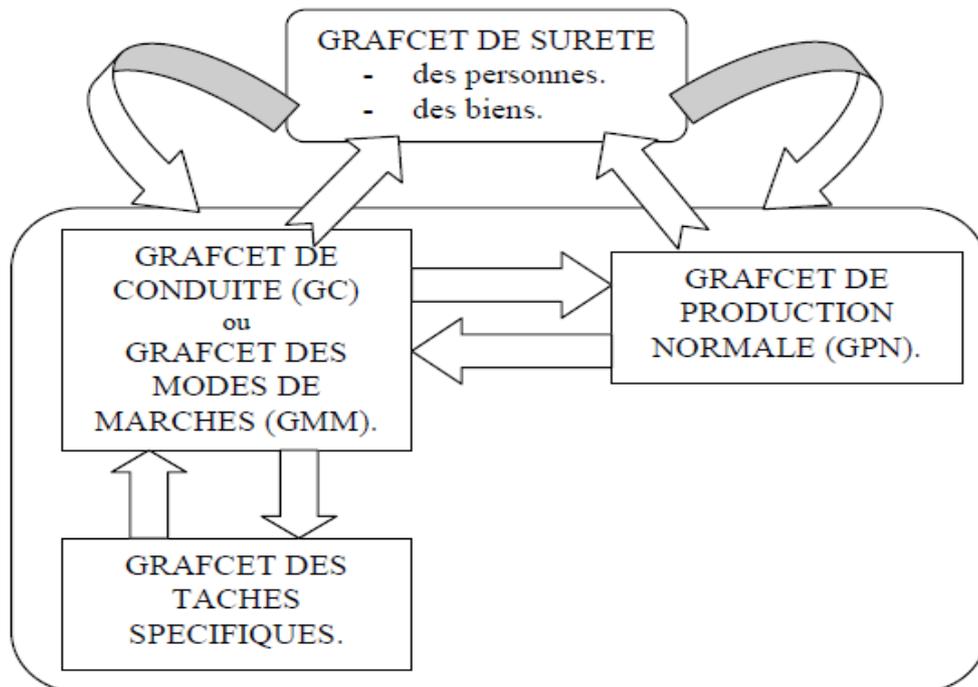


Figure.4.19. Réalisation du GRAFCET à partir du GEMMA

#### 4.8.1. Effet du GEMMA sur le Grafcet :

Le GEMMA affecte directement le GRAFCET de niveau 2, car il envisage tous les cas de fonctionnement autres que le fonctionnement normal qui est couvert par le GRAFCET.

Cette combinaison du GRAFCET et du GEMMA mène à deux approches possibles :

- Enrichissement du GRAFCET ;
- Découpages en tâches.

##### 4.8.1.1. Elaboration du Grafcet de conduite d'un système :

L'élaboration du grafcet de conduite (GC) ou de modes de marches (GMM) se fait en suivant le parcours GEMMA et en appliquant les règles suivantes:

- 1 rectangle-état = 1 étape + action associée.
- 1 condition logique = 1 transition + réceptivité associée
- Plusieurs départ d'un rectangle état = Divergence de sélection de séquences
- Plusieurs arrivées sur un rectangle état = Convergence de sélection de séquences

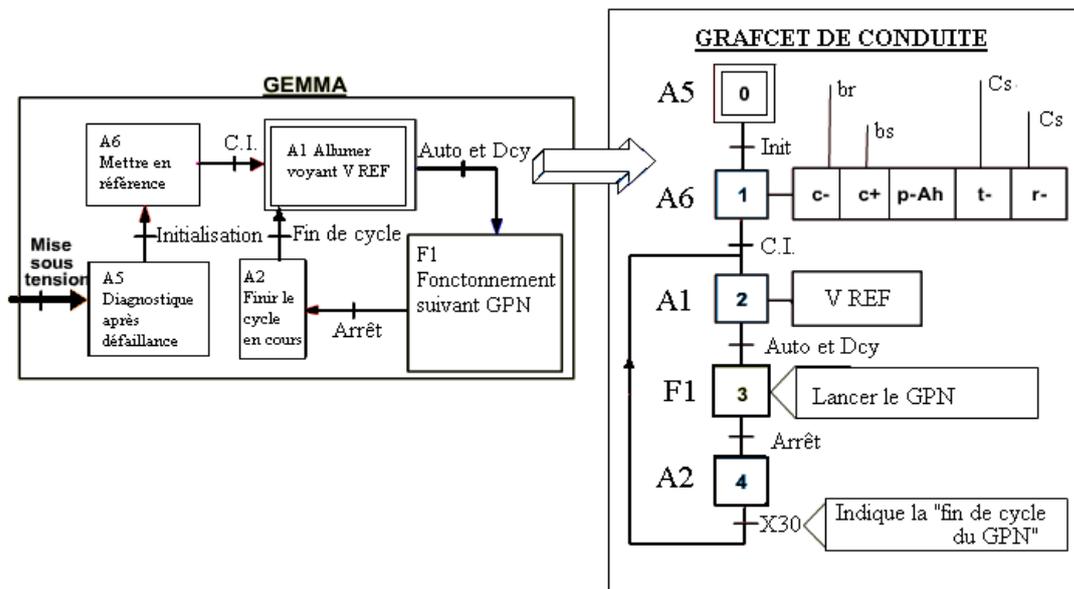


Figure.4.20. Elaboration du Grafcet de conduite

##### 4.8.1.2. Enrichissement du Grafcet :

Lorsque le GRAFCET de niveau 2 et le GEMMA sont simples, l'enrichissement du GRAFCET est une très bonne solution. Les effets de l'enrichissement d'un GRAFCET sont simplement l'ajout d'aiguillages et d'étapes supplémentaires au GRAFCET de niveau 2. Ainsi, supposons un GEMMA dans lequel une marche de clôture est prévue et que le

GRAFCET de niveau 2 de la production normale se termine ainsi avant d'appliquer le GEMMA :

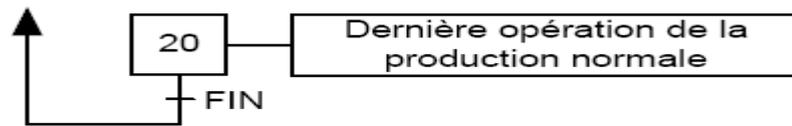


Figure.4.21. GRAFCET de niveau 2 de la production normale

Dans le GEMMA, la marche de clôture se fait si une condition (identifiée par la variable «CLO») est remplie :

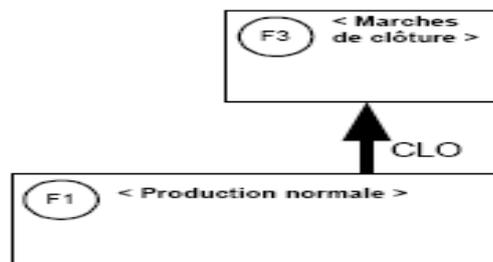


Figure.4.22. la marche de clôture se fait si une condition « la variable CLO »

Si CLO est inactif, la machine reste en production normale. Sinon, la machine exécute sa marche de clôture. Ce qui se traduit par ce nouveau GRAFCET :

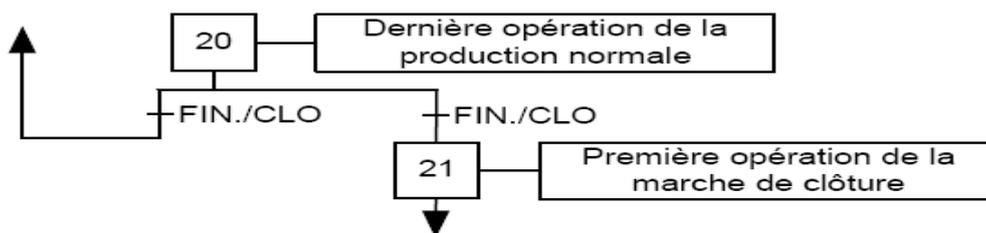


Figure.4.23. la condition CLO dans les deux cas, inactif et actif

Il y a donc création d'un aiguillage correspondant à la décision entre : entamer le début de la marche de clôture ou poursuivre la production normale.

Si le GRAFCET de départ est compliqué ou si le GEMMA est complexe, cette solution est moins praticable car le GRAFCET résultant devient illisible.

#### 4.8.1.3. Découpage en tâches

L'utilisation de tâches permet de simplifier le passage du GRAFCET de niveau 2 et du GEMMA à un ensemble de GRAFCET finaux, chacun ayant une fonction bien définie. Cela fait en sorte de ne pas avoir tout en un seul GRAFCET illisible.

Une fois ces tâches créées, il faut évaluer les conditions d'évolution entre elles. En effet, il est nécessaire d'éviter les problèmes dus à un manque de coordination entre les tâches.

Pour ce faire, on peut coordonner ces tâches de deux façons :

- Coordination horizontale ;
- Coordination verticale (ou hiérarchisée).

❖ **Coordination horizontale :**

Cette technique de coordination entre les tâches est utilisée lorsque les conditions suivantes sont remplies :

- Aucune de ces tâches n'est prééminente ;
- Le nombre de tâches est réduit ;
- Les liaisons entre les tâches sont réduites.

En coordination horizontale, une seule tâche à la fois peut être active. On compare cela avec une course relais, ou un coureur à la fois porte le relais.

Pour montrer ce qu'il en est, prenons le GRAFCET de niveau 2 initial suivant :

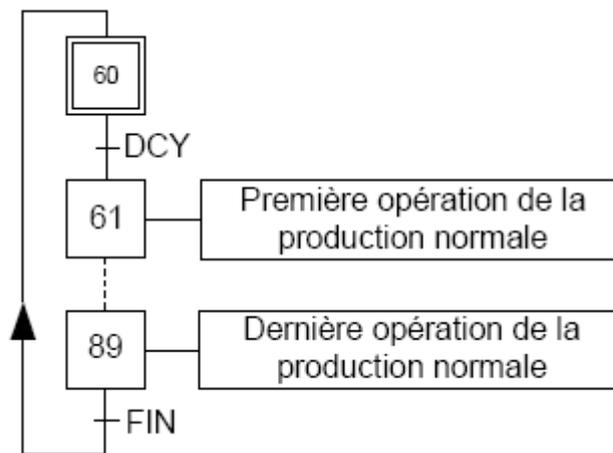


Figure.4.24. GRAFCET de niveau 2 initial

Combiné avec le GEMMA suivant :

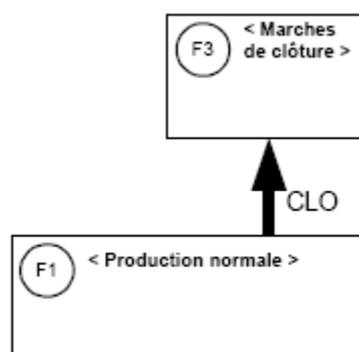


Figure.4.25. GRAFCET de niveau 2 initial combiné avec le GEMMA

Ce qui permet d'obtenir le grafcet découpé en tâches suivant :

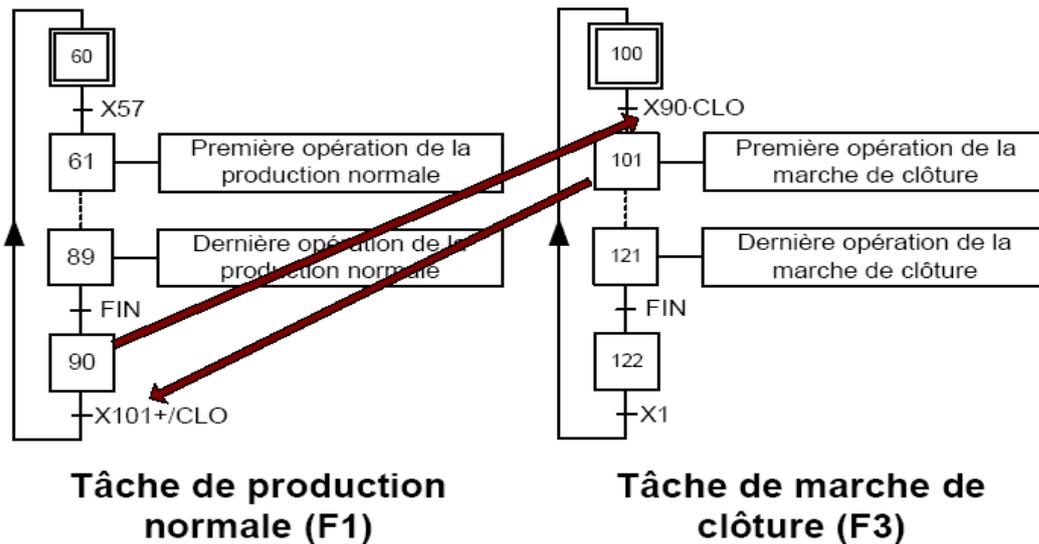


Figure.4.26. GRAFCET découpé en tâches

La tâche de production normale (F1) suit la tâche de la marche de préparation (F2) non montrée ici. Lorsque la tâche F2 est terminée, elle démarre la tâche F1.

Lorsque le capteur « CLO » est au niveau logique 1 et que la tâche F1 est à l'étape 90 (tâche F1 complétée), la tâche de la marche de clôture (F3) démarre. Cette tâche passe à l'étape 101, ce qui retourne la tâche F1 en étape d'attente (étape 60).

L'étape 122 marque la fin de la tâche F3 et déclenche une autre tâche non montrée.

L'exemple montre bien l'enchaînement, la coordination horizontale entre la tâche F1 et la tâche F3.

❖ **La coordination verticale :**

Cette technique utilise des tâches disposées selon une structure hiérarchique. Des tâches de niveau hiérarchique supérieur gèrent un ensemble de tâches de niveau hiérarchique inférieur. Pour un automatisme donné, il peut y avoir plusieurs niveaux. Poussé à sa limite, le GEMMA génère un GRAFCET de niveau supérieur et les rectangles-états génèrent des GRAFCET de niveau inférieur. Si on applique cette approche à l'exemple précédent, on aurait les GRAFCET suivants :

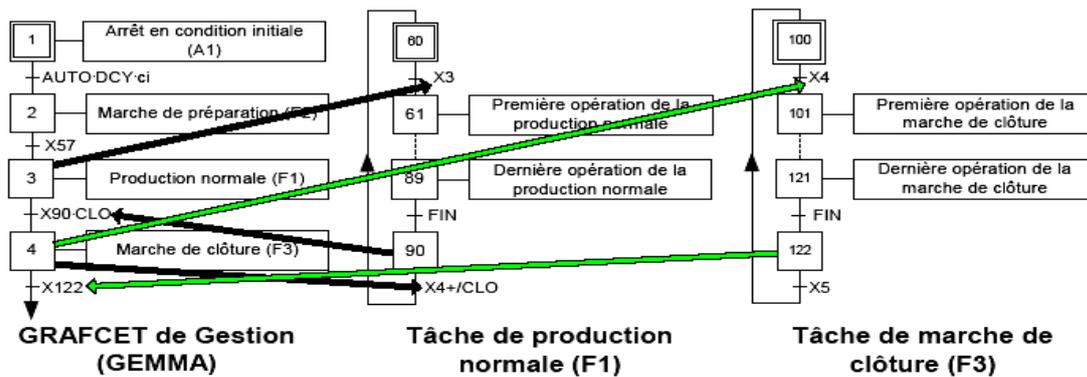


Figure.4.27. GEMMA génère un GRAFCET de niveau supérieur

Dans cet exemple, chaque étape du GRAFCET de Gestion correspond à un rectangle-état du GEMMA. Ce GRAFCET est le maître du système. Les autres GRAFCET montrés sont deux des GRAFCET esclaves de niveau hiérarchique inférieur. Lorsque le GRAFCET de Gestion se trouve à l'étape 3 il démarre le GRAFCET de production normale (F1). Celui-ci tourne tant que le GRAFCET de Gestion reste à l'étape 3. Le signal «CLO» permet le passage du GRAFCET de Gestion à l'étape 4.

La tâche F1 est alors remise en attente (étape 60) et la tâche de la marche de clôture (F3) est démarrée. Lorsque la marche de clôture est terminée (étape 122), le GRAFCET de Gestion passe à l'étape suivante (peut-être l'étape 5) et remet la tâche F3 en attente.

#### 4.8.1.4. Cas de l'arrêt d'urgence :

L'arrêt d'urgence ne peut être considéré comme une entrée quelconque. Entre autres, parce que l'arrêt d'urgence doit être câblé. Une autre raison est l'exemple d'enrichissement suivant ou l'arrêt d'urgence se traduit par une séquence d'urgence à exécuter :

Si le GRAFCET comporte un grand nombre d'étapes, cette approche devient rapidement illisible. Pour cette raison, l'arrêt d'urgence doit être traité de façon particulière pour pouvoir avoir une représentation beaucoup plus lisible

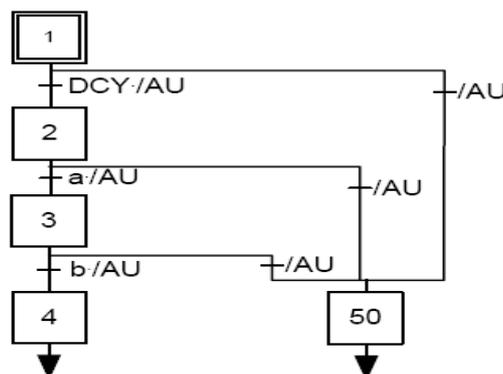


Figure.4.28. GRAFCET comporte le cas de l'arrêt d'urgence

Dans un automatisme, on peut retrouver deux types d'arrêts d'urgence :

- Les arrêts d'urgence sans séquence d'urgence ;
  - ❖ Arrêt par inhibition des actions ;
  - ❖ Arrêt figeage.
- Les arrêts d'urgence avec séquence d'urgence.
  - ❖ Tâche d'arrêt d'urgence ;
  - ❖ Symbole de regroupement.

**4.8.1.5. Les arrêts d'urgence par inhibition des actions :**

Lors d'un arrêt d'urgence, on peut simplement désirer que toutes les actions soient bloquées, inhibées. Donc, toutes les actions doivent être remises au niveau logique 0 lors de l'arrêt d'urgence. Cela correspond à l'écriture suivante :

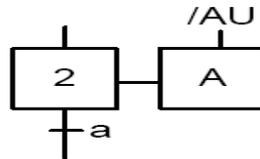


Figure.4.29. arrêts d'urgence par inhibition des actions

L'action A est donc conditionnelle à l'absence de l'arrêt d'urgence. Cette condition devrait être ajoutée à toutes les actions du GRAFCET.

**4.8.1.6. Les arrêts d'urgence par figeage :**

Certains arrêts d'urgence ont comme effet de figer le GRAFCET pour en empêcher son évolution. Ce figeage est obtenu en écrivant des réceptivités conditionnelles à l'absence de l'arrêt d'urgence. L'arrêt d'urgence entraîne un blocage du GRAFCET par non-franchissement des transitions puisque les réceptivités seront fausses. Voici un exemple d'écriture :

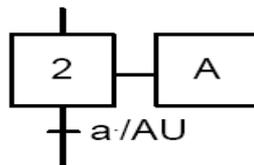


Figure.4.30. arrêts d'urgence par figeage

Ainsi, si à l'étape 2 la condition « a » est vraie et qu'il n'y a pas d'arrêt d'urgence, le GRAFCET passera à l'étape suivante. Par contre lors d'un arrêt d'urgence, la réceptivité reste fausse quelque soit « a », ce qui bloque le GRAFCET à l'étape 2.

#### ***4.8.1.7. Les arrêts d'urgence utilisant une tâche spéciale :***

Dans certains cas, il faut que l'arrêt d'urgence suive une certaine séquence pour limiter les dégâts et faire certains dégagements. La sécurité d'une telle séquence doit être bien évaluée pour s'assurer qu'il n'y a aucun risque pour le personnel et le matériel.

## Références bibliographiques

---

1. Jutard, A., & Betemps, M. (1997). *Système de commande. Support de cours en systèmes automatisés [en ligne], INSA de Lyon.*
2. Mabrouk, M. (1996). *Proposition d'une méthode et d'un outil d'aide à la reconfiguration des systèmes automatisés de production (Doctoral dissertation, Valenciennes).*
3. Brahami, Y., Boukandoul, W., & Yahiaoui, B. (2017). *Etude de la commande par un automate programmable industriel d'un compacteur tubolaire Bianco au sein de l'entreprise ICOTAL (BEJAIA) (Doctoral dissertation, Université A/Mira de Bejaia).*
4. Moreno, S., & Peulot, E. (1996). *Le GRAFCET: conception, implantation dans les automates programmables industriels: bac STI, STS, IUT, IUFM, écoles d'ingénieurs, formation continue. Educavivre.*
5. Michel, G., & Girard, B. (1988). *Les API: architecture et applications des automates programmables industriels. Dunod.*
6. BERTRAND, M. (2010). *Automates programmables industriels.*
7. Morais, A., & Visser, W. (1987). *Programmation d'automates industriels: adaptation par des débutants d'une méthode de spécification de procédures automatisées. Psychologie française, (32), 253-259.*
8. DUMÉRY, J. J. (2007). *GRAFCET-Structuration des descriptions. Applications.*
9. Madi, W., Iaichouchen, A., & Megri, S. (2017). *Etude et simulation d'un API S7-300 et programmation d'une centrifugeuse discontinue (Doctoral dissertation, Université Abderrahmane Mira).*
10. Bertrand, M. (2001). *Automates programmables industriels. Techniques de l'ingénieur. Informatique industrielle, 3(S8015), S8015-1.*
11. P. Le Brun, *Automates programmables, 1999.*
12. Frederic P. Miller, Agnes F. Vandome, John McBrewster, *Automates Programmables Industriels : Programmation informatique, Automatique, Industrie, Programmation (informatique), Interrupteur, Automaticien, Edition Alphascript Publishing 2010.*