



**P r o d u c t i o n   i n d u s t r i e l l e  
a u t o m a t i s é e   e t   i n f o r m a t i s é e**

---

**Chapitre 8**  
**Logique, interfaces et sécurité**  
**des systèmes automatisés**

---

**C**e chapitre 8 est celui, non pas des technologies, mais des techniques d'automatisation. Une chose est l'ensemble des technologies utilisées pour réaliser un système automatisé, une tout autre chose est, en amont, l'ensemble des techniques utilisées pour le concevoir et en garantir le fonctionnement ; et ce, quelle que soit la technologie qui le matérialise.

En effet, on dispose pour matérialiser un système automatisé de plusieurs technologies. La mécanique avec ses bons vieux programmeurs à cames et l'électromécanique avec ses relais sont aujourd'hui presque plus utilisées sans avoir pour autant complètement disparu. La fluidique, pour la part qui lui revient, reste pertinente pour son immunité aux parasites électromagnétiques et pour éviter la conversion d'énergies sur des machines pneumatiques. Mais la très grande majorité des systèmes automatisés sont aujourd'hui réalisés à partir de solutions électroniques et informatiques.

Si les technologies de réalisation sont diverses, les techniques de conception des systèmes automatisés le sont beaucoup moins. Il s'agit de concepts, de modes de représentation et de types particuliers de raisonnement. Il s'agit de la façon dont sont représentées, converties et traitées des informations analogiques, logiques et numériques. Il s'agit de la logique des automatismes et de son écriture. Il s'agit de la façon dont sont conçus, non seulement les réseaux de communication, mais aussi les interfaces entre des systèmes différents et entre l'homme et ces systèmes. Il s'agit enfin, après le GRAFCET<sup>1</sup> qui décrit le fonctionnement d'un système automatisé, d'un autre outil graphique, le GEMMA, qui garantit la sécurité de ce fonctionnement.

---

<sup>1</sup> Etudié au chapitre 7 sur la *Conception des systèmes automatisés*

## Logique, interfaces et sécurité des systèmes automatisés

### Table des matières du chapitre 8

1. L'information et son support .....	5
1.1. Informations logiques, numériques et analogiques.....	5
1.2. Conversion Numérique-Analogique (NA) / Analogique-Numérique (AN) .....	6
1.3. Acquisition de données : de la grandeur physique à l'information traitable .....	8
1.4. Traitement des données : l'Automate Programmable Industriel (API) .....	9
2. Les Interfaces Homme-Machine (IHM) et Machine-Machine (IMM).....	11
2.1. Dialogues avec les systèmes automatisés de fabrication .....	11
2.2. Supervision de la fabrication dans un système productique .....	13
2.3. Niveau 1 & 2 : partie commande et réseau local industriel .....	15
3. La logique et son écriture.....	17
3.1. Les propositions logiques .....	17
3.2. Les variables logiques.....	20
3.3. Les opérateurs logiques de base.....	21
3.4. Les fonctions logiques combinatoires.....	22
3.5. Les deux théorèmes d'Auguste de Morgan .....	23
3.6. Les fonctions logiques séquentielles.....	24

4. Sécurité des systèmes automatisés.....	25
4.1. Guide d'Etudes des Méthodes de Marche et d'Arrêt (GEMMA – ADEPA 1981) .....	25
4.2. Sélection des modes nécessaires de marche et d'arrêt.....	30
4.3. Détermination des conditions de passage d'un mode à l'autre.....	31
4.4. Les trois boucles fonctionnelles du GEMMA .....	34
4.5. La localisation statistique des défaillances matérielles.....	36
4.6. Les notions de fonctionnement normal et de fonctionnement en mode dégradé.....	37
Conclusion .....	39

---

# 1. L'information et son support

Il s'agit ici, dans un premier temps, de décliner et de caractériser les trois formes analogique, logique et numérique que peut prendre l'information traitée par la partie commande d'un système automatisé et de montrer comment on passe d'une forme à l'autre. Puis, dans un second temps, il s'agit de montrer comment la partie commande d'un système automatisé acquiert puis traite des données pour générer les ordres que doit exécuter la partie opérative d'un système automatisé.

## 1.1. Informations logiques, numériques et analogiques

Pour pouvoir, dans le déroulement d'une séquence automatisée, prendre une décision (lancement d'une tâche, exécution d'une action programmée, vérification d'un résultat, etc.), il est nécessaire de disposer, à tout instant, d'informations sur l'état du système et de son environnement. Deux cas doivent être distingués selon que la responsabilité de la décision relève d'une intervention de l'opérateur ou qu'elle relève d'une information générée par la partie commande.

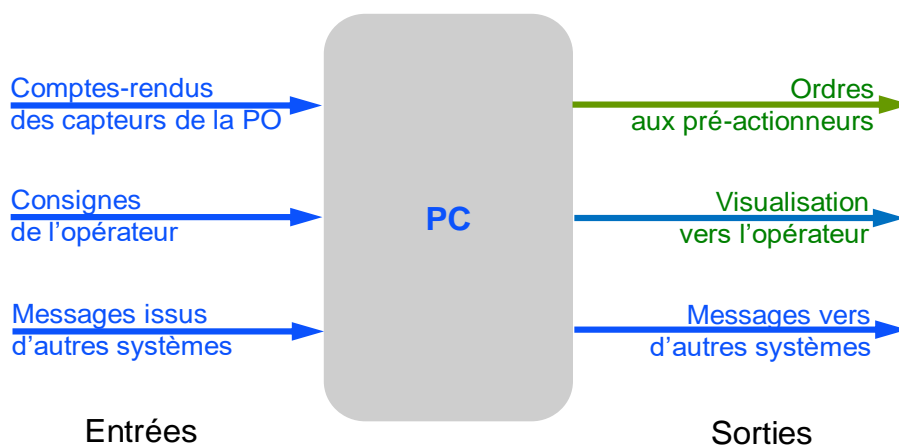
### La décision relève d'une intervention de l'opérateur.

Pour intervenir, l'opérateur doit pouvoir disposer d'informations sensorielles (visuelles, auditives, etc.) reçues directement ou le plus souvent par l'intermédiaire de capteurs qui transmettent ces informations sous une forme visuelle (voyants, indicateurs à aiguille, messages sur écrans et afficheurs, etc.) ou sonore (avertisseurs, messages vocaux, etc.). A partir de ces informations, l'opérateur prend des décisions qu'il traduit, pour le système, sous forme de consignes ou d'actions à exécuter.

### La décision relève d'une information générée par la partie commande.

La partie commande doit disposer, par l'intermédiaire des capteurs, d'informations sur l'état de la partie opérative et sur l'environnement du système, sur les consignes émises par l'opérateur, sur les messages éventuels communiqués par d'autres systèmes, etc.

En traitant ces informations, la partie commande devient capable de générer et d'émettre, en direction de la partie opérative, les ordres d'exécution des gestes physiques.



## Les trois catégories d'informations

Selon le besoin ou l'usage qui en est fait, les informations-source se rangent dans trois catégories.

- Les **informations logiques**, ce sont celles correspondant à des données de type « vrai ou faux » telles que « *la porte du palettiseur est fermée* », « *un carton est arrivé en fin de convoyeur* », etc.
- Les **informations numériques**, ce sont celles correspondant à des données plus intégratrices ou discriminantes telles qu'une quantité (*nombre de flacons remplis*) ou une position dans un rang (*ordre de rangement*), etc.
- Les **informations analogiques**, ce sont celles correspondant au suivi permanent d'une grandeur physique (*l'enregistrement de la température d'un four*, par exemple) se traduisant par une information analogique par opposition à logique.

Information logique (deux états)	Information numérique (états dénombrables)	Information analogique (état d'une grandeur)
<ul style="list-style-type: none"> <li>• Présence/Absence</li> <li>• Vrai/Faux</li> <li>• Fait/Non-fait</li> <li>• ...</li> </ul>	<ul style="list-style-type: none"> <li>• Rang dans une liste</li> <li>• Cardinal d'un ensemble</li> <li>• Note d'un devoir</li> <li>• ...</li> </ul>	<ul style="list-style-type: none"> <li>• Valeur d'une grandeur</li> <li>• Vitesse de déplacement</li> <li>• Température d'un four</li> <li>• ...</li> </ul>

### 1.2. Conversion Numérique-Analogique (NA) / Analogique-Numérique (AN)

Les informations pouvant revêtir les trois formes précédentes, il est très souvent nécessaire de convertir la représentation d'une de ces trois formes dans l'autre ; et ce, aussi bien dans un sens que dans l'autre. C'est ce que font les conversions NA/AN (Numérique/Analogique et Analogique/Numérique) qui transforment un signal électrique d'une forme de représentation à l'autre et réciproquement.

#### La forme initiale est analogique.

Chacune des grandeurs physiques (*la température d'un four à réguler*, par exemple) intervenant dans le traitement automatisé est prélevée par un capteur ou senseur qui la transforme en une autre grandeur physique, électrique celle-là, qui lui est proportionnelle.

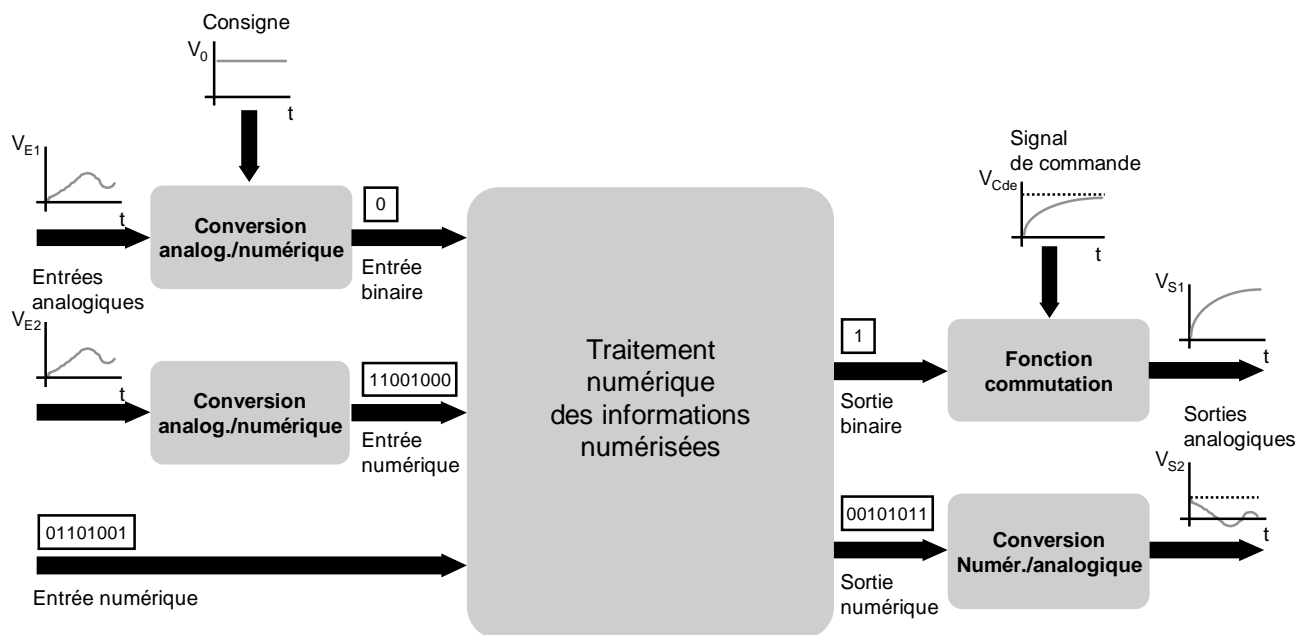
Il y a donc nécessité d'effectuer la conversion Analogique/Numérique de cette grandeur électrique en vue de son traitement automatisé généralement en logique binaire.

### La forme initiale est numérique.

Chacune des informations résultant du traitement automatisé est exprimée par un nombre généralement en code binaire. Ce nombre comporte d'autant plus de chiffres que la traduction doit être précise.

Il y a donc nécessité d'effectuer la conversion Numérique/Analogique de cette grandeur à la sortie de l'automate en vue probablement de piloter les actionneurs de la partie-opérative du système automatisé.

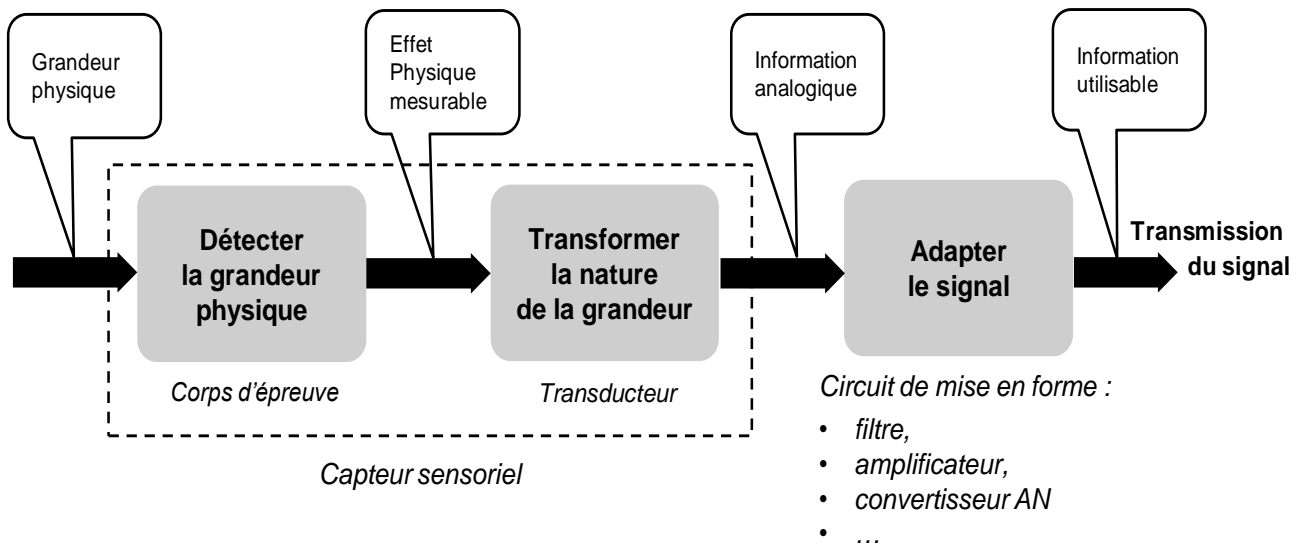
Le schéma suivant résume les différentes conversions que peuvent subir les grandeurs ou signaux d'entrée et de sortie de l'automate chargé du traitement numérique des informations numérisées.



Remarque :

Il convient de noter que la conversion Analogique/Binaire est un cas particulier de la conversion Analogique/Numérique. C'est ce qui se passe par exemple lorsque que l'on veut comparer deux grandeurs analogiques (*la température effective d'un four et une température de consigne, par exemple*). Ces deux grandeurs  $V_{E1}$  et  $V_{E2}$  sont appliquées aux deux entrées d'un comparateur dont la sortie passe de l'état [0] à l'état [1] quand, par exemple,  $V_{E2}$  devient supérieure à  $V_{E1}$  et, réciproquement, retourne à l'état [0] quand  $V_{E2}$  redevient inférieure à  $V_{E1}$ .

### 1.3. Acquisition de données : de la grandeur physique à l'information traitable



**La fonction détecter la grandeur physique est assurée par le corps d'épreuve.**

On désigne ainsi l'objet transformé sur le poste de travail qui, subissant directement l'action d'une grandeur physique, produit, à son tour, un effet physique mesurable.

Cet effet génère, à son tour, une information analogique sous la forme d'un signal mécanique, pneumatique ou électrique. Une fois mis en forme, ce signal constitue une donnée qui, saisie par l'automate, va être traitée par celui-ci.

Exemples :

- La pression d'un fluide déforme la membrane d'un capteur qui génère un signal électrique dont l'amplitude est proportionnelle à la pression du fluide.
- L'augmentation de la température dilate une tige métallique dont l'extrémité est solidaire du curseur d'un potentiomètre.

**La fonction transformer la grandeur physique est assurée par un transducteur.**

On désigne ainsi, par exemple, le composant qui produit un signal proportionnel à la grandeur physique. Ce signal est généré dans une énergie utilisable par les moyens de traitement dont dispose l'automate.

Par exemple ...

- ... dans le contact à pression utilisé sur un palettiseur, la pression d'air comprimé produit un déplacement de la membrane (le corps d'épreuve). Ce déplacement assure la commutation d'un signal électrique par la fermeture d'un contact électrique. Ce contact à pression joue ici le rôle de transducteur.

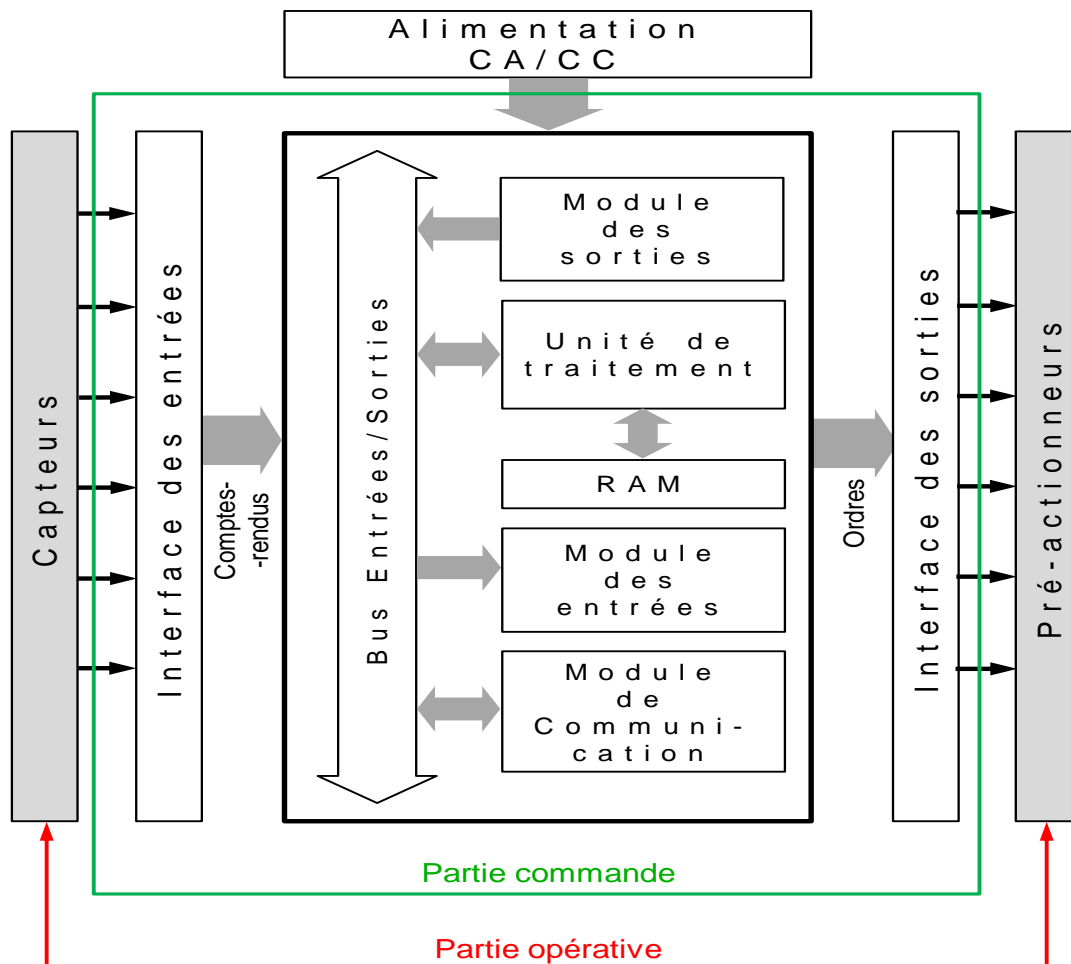
Le signal produit par un transducteur nécessite souvent une mise en forme qui consiste le plus souvent à filtrer les signaux pour en éliminer les parasites ou à en modifier la forme selon l'usage qui en sera fait par la suite. Par exemple ...

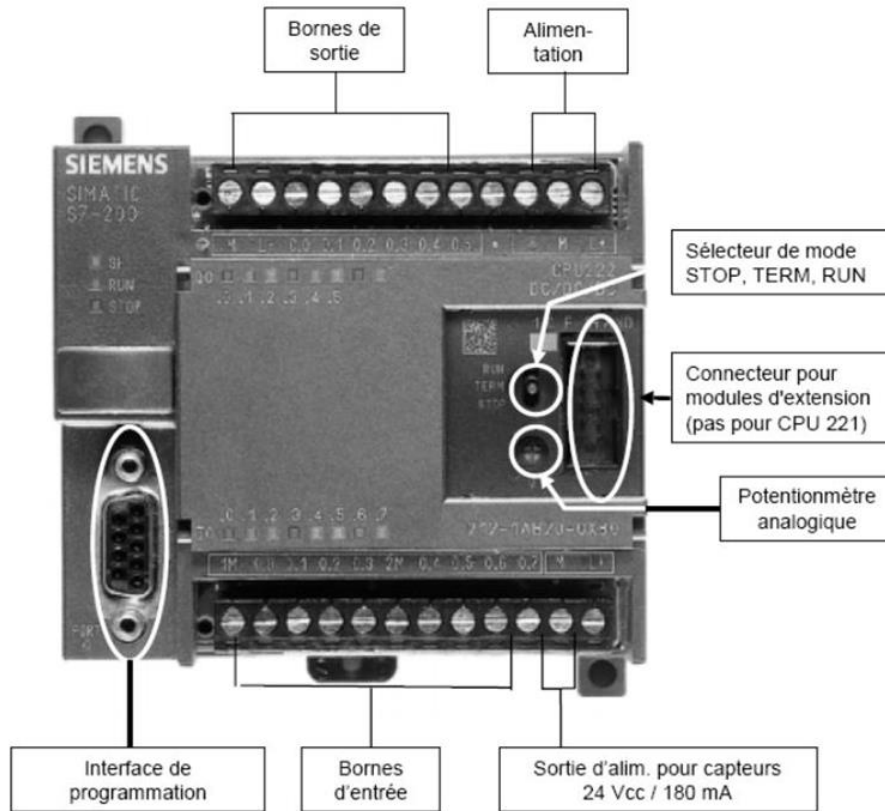
- ...le capteur d'ensoleillement d'un store est constitué d'un phototransistor dont le signal de sortie, proportionnel au flux lumineux reçu, est comparé à un niveau de seuil au-delà duquel le store doit être déployé.

#### 1.4. Traitement des données : l'Automate Programmable Industriel (API)

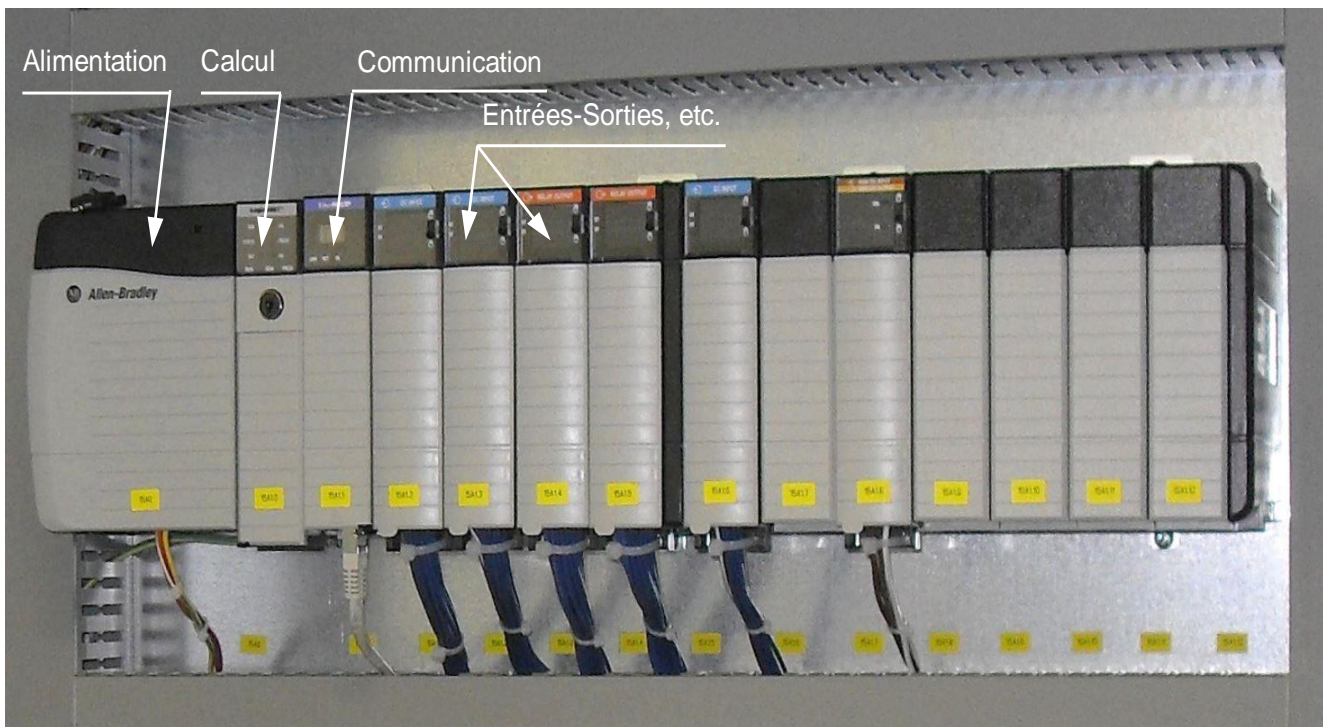
Un automate programmable est en quelque sorte un ordinateur dédié au pilotage de la partie opérative d'un système automatisé. Dépourvu d'un écran et d'un clavier permanents, cet ordinateur est toutefois un peu plus que la simple partie commande d'un système automatisé car il assure plusieurs fonctions :

- de **conversion en informations numérisées des signaux analogiques** envoyés par les capteurs de la partie opérative ;
- de **traitement des informations numérisées** pour générer les ordres à faire exécuter par les actionneurs de la partie opérative ; c'est sa fonction principale ;
- de **conversion de puissance** pour déclencher le mouvement des actionneurs physiques de la partie opérative ;
- de **communication avec des serveurs et un ordinateur de supervision** auxquels il est connecté par un réseau local industriel.





NB : automate programmable industriel pris chez Siemens



NB : automate programmable industriel pris chez Allen Bradley

## 2. Les Interfaces Homme-Machine (IHM) et Machine-Machine (IMM)

Comme on l'a vu au chapitre 6 sur l'organisation et l'implantation de la fabrication, tout site de production industrielle automatisée comporte généralement plusieurs systèmes automatisés qui effectuent les différentes opérations figurant sur la gamme de fabrication d'un ou de plusieurs produits. Ces différents systèmes (machines spéciales, MOCN, robots manipulateurs, systèmes de convoyage ou transfert, ...) ne fonctionnent évidemment pas indépendamment les uns des autres. Ils sont au contraire généralement interconnectés sur un même Réseau Local Industriel (RLI) et soumis à une unique supervision en charge du pilotage de l'ensemble du site.

Chaque système automatisé est doté d'une double interface :

- bien sûr une interface Machine-Machine qui lui permet de se synchroniser avec les autres systèmes du site,
- mais aussi une interface Homme-Machine qui permet à l'opérateur d'intervenir pour le mettre en route, le programmer, le régler, le surveiller, le maintenir en mode nominal ou le mettre en mode dégradé de fonctionnement, etc.

Il s'agit donc ici de s'intéresser singulièrement aux différents dialogues que tout opérateur entretient avec des systèmes automatisés ; et ce, jusqu'à la supervision de l'ensemble des systèmes interconnectés sur le site de production industrielle automatisée.

### 2.1. Dialogues avec les systèmes automatisés de fabrication

Le dialogue avec les systèmes automatisés concerne, en fait, des personnes aux rôles très différents selon que l'on se situe dans la phase qui précède ou dans la phase qui suit leur mise en œuvre. Il s'agit :

- des personnels d'études, de réalisation et de mise au point dans la première phase ;
- des personnels d'exploitation et de surveillance, de réglage et de maintenance dans la seconde phase.

#### Dialogue de développement

S'agissant de la partie commande, sa programmation se fait sur le site, grâce à un terminal dit « de programmation », intégré ou non à l'automate programmable. Cette programmation peut être effectuée hors du site à l'aide d'un micro-ordinateur doté d'un logiciel de programmation ou d'assistance à la programmation. Dans ce dernier cas une liaison locale ou à distance est nécessaire pour le chargement du programme.

Les réglages initiaux de la partie opérative et des capteurs relèvent de la même logique qu'en phase d'exploitation lors des réglages et de la maintenance.

## Dialogue de diagnostic

S'agissant à la fois de la partie commande et de la partie opérative, il faut, pour poser un diagnostic de fonctionnement du système, pouvoir accéder aux modes de marche ou d'arrêt, aux modes d'essai ou de vérification du système. Cet accès est le plus souvent réservé à des personnels spécialisés qui, pour opérer, ont recours à une clef de déverrouillage dont ils sont les seuls à disposer.

Ces deux types de dialogues s'effectuent à partir d'éléments différents qui sont connectés, soit en permanence, soit momentanément, à la partie commande. Il s'agit :

- du **pupitre de commande** avec ses boutons, (M/A, sélecteur de modes de marche, boutons de commande individuelle des pré-actionneurs, « coup de poing » d'arrêt d'urgence, etc.) ;
- d'**éléments de visualisation** avec ses voyants présents sur la partie opérative (diodes électroluminescentes et voyants indiquant l'état physique des actionneurs et des capteurs) ;
- d'**appareils de mesure** éventuels installés à demeure (vue-mètres, manomètres, etc.) ou temporairement (contrôleurs, voltmètres, pinces ampèremétriques, etc.) ;
- d'un **terminal amovible** d'exploitation ou de diagnostic à brancher le plus souvent sur un API.



NB : console amovible d'exploitation à afficheurs XBT-A701-01 Schneider Télémécanique utilisé par l'opérateur pour saisir des données



NB : console amovible de programmation à afficheurs TSX-T407-1 Schneider Télémécanique utilisée par l'opérateur pour programmer l'automate

## Dialogue de conduite

Ce dialogue entre l'opérateur et sa machine ou son système s'opère à partir du pupitre ou de la console d'exploitation. Il concerne :

- la mise en route ou la marche éventuelle de préparation,
- la marche normale,
- la marche éventuelle de clôture,
- les cas d'arrêts particuliers ou de marche en mode dégradé.

Ce dialogue, qui consiste pour l'opérateur à élaborer des consignes et recevoir des comptes-rendus en provenance du système, passe par un ensemble de boutons, claviers, voyants et afficheurs.

## 2.2. Supervision de la fabrication dans un système productique

La supervision de la fabrication ne s'effectue pas qu'au niveau 1 du système productique, le niveau où se trouvent les parties commande des systèmes automatisés de fabrication. Elle s'effectue en fait sur au moins trois des cinq niveaux d'un système productique :

- au niveau 1 (réseaux locaux industriels 1<sub>A</sub> et 1<sub>B</sub>), **la supervision des systèmes automatisés,**
- au niveau 2 (réseau local 1), **la supervision de l'atelier de fabrication,**
- au niveau 3 (réseau local 2), **la supervision de l'usine de production.**

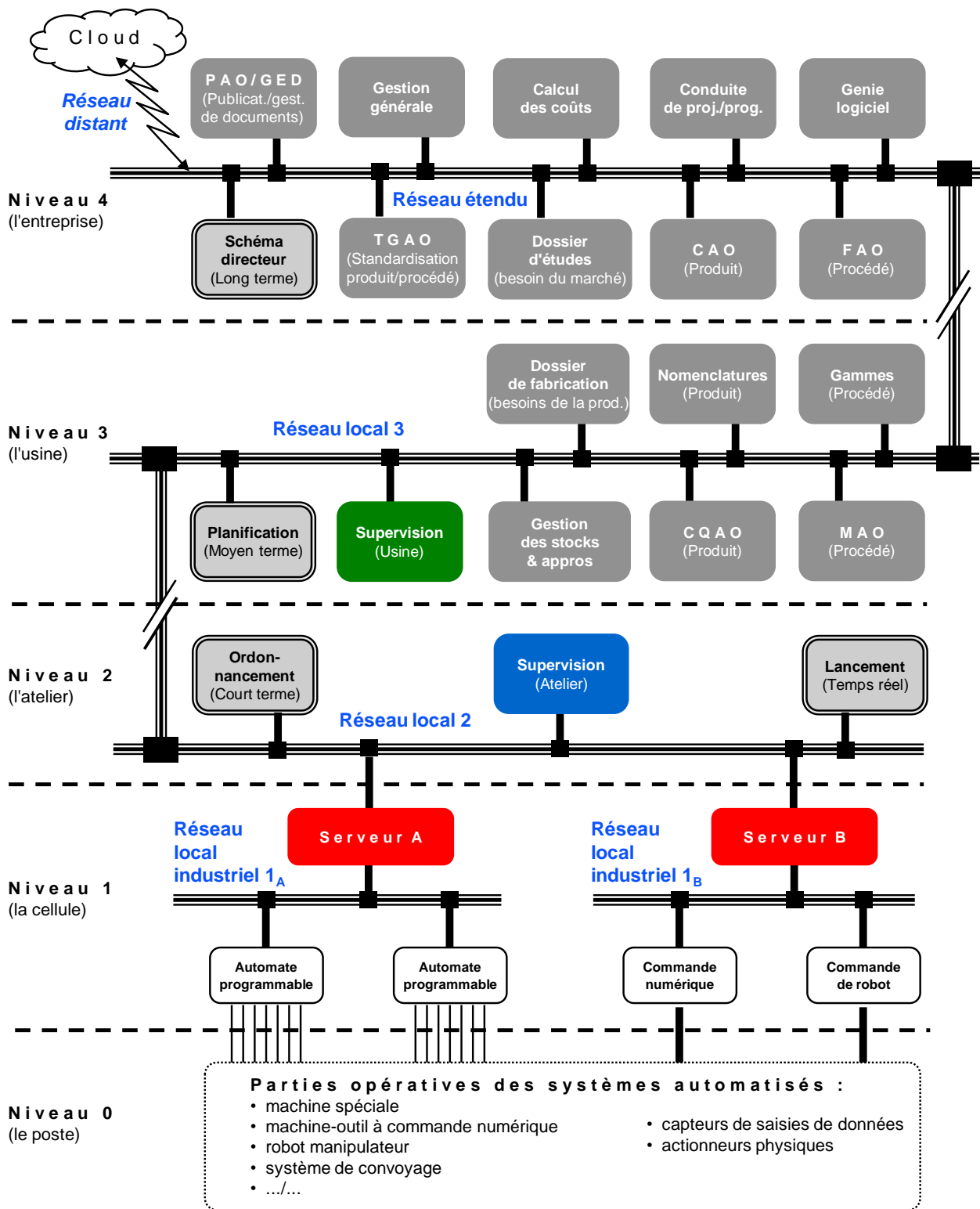
S'effectuant toujours sur un niveau donné d'un système productique, la supervision est assurée par un ordinateur doté des logiciels de pilotage, de coordination et de surveillance des systèmes présents sur ce niveau. Nécessairement doté d'un terminal de supervision, cet ordinateur :

- saisit des données aux différents niveaux du système productique,
- génère les ordres nécessaires aux systèmes supervisés sur le niveau pour exécuter les tâches qui leur sont assignées,
- coordonne l'exécution de ces différentes tâches,
- surveille le résultat de la bonne exécution de ces différentes tâches.

L'accès à un ordinateur de supervision est réservé aux responsables de la fabrication et de la gestion de la production.

On trouvera à la page suivante le schéma présentant l'architecture de tout système productique avec ces cinq niveaux de 0 à 4. Sur cette architecture, sont représentés :

- en rouge, les serveurs qui supervisent les systèmes automatisés au niveau 1,
- en bleu, l'ordinateur qui supervise l'atelier de fabrication au niveau 2,
- en vert, l'ordinateur qui supervise l'usine de production au niveau 3.



Au niveau 1 des parties commande des systèmes automatisés et au niveau 2 de l'atelier où se fait leur supervision, il est évidemment nécessaire de mettre en relation les machines ou les serveurs à coordonner et à surveiller. La liaison qui assure cette mise en relation est :

- d'abord de **nature matérielle** par les flux de matières premières et de produits semi-ouvrés qui transitent d'un poste de travail à l'autre ;
- ensuite de **nature informationnelle** pour la coordination et la surveillance des opérations de fabrication effectuées par chacun des systèmes automatisés.

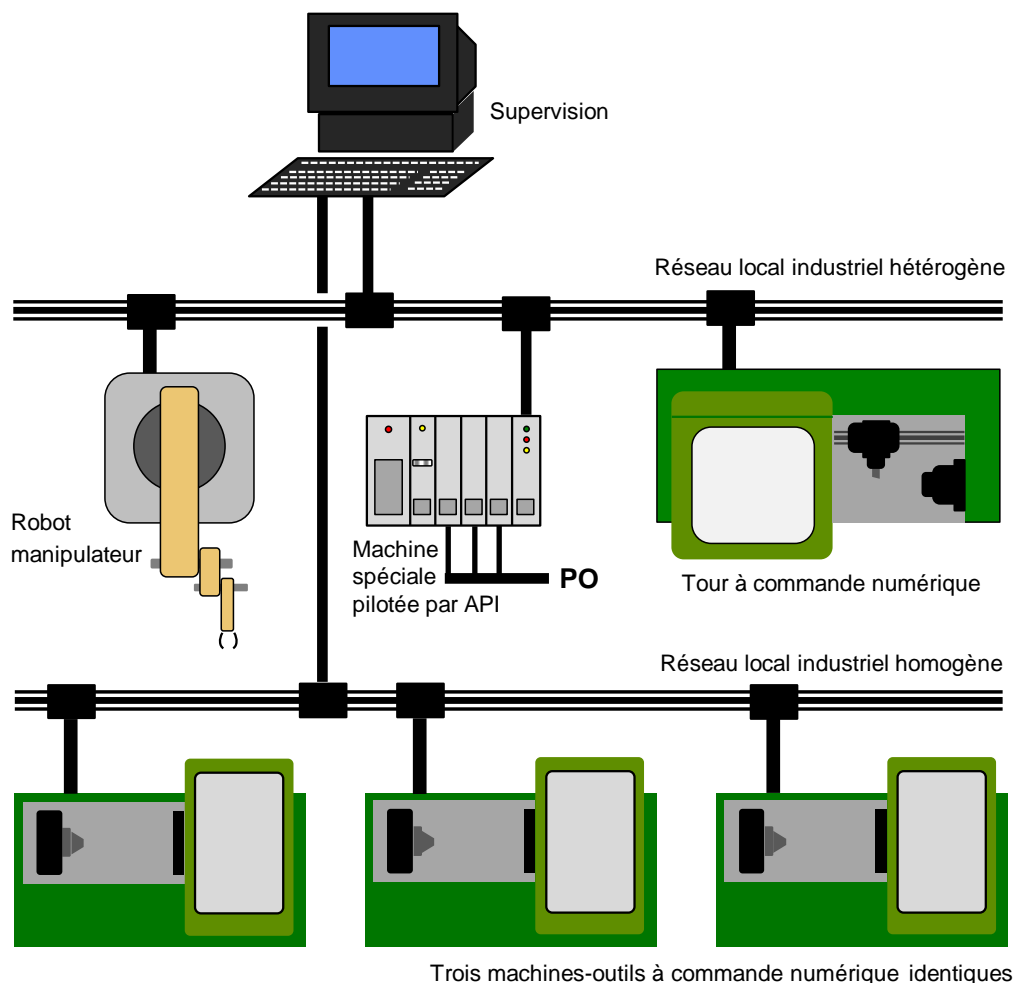
La liaison de nature informationnelle nécessite la mise en œuvre de moyens matériels et logiciels complexes. Il s'agit :

- au niveau 1, de la liaison des parties commande des systèmes automatisés à leur Réseau Local Industriel 1<sub>A</sub> ou 1<sub>B</sub> pour en assurer la conduite ;
- au niveau 2, de la liaison de chaque serveur A ou B à leur réseau local 2 pour assurer la supervision de l'atelier ;
- au niveau 3, de la liaison du réseau local 2 au réseau local 3 pour assurer la supervision de l'usine ;
- au niveau 4 de la liaison du réseau local 3 aux réseaux étendu (Intranet) et distant (*Cloud*) de l'entreprise pour assurer le pilotage du système de production.

### 2.3. Niveau 1 & 2 : partie commande et réseau local industriel

Un Réseau Local Industriel (RLI) met en relation les parties commandes des systèmes automatisés dont l'activité doit être coordonnée ; et ce, de la manière la plus simple et la plus transparente possible.

On distingue deux types de réseaux locaux industriels, les réseaux homogènes et les réseaux hétérogènes.



- Les **réseaux homogènes** mettent en communication des matériels de même nature, connectés directement sur le même câble par l'intermédiaire de simples coupleurs.
- Les **réseaux hétérogènes** mettent en communication des matériels de natures différentes (des automates programmables, des ordinateurs, des robots, des machines à commande numérique, etc.). Ces réseaux nécessitent la mise en œuvre d'interfaces de communication pour assurer la compatibilité des matériels et des logiciels devant communiquer entre eux.

La partie logique des interfaces coordonne les échanges d'un point de vue temporel en suivant un protocole qui assure la validité des données transmises.

### 3. La logique et son écriture

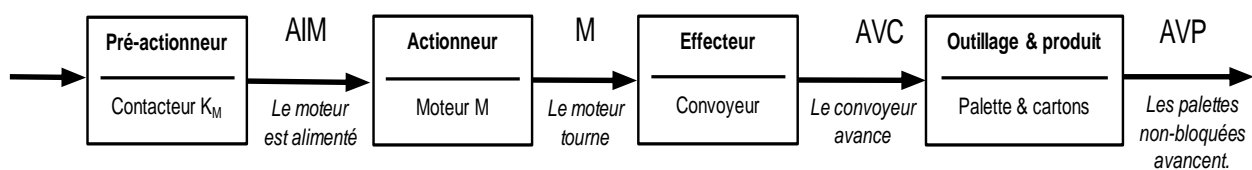
A différentes reprises au cours de ce chapitre et dans quelques-uns des chapitres précédents, a été évoquée la notion de logique. Il convient dans ce qui suit d'examiner très précisément la logique qui régit le fonctionnement des systèmes automatisés. Ce faisant nous allons montrer que cette logique est en fait un langage binaire qui n'est autre que l'**algèbre de Boole**.

Ce langage met en effet en relation des variables binaires qui ne peuvent prendre que deux valeurs 0 ou 1. Il soumet ces variables à trois opérateurs de base : l'opérateur NON de négation, l'opérateur ET de conjonction et l'opérateur OU de disjonction. Combinés entre eux, ces trois opérateurs donnent naissance à cinq fonctions logiques combinatoires NON-ET, NON-OU, Inhibition, OU exclusif et Identité. D'autres fonctions logiques complexes peuvent bien sûr être élaborées à partir des trois opérateurs de base et de ces cinq fonctions logiques combinatoires. Les mêmes règles s'appliquent à ces opérateurs de base et à ces fonctions logiques combinatoires ; ce sont de véritables règles de calcul, les règles de l'algèbre de Boole, partie intégrante des mathématiques.

#### 3.1. Les propositions logiques

Énoncer une propriété sur un objet, c'est formuler une phrase comme celles énoncées ci-dessous.

- Le bouton vert *Marche* est actionné.
- Le moteur *M* du convoyeur transporteur est sous tension.
- *Les palettes avancent*.
- La tige du vérin *V* d'arrêt des palettes est sortie : *la palette est arrêtée*.
- Le moteur *M* est arrêté.
- etc.



Ces énoncés sont des **propositions logiques** relatives à des objets auxquels elles font référence ; **le mot logique voulant ici dire que ces propositions sont soit vraies, soit fausses**.

Certaines propositions sont relatives :

- à des éléments de la partie commande (les boutons du pupitre de commande, par exemple) ;
- à des constituants de la partie opérative (un moteur, un vérin, par exemple) ;
- à des éléments de supervision du système productique global (un témoin d'alerte qui montre que la ligne 4 de fabrication dans l'atelier B fonctionne selon le mode dégradé n°3, par exemple).

Une proposition logique pouvant prendre deux valeurs, on convient, pour simplifier les notations, de donner un nom symbolique à chacune de ces deux valeurs : *vrai* ou *faux*, 1 ou 0, 0 ou 1, +1 ou -1 et *true* ou *false*. Par convention, on adopte tel ou tel couple de noms selon le type de logique choisi.

Proposition ...	Logique naturelle	Logique positive	Logique négative	Logique complémentaire	Logique informatique
... vraie	vrai	1	0	+1	true
... fausse	faux	0	1	-1	false

Cette convention étant adoptée, on note la valeur d'une proposition sous la forme indiquée ci-dessous.

- Ecrite en langage naturel, la valeur de la proposition *le moteur M tourne* s'écrira :

$$[\text{le moteur M tourne}] = \text{vrai (ou } = 1).$$

Ce qui ici signifie qu'effectivement le moteur tourne.

- Ecrite en langage naturel, la valeur de la proposition *les palettes non-bloquées avancent* s'écrira :

$$[\text{les palettes non-bloquées avancent}] = \text{faux (ou } = 0).$$

Ce qui ici signifie que les palettes sont arrêtées.

Parmi les propositions logiques [mise entre crochets] énoncées précédemment on constate que certaines semblent a priori liées par le fonctionnement que l'on pense être celui du système. Par exemple :

*Il est certain que le fait que la palette avance est lié au fait que le moteur M tourne et il est probable que le moteur tourne soit lié au fait que le bouton vert marche soit, ou ait été, actionné.*

Les propositions correspondantes sont liées par des relations de cause à effet, ou relations causales.

Les relations de cause à effet peuvent également être représentées par des propositions logiques, appelées propositions inductives ou inductions ; elles se présentent sous la forme générale :

$$[\text{SI ... (cause), ALORS ... (effet)}].$$

C'est l'écriture [SI ..., ALORS ...] qui exprime que, de la cause, on déduit l'effet.

L'induction [SI..., ALORS...] peut prendre quatre formes différentes selon le type de langage adopté : logique, mathématique, booléen ou informatique.

**Forme 1 : induction logique**

SI proposition a, ALORS proposition b

**SI a ALORS b****Forme 2 : implication mathématique**

Proposition a entraîne proposition b

**a  $\Rightarrow$  b****Forme 3 : algèbre de Boole**

Proposition a équivaut à proposition b

**a = b****Forme 4 : implication informatique**

Proposition b générée par la proposition a

**a  $\leftarrow$  b**

L'existence d'une **relation d'induction** exprime une règle de fonctionnement du système. Toute règle a bien sûr été voulue par le concepteur et la réalisation technique a permis de la concrétiser.

Deux exemples :

- L'induction [*SI le moteur M tourne, ALORS les palettes avancent.*] résulte de la conception de la partie opérative ; l'arbre du moteur M est, en effet, relié à une poulie qui entraîne le convoyeur sur lequel sont posées les palettes.
- L'induction [*le bouton « Marche » est actionné ET le bouton « Arrêt » n'est pas actionné ET la ligne moteur M est alimentée*] a été réalisée sur la partie commande du moteur par câblage et non par programmation de l'automate du système.

### 3.2. Les variables logiques

L'écriture littérale en français des propositions logiques est lourde et peu pratique. Aussi préfère-t-on remplacer ces propositions littérales par des expressions réduites qui résument l'état de l'objet ou son comportement observé. On appelle ces expressions : variables logiques.

Le tableau suivant propose deux formes de variables logiques correspondant à plusieurs propositions logiques.

- La première forme remplace la proposition par une expression logique composée d'un mot et d'un symbole.  
Ex : [*marche\_C*] = vrai si [*le bouton rouge « Arrêt » n'est pas actionné*] = vrai
- La deuxième forme remplace la proposition par une variable logique symbolique.  
Ex : [*M*] = faux si [*le moteur M tourne*] = vrai

Proposition logique	Variables logiques	
	1 <sup>ère</sup> forme	2 <sup>ème</sup> forme
[le bouton vert Marche est actionné]	marche_C	m
[le bouton rouge Arrêt n'est pas actionné]	arrêt_C	¬m
[la ligne de moteur M est sous tension]	tension_M	u
[le moteur M tourne]	tourner_M	M
[les palettes avancent]	avance_palettes	AVP

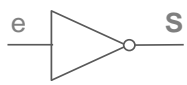
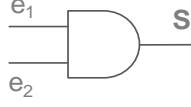
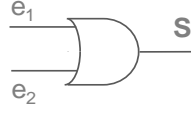
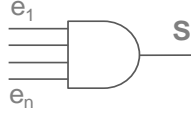
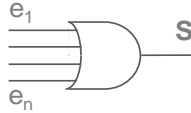
### 3.3. Les opérateurs logiques de base

Les propositions logiques, aussi complexes soient-elles, peuvent toutes s'exprimer à partir d'expressions combinant des propositions logiques élémentaires avec la conjonction de coordination, ET, la conjonction de disjonction, OU, et l'adverbe de négation, NON.

L'**algèbre de Boole** donne un fondement mathématique au langage logique naturel.

- Les propositions logiques élémentaires sont représentées par des variables logiques d'entrée des opérateurs :  $e$ ,  $[e_1]$ , ... et  $[e_n]$ .
- La proposition logique résultante est représentée par la variable de sortie  $S$  de chaque opérateur.
- Les opérations logiques de négation, de conjonction et de disjonction sont respectivement représentées par les opérateurs logiques de base : NON, ET et OU.
- Le modèle logique exprimant la relation entre des propositions logiques élémentaires,  $[e_1]$ ,  $[e_2]$ ,  $[e_3]$ , ..., et une proposition logique résultante  $[S]$  est représenté par une fonction logique.

Pour chacun de ces trois opérateurs logiques de base NON, ET et OU, le tableau suivant donne leur expression logique, leur représentation normalisée, leur table de vérité et leur forme généralisée.

Opérateurs logiques de base	NON (NO)	ET (AND)	OU (OR)																																				
Expression logique	$S = \bar{e}$	$S = e_1 \cdot e_2$	$S = e_1 + e_2$																																				
Représentation normalisée (1 à 2 entrées)																																							
Table de vérité (1 à 2 entrées)	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>e</td><td>S</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	e	S	0	1	1	0	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td><math>e_1</math></td><td><math>e_2</math></td><td>S</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	$e_1$	$e_2$	S	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td><math>e_1</math></td><td><math>e_2</math></td><td>S</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	$e_1$	$e_2$	S	0	0	0	0	1	1	1	0	1	1	1	1
e	S																																						
0	1																																						
1	0																																						
$e_1$	$e_2$	S																																					
0	0	0																																					
0	1	0																																					
1	0	0																																					
1	1	1																																					
$e_1$	$e_2$	S																																					
0	0	0																																					
0	1	1																																					
1	0	1																																					
1	1	1																																					
Forme généralisée (n entrées)		 <p>S à 1 si toutes les entrées à 1</p>	 <p>S à 1 si au moins 1 entrée à 1</p>																																				

### 3.4. Les fonctions logiques combinatoires

Les fonctions logiques, aussi complexes soient-elles, peuvent toutes s'exprimer à partir d'expressions combinant les trois opérateurs logiques de base NON de négation, ET de coordination et OU de disjonction.

Le tableau qui suit donne ces mêmes quatre éléments pour cinq **fonctions logiques combinatoires** particulières : NON-ET, NON-OU, Inhibition, OU exclusif et Identité. De telles fonctions logiques sont dites combinatoires parce que chacune d'elle est une combinaison particulière des trois opérateurs de base précédents comme le montre leurs expressions logiques respectives.

Fonctions logiques combinatoires	NON-ET (NAND)	NON-OU (NOR)	Inhibition	OU exclusif	Identité																																																																											
Expression logique	$S = \overline{e_1 \cdot e_2}$	$S = \overline{e_1 + e_2}$	$S = e_1 \cdot \overline{e_2}$	$S = e_1 \oplus e_2$ $S = e_1 \cdot \overline{e_2} + \overline{e_1} \cdot e_2$	$S = e_1 \odot e_2$ $S = e_1 \cdot e_2 + \overline{e_1} \cdot \overline{e_2}$																																																																											
Représentation normalisée (2 entrées)																																																																																
Table de vérité (2 entrées)	<table border="1"> <thead> <tr><th>e<sub>1</sub></th><th>e<sub>2</sub></th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	e <sub>1</sub>	e <sub>2</sub>	S	0	0	1	0	1	1	1	0	1	1	1	0	<table border="1"> <thead> <tr><th>e<sub>1</sub></th><th>e<sub>2</sub></th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	e <sub>1</sub>	e <sub>2</sub>	S	0	0	1	0	1	0	1	0	0	1	1	0	<table border="1"> <thead> <tr><th>e<sub>1</sub></th><th>e<sub>2</sub></th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	e <sub>1</sub>	e <sub>2</sub>	S	0	0	0	0	1	0	1	0	1	1	1	0	<table border="1"> <thead> <tr><th>e<sub>1</sub></th><th>e<sub>2</sub></th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	e <sub>1</sub>	e <sub>2</sub>	S	0	0	0	0	1	1	1	0	1	1	1	0	<table border="1"> <thead> <tr><th>e<sub>1</sub></th><th>e<sub>2</sub></th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	e <sub>1</sub>	e <sub>2</sub>	S	0	0	1	0	1	0	1	0	0	1	1	1
e <sub>1</sub>	e <sub>2</sub>	S																																																																														
0	0	1																																																																														
0	1	1																																																																														
1	0	1																																																																														
1	1	0																																																																														
e <sub>1</sub>	e <sub>2</sub>	S																																																																														
0	0	1																																																																														
0	1	0																																																																														
1	0	0																																																																														
1	1	0																																																																														
e <sub>1</sub>	e <sub>2</sub>	S																																																																														
0	0	0																																																																														
0	1	0																																																																														
1	0	1																																																																														
1	1	0																																																																														
e <sub>1</sub>	e <sub>2</sub>	S																																																																														
0	0	0																																																																														
0	1	1																																																																														
1	0	1																																																																														
1	1	0																																																																														
e <sub>1</sub>	e <sub>2</sub>	S																																																																														
0	0	1																																																																														
0	1	0																																																																														
1	0	0																																																																														
1	1	1																																																																														
Forme généralisée (n entrées)	 S = 1, si au moins 1 entrée à 0	 S = 1, si toutes les entrées à 0	 S = 1, si : entrée inhibée à 0 autres entrées à 1	 S = 1, si une et une seule entrée à 1	 S = 1 si toutes les entrées au même état																																																																											

### 3.5. Les deux théorèmes d'Auguste de Morgan

On doit au mathématicien britannique Auguste de Morgan, la formulation au 19<sup>ème</sup> siècle de deux théorèmes applicables aux expressions logiques comportant au moins deux variables. Ils s'énoncent en langage naturel et en langage mathématique.

Le premier théorème s'énonce ainsi :

- en langage naturel :

La négation de la conjonction de deux propositions équivaut à la disjonction des négations des deux propositions ; soit en d'autres termes :

**Non (e<sub>1</sub> Et e<sub>2</sub>) équivaut à Non e<sub>1</sub> Ou Non e<sub>2</sub> ;**

- en langage mathématique :

$$\overline{e_1 \cdot e_2} \Leftrightarrow \overline{e_1} + \overline{e_2}$$

Le second théorème s'énonce ainsi :

- en langage naturel :

La négation de la disjonction de deux propositions équivaut à la conjonction des négations des deux propositions ; soit en d'autres termes :

**Non (e<sub>1</sub> Ou e<sub>2</sub>) équivaut à Non e<sub>1</sub> Et Non e<sub>2</sub> ;**

- en langage mathématique :

$$\overline{\overline{e_1} + \overline{e_2}} \Leftrightarrow \overline{\overline{e_1} \cdot \overline{e_2}}$$

Ces deux théorèmes sont particulièrement utiles pour simplifier des expressions logiques complexes.

Remarque importante :

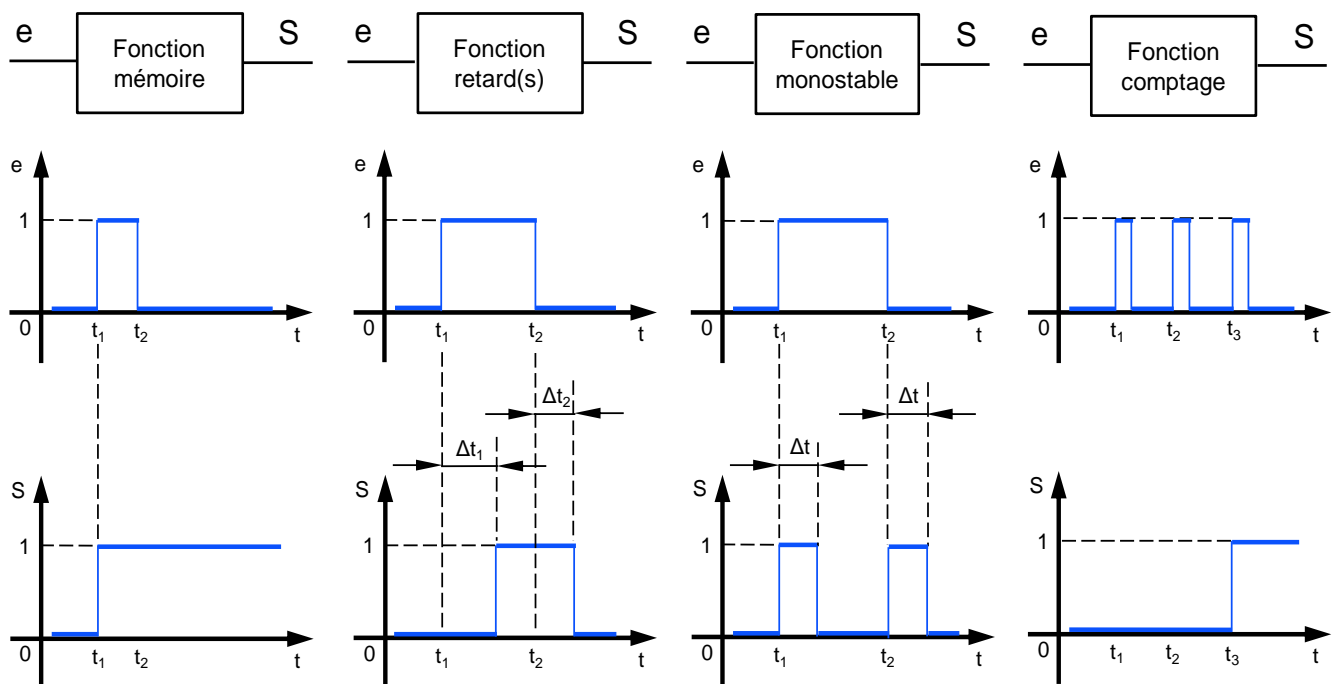
Ces deux théorèmes sont généralisables à n variables. En d'autres termes, appliqués ici à deux variables e<sub>1</sub> et e<sub>2</sub>, ces deux théorèmes s'appliquent de la même façon à n variables e<sub>1</sub>, ..., e<sub>n</sub>.

### 3.6. Les fonctions logiques séquentielles

Aux fonctions logiques combinatoires s'ajoutent **quatre fonctions logiques séquentielles**. On a recours à celles-ci chaque fois que le temps intervient dans le fonctionnement d'un système.

Ces quatre fonctions logiques séquentielles sont les suivantes.

- La **fonction mémoire** : un effet reste maintenu alors que la cause qui l'a produit a disparu.
- La **fonction retard ou temporisation** : l'état de la sortie dépend non seulement de l'état de l'entrée, mais aussi d'un décalage temporel entre la réaction de la sortie et le changement d'état de l'entrée.
- La **fonction monostable ou générateur d'impulsions** : le signal de sortie est émis pendant une durée déterminée qui démarre à l'instant du changement d'état de l'entrée.
- La **fonction comptage** : une sortie prend un état logique déterminé après qu'une séquence de valeurs d'entrée a été obtenue. Par exemple, dans un comptage par trois, la sortie S devient vraie quand l'entrée e est devenue vraie trois fois de suite.



#### Retard

Conservation de l'état en sortie après disparition de l'état en entrée

#### Temporisation

Délais différents entre les changements d'état en entrée et en sortie

#### Génération d'impulsions

Délai constant entre les changements d'état en entrée et en sortie

#### Comptage par 3

Changement d'état en sortie à la 3<sup>ème</sup> impulsion en entrée

## 4. Sécurité des systèmes automatisés

Une chose est la situation théorique selon laquelle un système automatisé parfaitement conçu et réalisé fonctionne de façon nominale, une tout autre chose est la réalité de tout système automatisé qui, même bien conçu et réalisé, peut ne plus être alimenté en matières premières, voir l'un de ses composants défaillir, présenter un bug dans sa programmation, etc. Bref, tout système automatisé, même bien conçu et parfaitement réalisé, a besoin de s'arrêter ne serait-ce que pour sa maintenance ou pour tenir compte des contraintes existant sur la ligne de fabrication, en amont ou en aval de la position qu'il occupe sur cette ligne. De plus, face à certaines des défaillances qu'il peut connaître, il peut être urgent, pour des raisons de sécurité, de mettre le système à l'arrêt. Et face à d'autres défaillances, il se peut que le système puisse ne pas être arrêté, mais mis en mode de fonctionnement dégradé, par exemple à un rythme ralenti ou avec une intervention humaine dans son cycle de fonctionnement.

Bref, on comprend que le GRAFCET ayant été établi et traduit en éléments technologiques réalisant autant sa partie opérative que sa partie commande, un système automatisé a aussi besoin que lui soit indiqué ce qu'il doit faire lorsque son fonctionnement sort de l'état nominal. Il en va de la sécurité de son fonctionnement et de la sécurité des opérateurs qui travaillent autour. C'est ce que va permettre le GEMMA, le Guide d'Etudes des Méthodes de Marche et d'Arrêt.

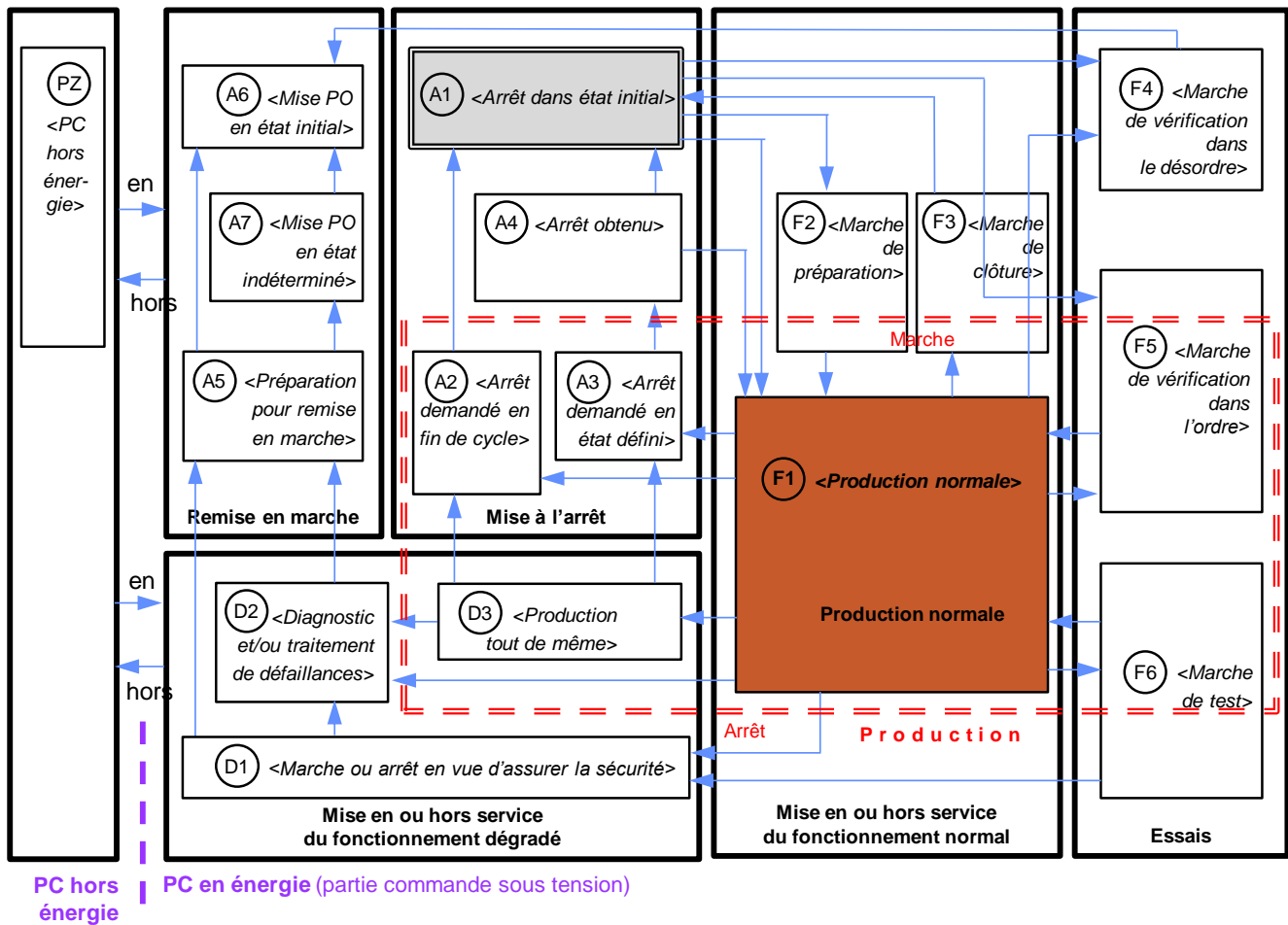
### 4.1. Guide d'Etudes des Méthodes de Marche et d'Arrêt (GEMMA – ADEPA 1981)

Les modes de marche et d'arrêt, caractérisant l'état de fonctionnement d'un système automatisé donné, ont été formalisés en France par l'Agence pour le Développement de la Production Automatisée (ADEPA) en 1981 sous la forme d'un guide graphique, le GEMMA.

Le GEMMA, c'est tout à la fois :

- un outil graphique constituée d'une association de rectangles et de check-lists,
- un ensemble de rectangles représentant des modes distincts de fonctionnement,
- une grille-support structurée sur laquelle trouve leur position et occupe toujours la même place chacun de ces rectangles,
- un vocabulaire précis énonçant les conditions de marche et d'arrêt du système.

Le schéma qui suit en donne une représentation complète sur laquelle apparaissent des pointillés de couleurs violette et rouge dont on va voir qu'ils délimitent plusieurs zones distinctes. Commençons par bien définir ces différentes zones.



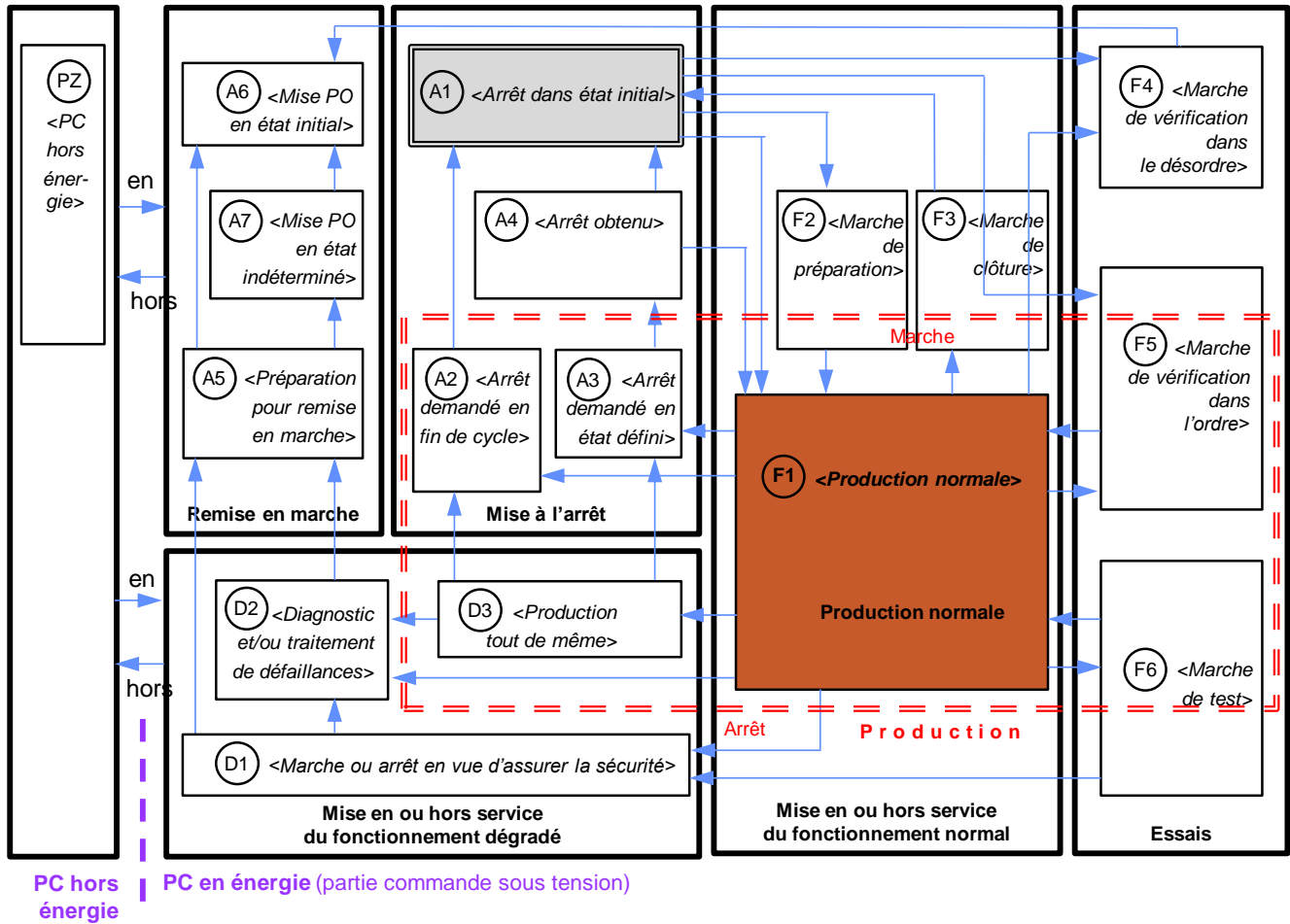
Le pointillé de couleur violette divise le GEMMA en deux zones, à gauche et à droite de ce pointillé.

- A gauche, un seul rectangle représentant le **mode PC hors énergie** selon lequel la partie commande n'est pas sous tension<sup>2</sup> et demeure par conséquent totalement inactive ; seules des actions dites actions réflexes ou venues du monde extérieur au système peuvent y avoir lieu.
- A droite, tous les autres rectangles représentant les **modes PC en énergie** selon lesquels la partie commande est sous tension et parfaitement active ; les actions prévues par le fonctionnement du système peuvent avoir lieu pour autant que le permettent les différents modes de marche ou d'arrêt, de mise en ou hors service du fonctionnement normal et de mise en sécurité ou en maintien de la disponibilité du système, etc.

Le double pointille rouge divise la zone PC en énergie du GEMMA là encore en deux zones, à l'intérieur et à l'extérieur du rectangle dit de production.

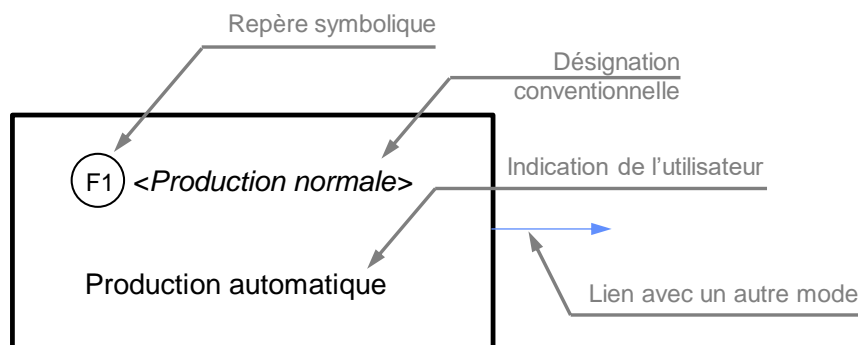
- A l'intérieur du rectangle en double pointillé rouge se trouvent la **zone en production** ; dans cette zone, le système produit la valeur ajoutée pour laquelle il a été conçu et réalisé.
- A l'extérieur du rectangle en double pointillé rouge se trouve la **zone hors production** ; dans cette zone, le système ne produit pas la valeur ajoutée pour laquelle il a été conçu parce que lui a été donné un ordre d'arrêt en fin de cycle, dans un état défini ou pour cause de défaillance.

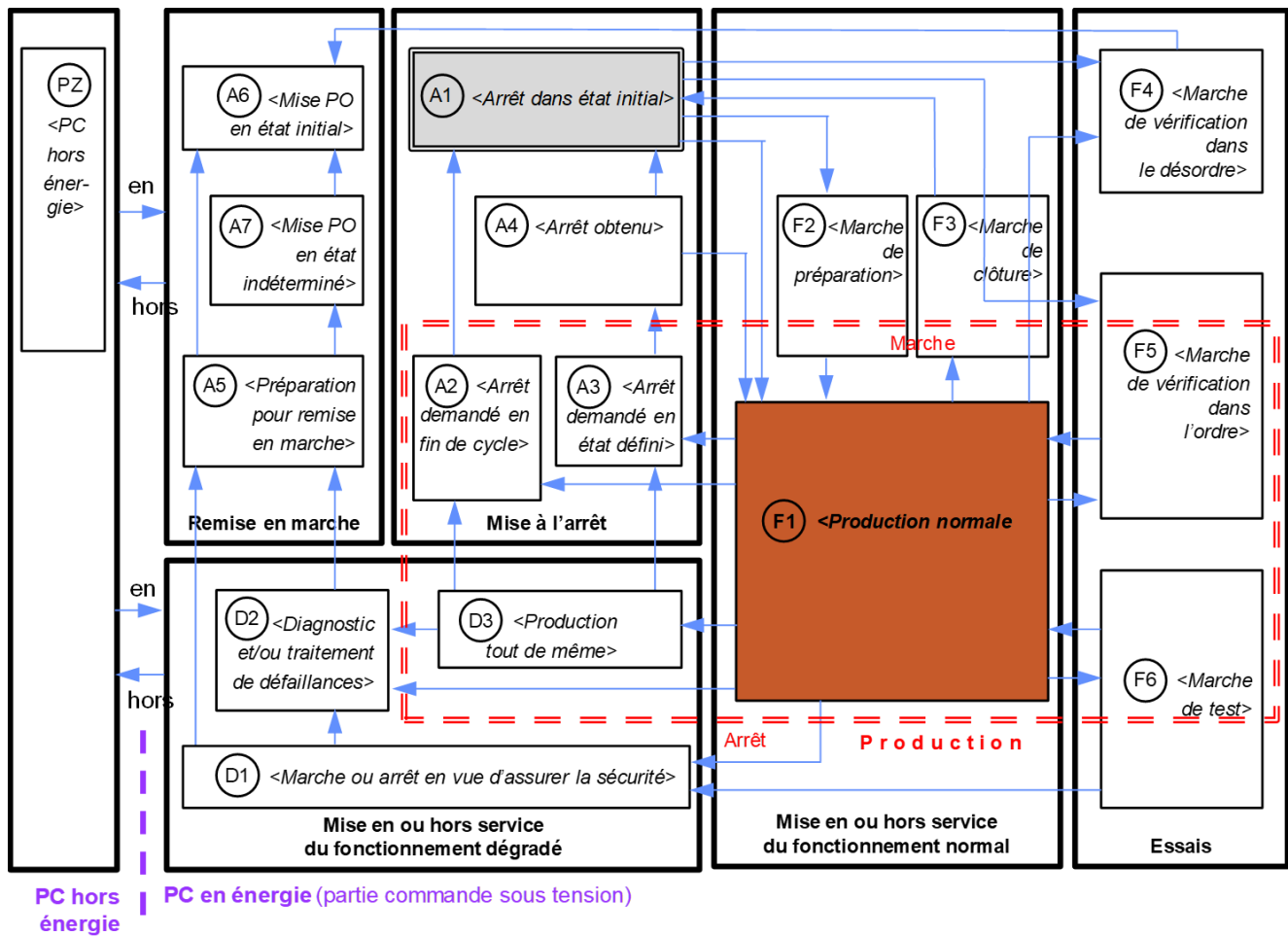
<sup>2</sup> Dans le cas d'une commande électrique, électronique ou informatique



Encadré dans un sous rectangle, chaque mode tel que A6, D2 ou F1 est caractérisé par quatre éléments :

- une **désignation dans un langage conventionnel** : (<Production normale>, par exemple),
- un **repère symbolique** : [F1], par exemple,
- des **liens** possibles avec d'autres mode,
- et une **indication qu'ajoute l'utilisateur** en langage naturel : (<Production automatique>, par exemple) ; dans cette indication, l'utilisateur précise le rôle du mode dans le cas particulier du système dont il établit le GEMMA.





Repérés chacun sur le GEMMA par un sous-rectangle, les modes de marche et d'arrêt sont de trois types.

- Désignés par les repères AX<sup>3</sup>, se trouvent les **modes de mise à l'arrêt et de remise en marche** correspondant à différentes phases d'exploitation de la machine :
  - A1 : < arrêt dans un état initial > ,
  - A2 : < arrêt demandé en fin de cycle > ,
  - A3 : < arrêt demandé en état défini > ,
  - A4 : < arrêt obtenu > ;
  
- Désignés par les repères DX<sup>4</sup>, se trouvent les **modes de mise en sécurité ou en maintien de la disponibilité** correspondant à différents états du système automatisé en cas de défaillance :
  - D1 < marche ou arrêt en vue d'assurer la sécurité > ,
  - D2 : < Diagnostic et/ou traitement de défaillances > ,
  - D3 : < production tout de même ><sup>5</sup>.

<sup>3</sup> A comme Arrêt

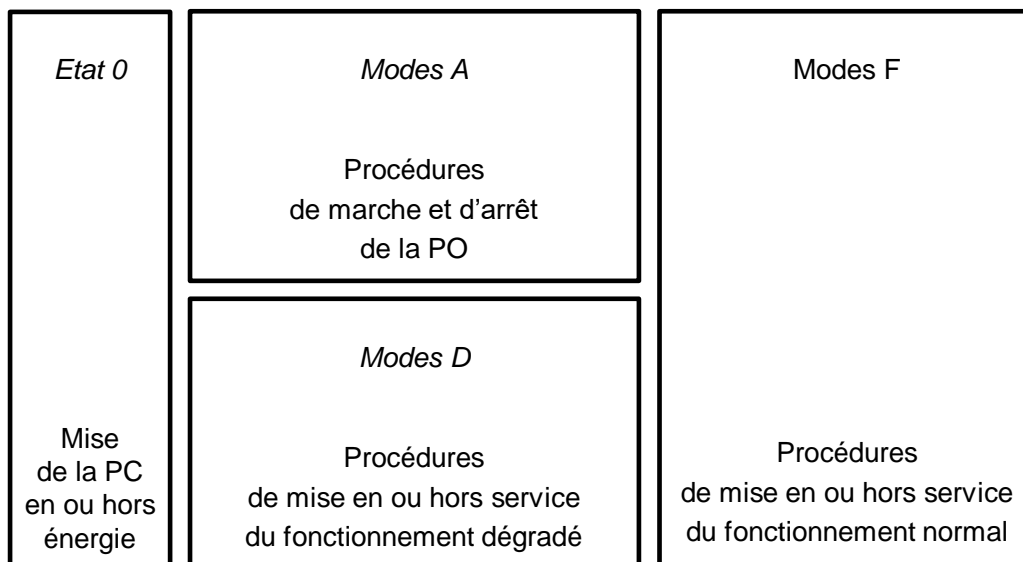
<sup>4</sup> D comme Défaillance

<sup>5</sup> C'est-à-dire : production en mode dégradé.

- Désignés par les repères FX<sup>6</sup>, se trouvent les **modes de mise en ou hors service du fonctionnement normal** correspondant à trois conditions précises de production du système automatisé :
  - F1 : < *production normale* > ,
  - F2 : < *marche de préparation* > ,
  - F3 : < *marche de clôture* > .

En résumé, le GEMMA distingue :

- les procédures de **marche et arrêt** de la partie opérative ; ce sont les **modes A** (A comme arrêt) ;
- les procédures de **mise en ou hors service du fonctionnement dégradé** de la partie opérative en cas de défaillance ; ce sont les **modes D** (D comme défaillance) ;
- les procédures de **mise en ou hors service du fonctionnement normal** de la partie opérative sans défaillance aucune ; ce sont les **modes F** (F comme fonctionnement).



Le GEMMA est donc un outil graphique destiné :

- d'abord à **sélectionner les modes de marche et d'arrêt** dont a besoin le système pour fonctionner en sécurité et produire la valeur ajoutée pour laquelle il a été conçu et réalisé ;
- ensuite à **déterminer les conditions de passage d'un mode à l'autre**.

Sur le GEMMA figurent tous les modes possibles dont chacun d'eux peut être nécessaire au bon fonctionnement d'un système automatisé quel qu'il soit.

Il y a deux modes particuliers qu'on trouvera nécessairement sur les GEMMA de tous les systèmes automatisés sans exception, ce sont :

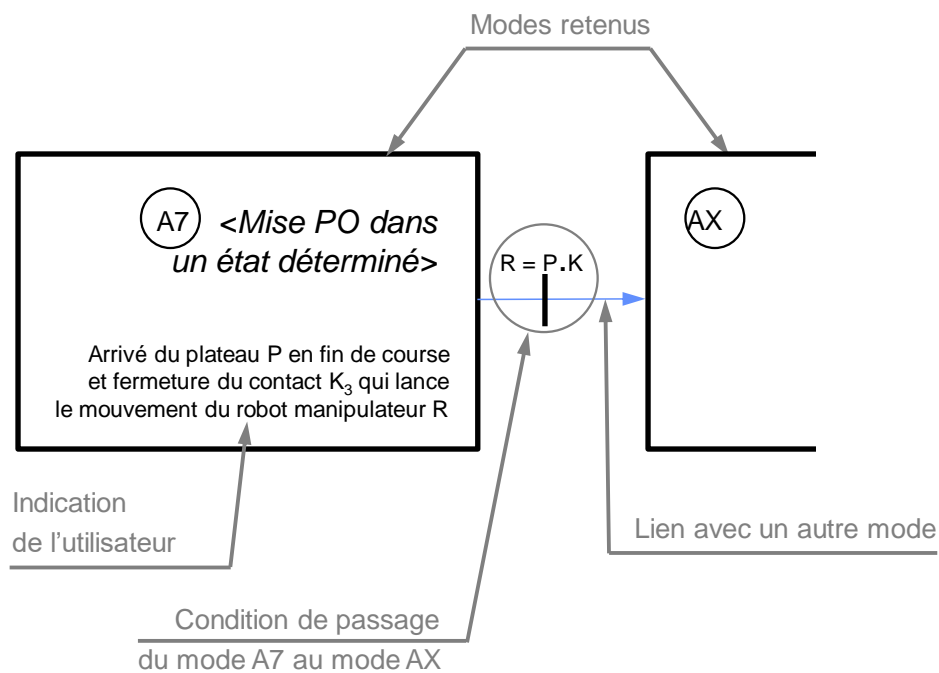
- le mode A1 de l'état initial, c'est-à-dire correspondant à l'état de repos du système,
- le mode F1 de la production normale, c'est-à-dire correspondant à son fonctionnement nominal.

<sup>6</sup> F comme Fonctionnement

### 4.2. Sélection des modes nécessaires de marche et d'arrêt<sup>7</sup>

La sélection des modes de marche et d'arrêt se fait une fois conçue la partie opérative et tracé le GRAFCET du système automatisé. Elle se fait par un balayage systématique de tous les modes proposés par le GEMMA et donc de tous les sous-rectangles qui le composent.

Lorsqu'un mode est retenu, l'utilisateur ajoute dans son rectangle l'indication, en langage naturel, de ce que signifie l'application de ce mode au système étudié.



Tous les modes non-retenus sont marqués d'une croix dans la zone du rectangle dédié à l'indication de l'utilisateur.



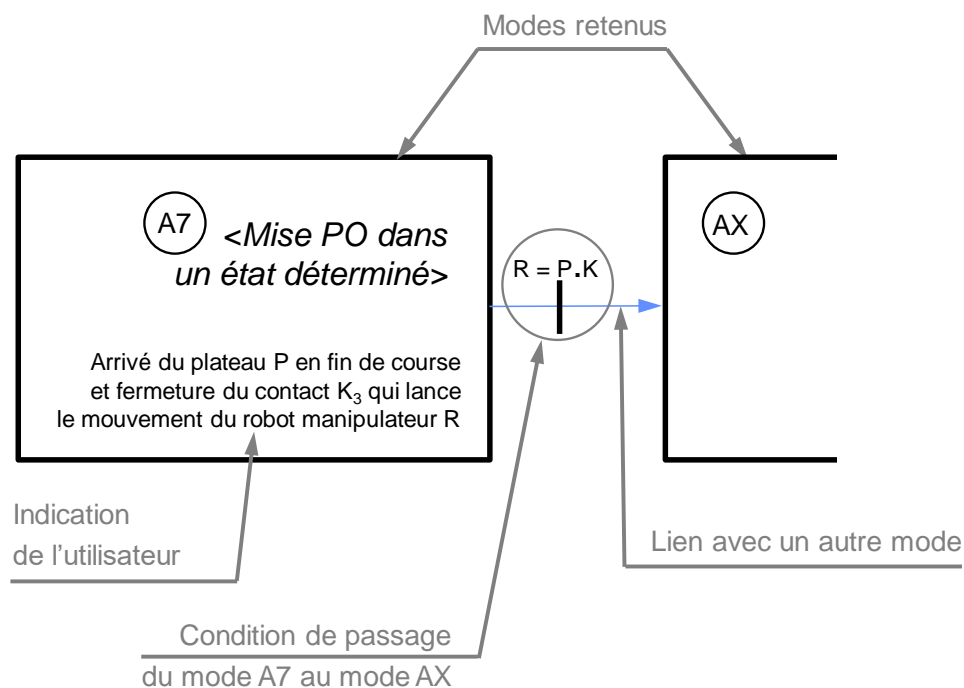
<sup>7</sup> D'après Technologue Pro

### 4.3. Détermination des conditions de passage d'un mode à l'autre<sup>8</sup>

Une fois sélectionnés les modes de marche et d'arrêt nécessaires au bon fonctionnement du système automatisé, il ne faut pas seulement déterminer les liens de passage d'un mode à l'autre, mais aussi à **quelle condition**, *quand condition il y a*, **doit se faire ce passage**.

Dans le schéma ci-dessous rappelé comme exemple, figure la condition de passage entre le mode A7 et un mode AX. L'existence de cette condition :

- est symbolisé par un trait vertical,
- est explicitée par une proposition logique, par exemple :  $R = P.K$ .



Pour déterminer, les conditions de passage d'un mode à l'autre, il est commode de commencer par les deux modes toujours présents sur le GEMMA de tous les systèmes automatisés quels qu'ils soient : le mode A1 d'Arrêt dans état initial, et le mode F1 de Production normale du système.

Si on part du mode A1, Arrêt dans état initial pour aller au mode F1, Production normale, il s'agira d'identifier :

- d'abord la suite des modes qui y conduit,
- puis les conditions de passage d'un mode au suivant ;

ces conditions pouvant résulter :

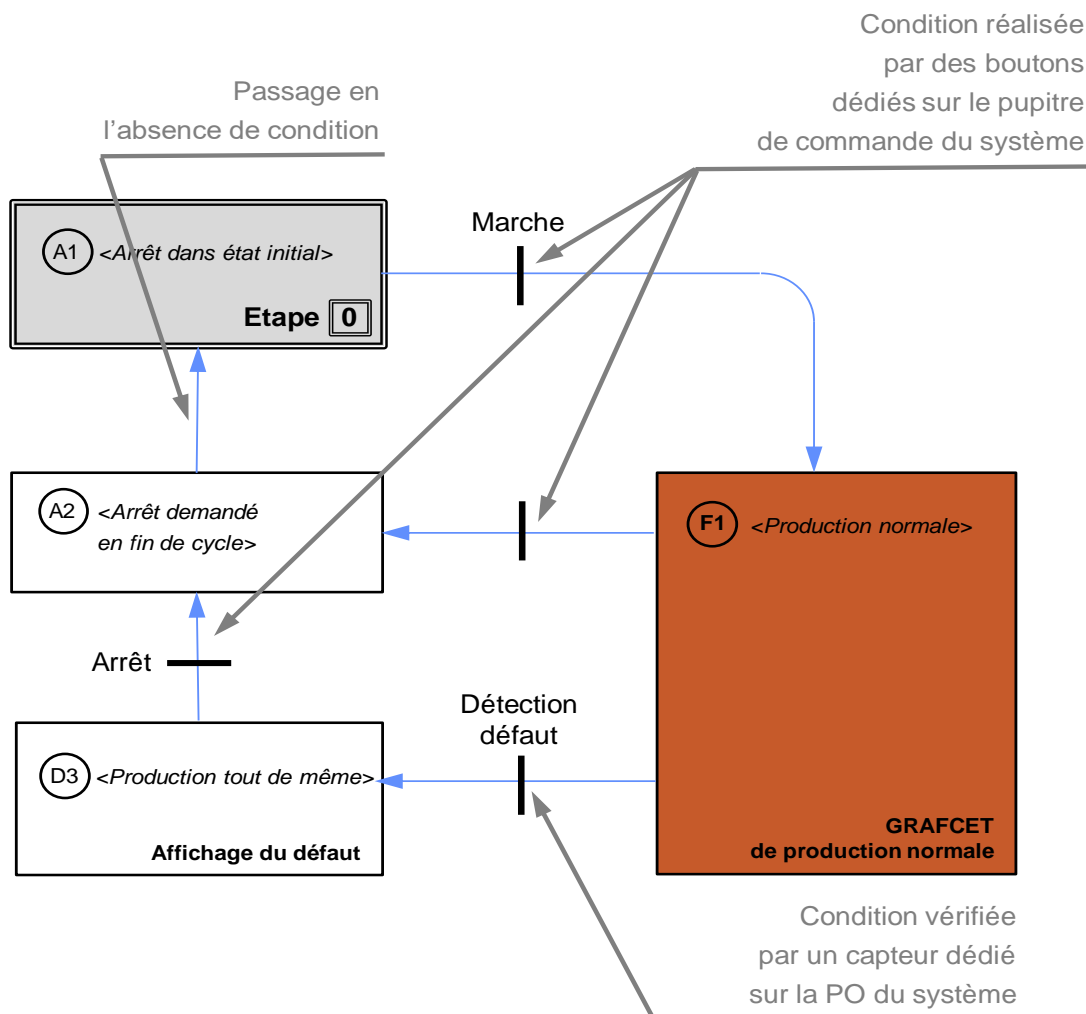
- aussi bien d'informations recueillies sur le système,
- que l'intervention extérieure d'un opérateur ou d'un dépanneur.

<sup>8</sup> D'après Technologue Pro

Si on part du mode F1 *Production normale*, il s'agira, là encore, d'identifier la suite des modes permettant, après arrêt de sécurité et préparation de la remise en route, le passage en mode manuel pour restaurer les conditions initiales et retourner à la *Production normale*.

On commence par conséquent par rechercher à quels autres modes sont reliés ces deux premiers modes A1 et F1. Puis, de proche en proche, on identifie tous les liens qui réunissent deux à deux les différents modes préalablement sélectionnés.

Ensuite, sur chaque lien identifié, on vérifie si le passage du mode au suivant se fait sous condition ou pas ; et si le passage se fait sous condition, on indique sous la forme qui convient le contenu de cette condition. Le schéma qui suit montre un exemple complet de GEMMA avec ses modes retenus de marche et d'arrêt, ses liens identifiés entre les différents modes et les conditions, quand il y en a, de passage d'un mode au suivant.



De l'analyse de cet exemple, on déduit deux conséquences importantes.

La première est que, sur une boucle donnée, on passe d'un mode à l'autre pour finalement retrouver :

- soit **un état de référence**, par exemple le mode A1 : < *Arrêt dans état initial* > ,
- soit **un état normal**, par exemple le mode F1 : < *Production normale* > ;

ces deux états étant ceux de départ sur la boucle systématiquement présent sur tout GEMMA.

La seconde est qu'au fond, établir le GEMMA d'un système automatisé se ramène toujours à **identifier des boucles fonctionnelles**. C'est ainsi que le schéma ci-dessus a fait apparaître les boucles suivantes :

- la boucle  $A1 \rightarrow F1 \rightarrow A2 \rightarrow A1$ ,
- la boucle  $A1 \rightarrow F1 \rightarrow D3 \rightarrow A2 \rightarrow A1$ .

Une boucle se forme dès lors que pour passer d'un mode à un autre mode, le respect des conditions de passage oblige à passer par un mode intermédiaire qui permettra d'atteindre l'état suivant et, le moment venu, l'état final sans risque de dysfonctionnement pour le système.

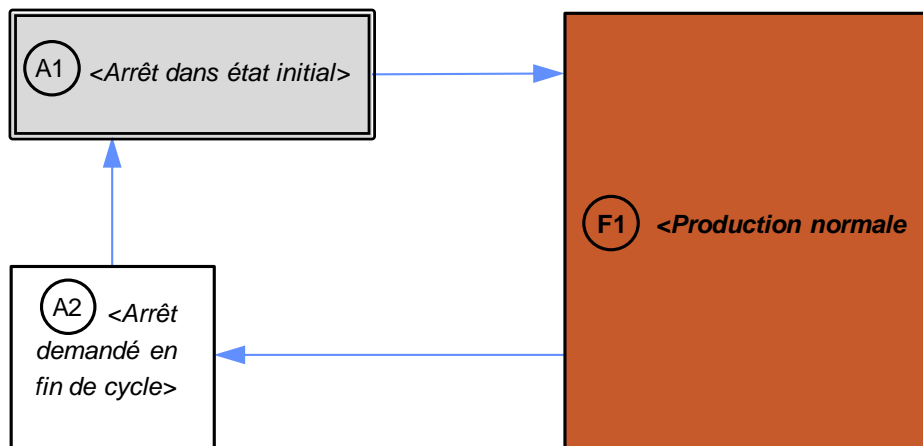
#### 4.4. Les trois boucles fonctionnelles du GEMMA<sup>9</sup>

De ce qui précède, on en déduit d'abord que l'élaboration d'un GEMMA s'organise toujours selon un même principe de recherche de boucles qui ne peuvent avoir que deux points de départ possibles : A1 et F1.

Ensuite, l'expérience montre que les boucles auxquelles on aboutit sont toujours de trois types :

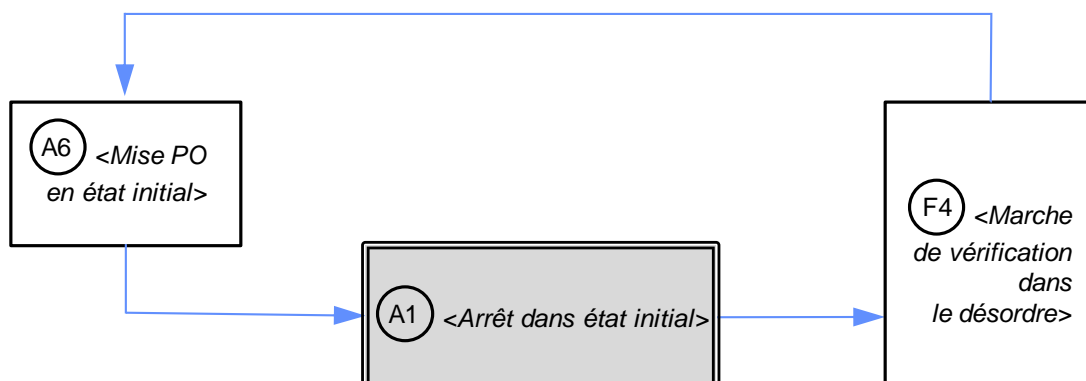
- la **boucle de marche normale** :  $A1 \rightarrow F1 \rightarrow A2 \rightarrow A1$ ,
- la **boucle de marche de réglage** :  $A1 \rightarrow F4 \rightarrow A6 \rightarrow A1$ ,
- la **boucle d'arrêt de sécurité** :  $F1 \rightarrow D1 \rightarrow A5 \rightarrow A6 \rightarrow A1$ .

La boucle  $A1 \rightarrow F1 \rightarrow A2 \rightarrow A1$  est une **boucle de marche normale**.



Elle décrit par conséquent le fonctionnement normal du système automatisé : à la fin de son cycle, le système se remet en position initiale et se tient prêt pour le cycle suivant.

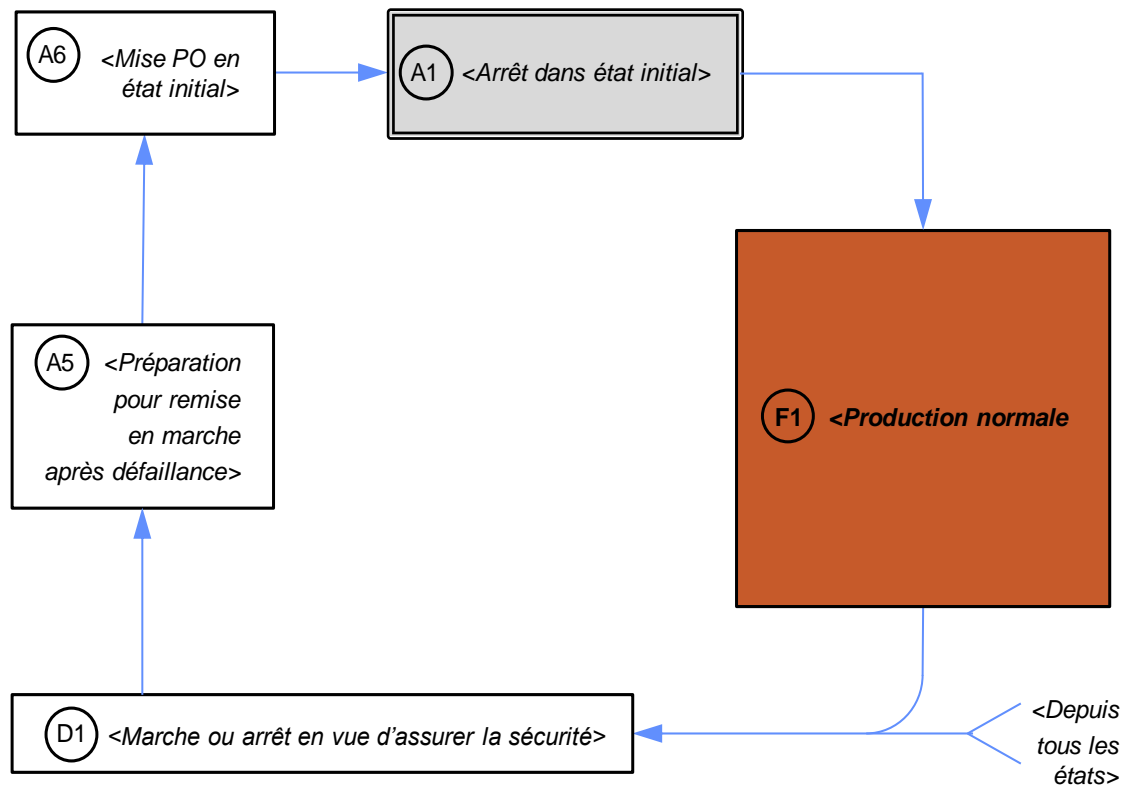
La boucle  $A1 \rightarrow F4 \rightarrow A6 \rightarrow A1$  est une **boucle de marche de réglage**.



Le mode F4 permet à l'utilisateur de tester tous les composants de la PO (capteurs, pré-actionneurs, actionneurs, etc.). Une fois les vérifications effectuées, le système passe du mode F4 au mode A6 qui lui permet de retrouver l'état initial A1.

<sup>9</sup> D'après Technologue Pro

La boucle F1 → D1 → A5 → A6 → A1 → F1 est une **boucle d'arrêt de sécurité**.



Elle permet de gérer tous les modes successifs d'un système automatisé depuis l'arrêt d'urgence lors d'une production normale jusqu'à la reprise de cette production normale. Il convient de noter que sur la flèche de liaison du mode F1 au mode D1 vient se greffer l'extrémité d'une flèche comportant une désignation conventionnelle *< Depuis tous les états >*. Cette désignation indique que le mode D1 *< Marche ou arrêt en vue d'assurer la sécurité >* peut être atteint depuis tous les autres modes retenus dans le GEMMA étudié : si les conditions requises pour passer en D1 sont réunies, alors le système passera dans ce mode D1.

### Deux remarques pour finir sur le GEMMA

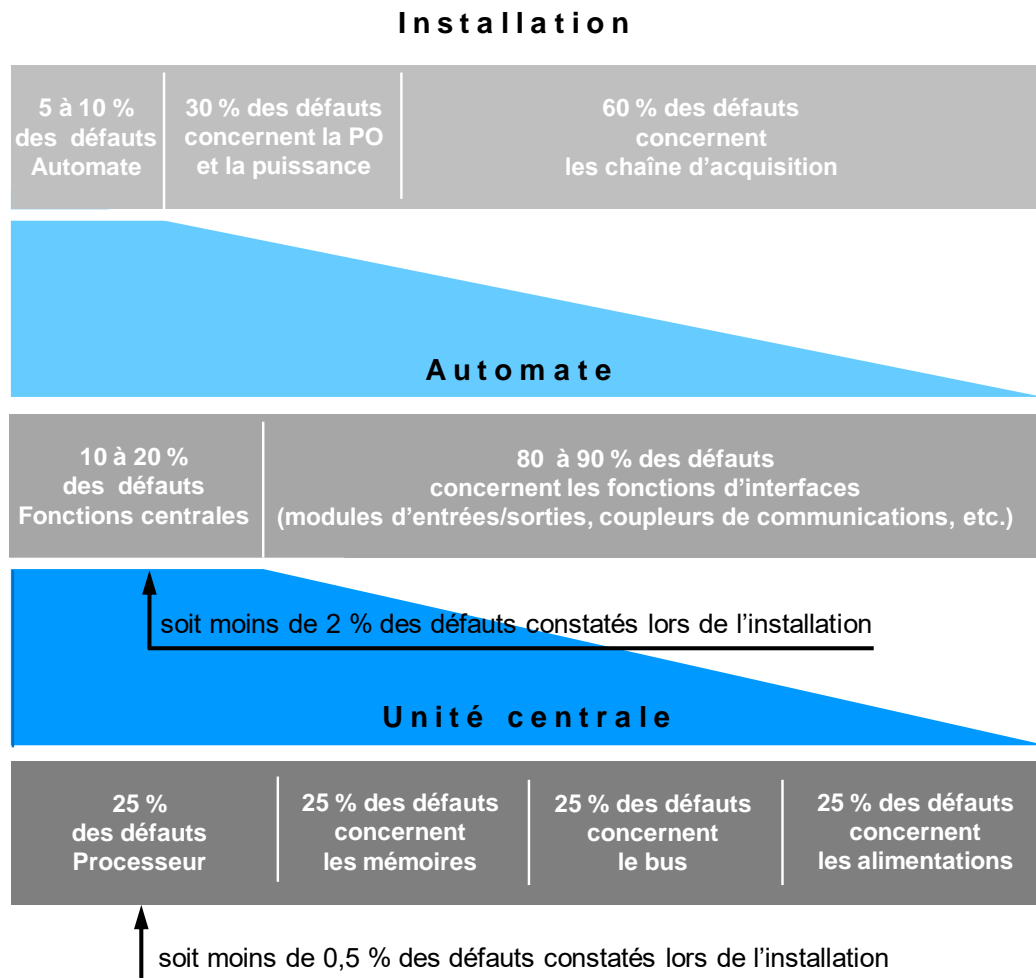
Première remarque importante : l'élaboration du GEMMA peut conduire à l'ajout de capteurs supplémentaires dans la partie opérative d'un système automatisé. Il s'agit là de capteurs supplémentaires non prévus par le fonctionnement du système, mais requis par la sécurité de son fonctionnement.

Seconde remarque de simple bon sens : une fois élaboré le GEMMA d'un système automatisé, il convient sans doute de ne plus le désigner par le sigle GEMMA qui, rappelons-le signifie Guide d'Etudes des Modes de Marche et d'Arrêt, mais par celui de :

**GMMA pour Graphe des Modes de Marche et d'Arrêt.**

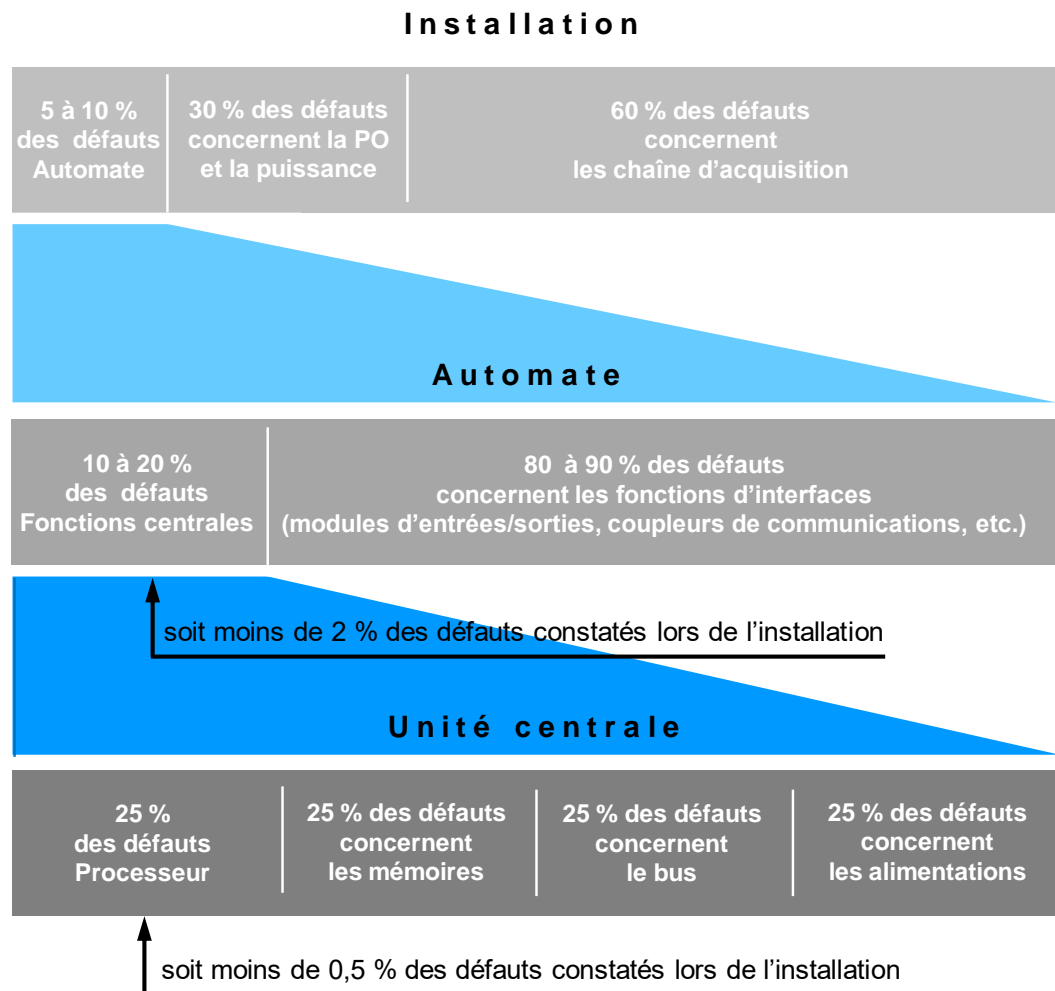
#### 4.5. La localisation statistique des défaillances matérielles

Un système automatisé met en jeu de très nombreux éléments qui, interagissant les uns avec les autres, sont autant de causes possibles de défaillances (main-d'œuvre, matières premières, produits semi-ouvrés, outils, technologies d'automatisation et d'informatisation, méthodes mises en œuvre, milieu environnant, énergies utilisées, etc.).



Des méthodes d'analyse de la sûreté permettent d'identifier les risques, d'évaluer leur gravité et leur probabilité d'apparition, puis de définir des procédures et solutions adaptées aux différentes situations possibles. Cependant il est important de rappeler que le risque nul n'existe pas et que le concepteur doit rechercher en permanence le compromis entre la sûreté et le coût.

S'agissant en particulier d'une installation commandée par un automate programmable, et bien que les situations rencontrées soient extrêmement variées, des éléments statistiques permettent de dégager une répartition générale des défauts.



Le graphique ci-dessus montre, d'un point de vue structurel, une localisation des défauts d'origine matérielle. On notera que ceux-ci se localisent principalement au niveau des chaînes d'acquisition (60%) et d'action (30%), l'automate représentant moins de 10 % des défauts et la part imputable à l'unité centrale restant très faible (de l'ordre de 2 %).

#### 4.6. Les notions de fonctionnement normal et de fonctionnement en mode dégradé

Généralement, la sécurité oriente, en cas de défaillance, le comportement d'un système automatisé ou de sa chaîne d'action vers l'arrêt : c'est la sécurité positive.

Cependant il existe des cas où elle peut être orientée vers la marche, c'est-à-dire vers le maintien du fonctionnement dans des conditions qui, ne pouvant plus être normales, sont dégradées mais encore tolérées par le système.

Le principe de base est donc d'orienter, en cas de défaillance, le comportement de la chaîne d'action **d'un mode de fonctionnement normal vers un mode de fonctionnement dégradé** de façon à ce qu'elle continue de fonctionner sans porter atteinte ni aux technologies, ni à l'organisation, ni bien sûr aux hommes présents sur le site.

Ces procédures de fonctionnement en mode dégradé sont prévues à l'avance ; elles concernent autant la partie commande que la partie opérative des systèmes concernés.

D'autres systèmes sont dits à *tolérance de fautes*. Ils continuent, en cas de défaillance, de fonctionner en mode normal parce qu'a(ont) été *a priori* doublé(s) l'(les) organe(s) dont on connaît, à la fois, le niveau de fiabilité et le caractère rédhibitoire de sa(leurs) défaillance(s).

## Conclusion

Dédié à la logique et aux modes de représentation du fonctionnement des systèmes automatisés, ce chapitre 8 est le dernier de l'ouvrage à s'intéresser aux techniques d'automatisation, le dernier par conséquent à traiter de questions techniques.

Ce chapitre a d'abord apporté des concepts fondamentaux sur l'information et son support. Il a ensuite caractérisé la nature du dialogue entre des systèmes interconnectés et du dialogue entre l'homme et ces systèmes. Il a enfin décliné pas à pas la logique du fonctionnement des systèmes et fournit les règles de son écriture. Enfin, au travers du GEMMA, il a proposé une véritable méthodologie de détermination des modes de Marche et d'Arrêt des systèmes, non seulement pour passer du fonctionnement nominal à un fonctionnement dégradé, mais aussi pour revenir au fonctionnement nominal.

A ce stade, il ne subsiste plus qu'une double question absolument fondamentale, celle du pourquoi et celle du comment de la productique. En d'autres termes, après la question des technologies de réalisation, après celles des techniques de conception, subsiste cette double question : qu'est-ce qui peut justifier de développer un système productique et comment s'y prendre pour conduire un projet en productique ?

La question du pourquoi a été abordée à la fin du chapitre 4 sur le système productique sous le titre : *La productique : pourquoi faire ?* Mais elle n'y a été qu'abordée dans la mesure où n'a pas été définie, à ce stade, ce que peut et doit être une stratégie de développement d'un système productique. Reste donc la question du comment s'y prendre qui commence par l'élaboration d'une stratégie de développement de la productique et qui se termine par la conduite de projets en productique. C'est ce à quoi sont consacrés les chapitres 9 et 10, les deux derniers de cet ouvrage sur la production industrielle automatisée et informatisée.

---