

TECHNOSUP

Les FILIÈRES TECHNOLOGIQUES des ENSEIGNEMENTS SUPÉRIEURS

AUTOMATISMES

*nouvelle
édition*

Développement des grafjets

**Des machines simples aux cellules flexibles
Du cahier des charges à la programmation**

Bernard REEB



Dans la même collection

- **Maintenance industrielle** (B) 288 p. Jean-Marie AUBERVILLE
- **Analyse et maintenance des automatismes industriels** (B) 192 p. Alain REILLER
- **TGAO La technologie de groupe,
Pour une meilleure organisation du système de production** (C) A. NADIF
- **Méthodes, productique et qualité** (B) 224 p. Jean-Marie CHATELET
- **Maîtriser la conduite de projet** (A) 192 p. Patrick ALONSO
- **Management de projet technique** (B) 192 p. C.CAZAUBON, G.GRAMACIA,
G.MASSARD
- **Organisation et génie de production** (B) 224 p. Francis LAMBERSEND
- **Méthode d'aide à la décision.
Approche théorique et études de cas** (B) 192 p. Robert LABBE

ISBN 978-2-7298-6203-9

©Ellipses Édition Marketing S.A., 2011

32, rue Bague 75740 Paris cedex 15



Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5.2° et 3°a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective », et d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit constituerait une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

PREFACE

Les graphes fonctionnels étapes-transitions (grafcets) sont nés du besoin de décrire de façon claire et fiable les processus de plus en plus complexes des automatismes industriels. Ils ont donc d'abord été développés dans les grandes entreprises mettant en oeuvre des dispositifs très sophistiqués, et ils ont été normalisés en France en 1982, puis en Europe en 1990. Maintenant largement répandu, le Grafcet est un outil puissant offrant les meilleures garanties, pour la modélisation et la transmission d'informations entre les différents acteurs d'un projet. Il permet de répondre facilement et vite à des problèmes variés, ce qui le rend même indispensable jusque dans les petites installations des P.M.I.

Pour traiter du Grafcet les ouvrages existants s'attachent essentiellement à en décrire le modèle et à préciser les recommandations de la normalisation. Mais pour aider à leur mise en oeuvre effective il convenait d'en développer la méthode de construction dans une approche raisonnée. C'est le pari de Bernard REEB qui formalise dans cet ouvrage la manière de les concevoir.

Bernard REEB est à la fois un enseignant confirmé et un praticien performant. A l'IUT de Mulhouse il a développé, en formation initiale et en formation par apprentissage, les différents enseignements d'automatismes industriels (automates, commande numérique, fabrication assistée par ordinateur, centre d'usinage, manipulation de robots...) et il assure le suivi des projets dans le domaine des automatismes. Il est également responsable du module complet *Automatismes combinatoires et séquentiels* dans la *Formation trinationale en mécatronique* qui regroupe les universités de Mulhouse, de Lörrach en Allemagne et de Bâle-Muttenz en Suisse.

L'ouvrage de Bernard REEB développe une construction logique progressive :

- Une première partie, introductive, permet de découvrir le Grafcet et présente les connaissances de base, pour comprendre son fonctionnement.
- Les deux parties suivantes sont consacrées à sa construction. Leur fil conducteur est à l'image des trois volets que comporte la méthode : l'analyse des fonctions, la recherche des séquences et le paramétrage. La présentation est étayée par des études de cas choisies pour bien isoler successivement chacun de ces aspects.
- Enfin, dans la dernière partie des exercices de synthèse et la description d'une unité de production permettent de combiner toutes les composantes de la méthode.

Sur un sujet technique aride et à un bon niveau, Bernard REEB a composé un ouvrage pédagogique, documenté et solide, avec un soin particulier porté à la progression des difficultés. Il se lit bien et il reste facilement accessible à tout étudiant expérimenté ou débutant. C'est l'ouvrage qui répond effectivement à l'attente de ceux qui sont, ou qui seront, appelés à construire des grafcets.

Le succès rencontré par la première édition maintenant épuisée conduisait tout naturellement à proposer cette nouvelle édition entièrement revue et complétée.

Claude CHEZE
Professeur des Universités
ancien Directeur d'Ecole d'ingénieurs et d'I.U.T.

BIBLIOGRAPHIE

Normes CEI 848 - UTE C 03-190/191 - NF C 03-190 - NF EN 61131-3

Réglementation générale du code du travail - Institut National de Recherche et de Sécurité

Documents constructeurs Ordinal Technologies, Schneider Electric, Siemens...

Le Grafcet : de nouveaux concepts (Grepa, Ed. Cepadues)

Le Gemma (Adepa)

Du Grafcet aux réseaux de Petri (René David, Hassan Alla, Ed. Hermès)

Automatique Informatique Industrielle (Christian Merlaud, Jacques Perrin, Jean-Paul Trichard, Ed. Dunod)

Les ateliers flexibles de production (Roger Bonetto, Ed. Hermès)

Note de l'auteur

Ma gratitude va à M. Claude CHEZE, Directeur de la collection Technosup. J'ai apprécié la qualité de ses conseils et sa juste exigence lors de l'élaboration de la première édition de cet ouvrage.

Je lui renouvelle mes chaleureux remerciements à l'occasion de cette deuxième édition.

A Laurence et Alexandra

TABLE DES MATIERES

PARTIE A : ELEMENTS DE BASE

I. PRESENTATION DU CONTEXTE

- 1. *Nécessité d'un langage commun* 7
- 2. *Découverte du Grafcet* 8
- 3. *Indications pour la mise en oeuvre* 13

II. PRINCIPES GENERAUX

- 1. *Le Grafcet : outil de modélisation* 23
- 2. *Niveau de précision d'un grafcet* 31
- 3. *Grafcets au niveau automate* 35
- 4. *Les modes de marches et d'arrêts* 41

III. EXERCICES D'INITIATION

- 1. *Cycle en triangle* 45
- 2. *Positionnement d'un chariot* 50
- 3. *Poste de perçage* 57

PARTIE B : STRUCTURE DES GRAFCETS

IV. NECESSITE D'UNE DEMARCHE

- 1. *L'expression du besoin* 71
- 2. *Le cycle de vie d'un système* 71
- 3. *Construction de grafcets* 73

V. ANALYSE DES FONCTIONS

- Etude de cas n°1 : poste de triage* 82

VI. SEQUENCES OBLIGÉES

- Etude de cas n°2 : cellule double* 99

PARTIE C : RECHERCHE DE LA FLEXIBILITE

VII. IDEES DIRECTRICES

- 1. *Le contexte industriel* 113
- 2. *La communication industrielle* 116
- 3. *Grafcet et flexibilité* 127

VIII. MISE EN OEUVRE

- Etude de cas n°3 : transfert linéaire de cellule* 131

PARTIE D : SYNTHESE

IX. EXERCICES DE SYNTHESE

- 1. *Bacs de trempe* 145
- 2. *Préparation de médicaments* 156
- 3. *Cellule de poinçonnage-contrôle* 164

X. APPLICATION

- Laboratoire didactique* 173

SIGLES

ADEPA	Agence Nationale pour le Développement de la Productique Appliquée à l'industrie
AF CET groupe	Association Française pour la Cybernétique Economique et Technique (le de travail « Systèmes Logiques » a créé le GRAFCET)
AFNOR	Association Française de Normalisation
CEI	Commission Electrotechnique Internationale
CIM	Computer Integrated Manufacturing (usine intégrée par ordinateur)
GEMMA	Guide d'étude des Modes de Marches et d'Arrêts
GMMA	Grafcet de gestion des Modes de Marches et d'Arrêts (ou grafcet de conduite)
GRAFCET	Graphe Fonctionnel de Commande Etapes-Transitions
GREPA	Groupe Equipement de la Production Automatisée réuni à l'ADEPA (a permis le développement, la diffusion et la normalisation du Grafcet)
ISO	International Standard Organization
OSI	Open Systems Interconnection
SFC	Sequential Function Chart (diagramme fonctionnel en séquence)
UTE	Union Technique de l'Electricité

I . PRESENTATION DU CONTEXTE

1 - NECESSITE D'UN LANGAGE COMMUN

Cet ouvrage est essentiellement consacré à la conception des machines de production. La réalisation d'un équipement industriel fait intervenir plusieurs services d'une même entreprise (méthodes, atelier de production, maintenance, service commercial, ergonomie...) et souvent plusieurs entreprises (le client, l'intégrateur, les sous-traitants...) Or, chacun a son expérience, ses habitudes, son savoir-faire et surtout son propre jargon.

En voici une illustration très simple. Pour utiliser un vocabulaire courant, on peut dire que la figure A-1 représente un *objet*. Un concepteur y voit cependant un *cylindre* en référence à sa géométrie, ou un *axe*, un *manchon* ou encore un *arbre* en référence à sa fonction. Pour un fabricant, c'est une *pièce*. Dans le domaine de la gestion de production, on parle d'*article* (élément de base d'une nomenclature). C'est pourtant le même *objet* pour tous : des malentendus peuvent donc nuire au bon déroulement d'un projet.

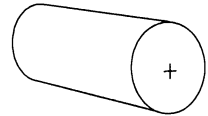


figure A-1

De plus, vers les années 70, la complexité croissante des automatismes industriels nécessite de nouveaux outils de modélisation. Sous la direction de Michel Blanchard, le groupe *Systèmes Logiques* de l'AFCET a répondu à ce besoin en créant en 1975 le Grafcet, dont le principe et le graphisme sont inspirés des Réseaux de Petri (1962, par Carl Adam Petri, mathématicien allemand). Dès alors, Jean-Paul Frachet a constitué le groupe de travail *Equipements de Production Automatisée* au sein de l'ADEPA afin de développer le Grafcet et de le promouvoir dans l'enseignement et l'industrie. Une démarche similaire a eu lieu simultanément en Allemagne pour aboutir à la norme DIN 40719.

A l'origine, le Grafcet est donc un outil de communication qui s'attache exclusivement à la description fonctionnelle des automatismes.

- Il a été normalisé en France en juin 1982 (NF C 03-190). En 1985, le GREPA a publié de nouveaux concepts.
- Une norme européenne a été consacrée au Grafcet (CEI 848 de 1988) sous la dénomination *diagramme fonctionnel pour systèmes de commande* ou *function chart for control systems*.
- La norme française a été revue suite à l'approbation internationale en donnant lieu à la norme UTE C 03-190 de novembre 1990, puis au complément UTE C 03-191 de juin 1993.
- La version française de septembre 1995 (NF C 03-190) est la plus récente.

Le concept Grafcet a été intégré dans la norme européenne sur les langages de programmation des automates sous la dénomination *SFC : diagramme fonctionnel en séquence* ou *sequential function chart* (CEI 61131-3 de juillet 1993, version française NF EN 61131-3 de novembre 1993).

Avant cette normalisation, les constructeurs avaient déjà proposé leurs propres solutions, très spécifiques, pour permettre la programmation sous représentation Grafcet. Maintenant, tous les automates se réfèrent à la norme CEI 61131-3. Cependant, les interfaces utilisateurs et les caractéristiques intrinsèques des automates sont toujours spécifiques à chaque constructeur, et les applications restent incompatibles d'une marque à l'autre.

Décliné en plusieurs niveaux, à la fois outil de modélisation et support pour la programmation, le Grafcet garantit aujourd'hui la transmission correcte des informations entre toutes les parties prenantes d'un projet d'automatisme, de la spécification à l'exploitation et à la maintenance, en passant par la réalisation.

2 - DECOUVERTE DU GRAFCET

Pour les lecteurs qui découvrent l'existence du Grafcet, voici une première approche de cet outil à travers la modélisation progressive d'un exemple simple. On suppose que *le système existe déjà* et qu'on peut donc en observer sa structure matérielle et aussi son fonctionnement.

Le système étudié est un cas industriel classique : un cycle de trempe (figure A-2).

2 - 1. Description de la machine

a) Description par une figure

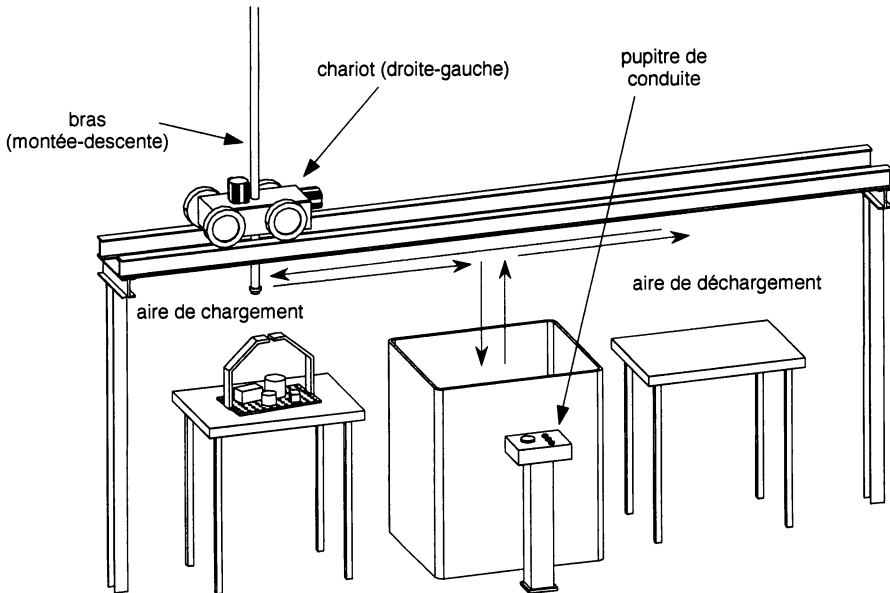


figure A-2

b) Description par un texte

Un opérateur installe un plateau qui contient des pièces à tremper sur le bras d'un chariot mobile. Il appuie ensuite sur le bouton *départ* installé sur un pupitre de conduite. Le chariot se déplace vers la droite jusqu'au bac de trempe. Le bras descend et plonge les pièces dans le liquide qui opère le traitement souhaité. Au bout de 8 minutes, le bras remonte. Puis le chariot se déplace de nouveau vers la droite et s'immobilise au-dessus de l'aire de déchargement. Un voyant clignote pour avertir l'opérateur que le cycle de trempe est terminé (l'opérateur peut avoir quitté ce poste pour effectuer un autre travail). Dès que l'opérateur a

déposé le plateau de pièces traitées, il appuie sur le même bouton *départ*, ce qui provoque cette fois le retour du chariot en position de chargement. Un nouveau cycle peut être demandé.

c) Description par une liste succincte

- L'opérateur installe le plateau et appuie sur le bouton *départ*.
- Le chariot se déplace jusqu'au bac.
- Le bras descend.
- Au bout de 8 minutes, le bras remonte.
- Le chariot se déplace jusqu'à l'aire de déchargement.
- Un voyant clignote, l'opérateur enlève le plateau et appuie sur le bouton *départ*.
- Le chariot revient en position de chargement.

Conclusion

Aucun des moyens de représentation ci-dessus ne suffit à lui seul à décrire le système.

- La figure ne décrit que très globalement l'installation. Elle peut être développée en un schéma technologique pour une connaissance plus fine des organes mis en oeuvre. Les dessins techniques de même que les schémas électro-pneumatiques et les nomenclatures apportent tous les détails quant à sa structure matérielle. Mais les mouvements ne peuvent être que devinés, et son fonctionnement n'est exprimé d'aucune manière.
- Le texte précise le fonctionnement du système, mais sa compréhension repose entièrement sur la figure. De plus, il doit être rédigé dans un langage qui ne laisse place à aucun doute. Cette rédaction est un exercice bien plus difficile que ce qu'on peut penser à priori.
- La liste est plus simple à rédiger *tant que l'évolution du système est linéaire*. Mais l'efficacité d'un tel document peut être compromise si plusieurs séquences sont exécutées simultanément, ou si des séquences sont exécutées dans certaines conditions seulement.

2 - 2. Construction intuitive à travers l'exemple

Malgré ses limites, la liste est un bon moyen de présenter simplement l'aspect dynamique du système. Mais pour que ce type de représentation puisse être généralisé, certaines règles doivent être observées. Ce chapitre permet de les découvrir progressivement.

a) Liste détaillée

La liste précédente, assez succincte, peut être complétée pour que tout ce qui se passe y figure d'une manière explicite.

(Le lecteur est vivement invité à rédiger sa propre liste et ensuite seulement à la comparer à celle proposée ci-dessous : on remarquera la difficulté à bien scinder, et quelquefois à combiner, les différentes « choses » qui se produisent, même dans un cycle aussi simple que celui-ci).

- L'opérateur installe un plateau.
- L'opérateur appuie sur le bouton *départ*.
- Le chariot se déplace vers la droite.
- Le chariot arrive au-dessus du bac.
- Le bras descend.
- Le bras arrive en position basse.
- Une temporisation de 8 minutes commence.
- Les 8 minutes sont écoulées.
- Le bras remonte.
- Le bras arrive en position haute.
- Le chariot se déplace vers la droite.
- Le chariot arrive au-dessus de l'aire de déchargement.
- Un voyant clignote, l'opérateur enlève le plateau.

- L'opérateur appui sur le bouton *départ*.
- Le chariot se déplace vers la gauche.
- Le chariot arrive au-dessus de l'aire de chargement.

b) Frontière entre opérateur et système

Les gestes de l'opérateur ne doivent pas tous être considérés de la même manière. Par exemple, la proposition *l'opérateur installe un plateau* n'a en soi aucune influence sur le fonctionnement du système alors que la proposition *l'opérateur appuie sur le bouton départ* constitue une cause de son évolution.

Puisque la liste représente le fonctionnement du système, on ne prend pas en compte les gestes de l'opérateur qui n'ont pas d'influence sur son évolution : dans cet exemple, il s'agit de la mise en place et de l'enlèvement du plateau.

c) Distinction entre action et réceptivité

L'étude porte sur la machine et non sur l'opérateur. Par conséquent, une action est l'effet d'un ordre émis par la machine et non par l'opérateur : le chariot se déplace *parce que* le système en a donné l'ordre (il s'agit bien d'une action produite par le système), alors que le fait d'appuyer sur le bouton départ n'est *pas* une action du point de vue du système (bien au contraire, il s'agit d'une information qu'il reçoit de son environnement).

Une telle information constitue une condition d'évolution. Elle permet au système de changer d'état (exemple : *si* le chariot arrive au-dessus du bac, *alors* il s'arrête et le bras descend). Du fait que les conditions d'évolution peuvent être très variées, le terme générique de *réceptivité* permet de les désigner quelle que soit leur nature.

d) Liste complète

On propose, provisoirement, de distinguer les actions et les réceptivités par une typographie différenciée (par exemple : actions en gras, réceptivités en italiques fins).

- *L'opérateur appuie sur le bouton départ.*
- **Le chariot se déplace vers la droite.**
- *Le chariot arrive au-dessus du bac.*
- **Le bras descend.**
- *Le bras arrive en position basse.*
- **Une temporisation de 8 minutes commence.**
- *Les 8 minutes sont écoulées.*
- **Le bras remonte.**
- *Le bras arrive en position haute.*
- **Le chariot se déplace vers la droite.**
- *Le chariot arrive au-dessus de l'aire de déchargement.*
- **Un voyant clignote.**
- *L'opérateur appuie sur le bouton départ.*
- **Le chariot se déplace vers la gauche.**
- *Le chariot arrive au-dessus de l'aire de chargement.*

Le point essentiel est de constater l'alternance entre les actions et les réceptivités.

e) Apport d'éléments graphiques et d'un vocabulaire approprié

- Les actions sont encadrées et les tirets qui les précèdent sont supprimés. Par contre, les tirets en face des réceptivités sont renforcés et appelés *transitions*. Le texte peut être condensé (figure A-3-a).
- A chaque action est associée une *étape*, représentée par un carré. Les étapes peuvent être numérotées afin de faciliter leur repérage et elles sont reliées entre-elles par des *liaisons orientées* (figure A-3-b).

L'alternance entre les actions et les réceptivités se traduit par l'alternance entre les étapes et les transitions. Cette caractéristique est en partie à l'origine du mot Grafcet (GRAPhe Fonctionnel de Commande Etapes-Transitions).

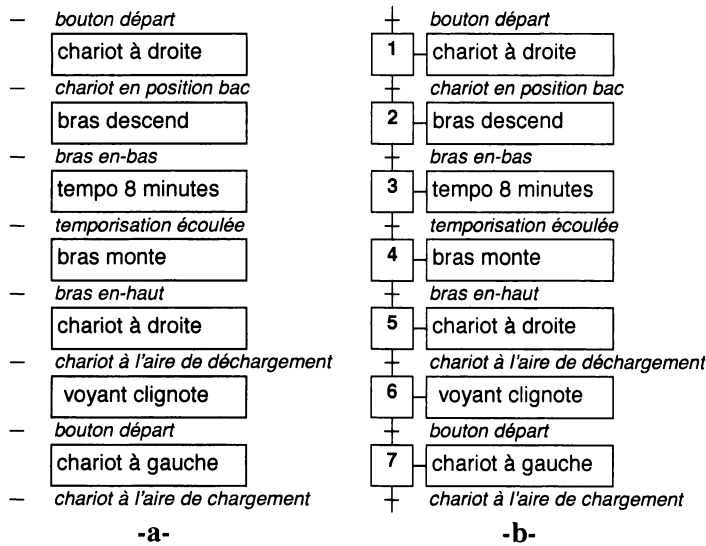


figure A-3

La figure A-3-b est cependant incomplète. En particulier, elle n'exprime pas le fait que les séquences peuvent se succéder les unes aux autres, et suggère *implicitement* le moment du début de la séquence (en-haut de la figure), ce qui est imprécis. Une analyse plus détaillée permettra de compléter ce diagramme afin d'aboutir à un grafcet complet.

f) Analyse de l'évolution du grafcet

L'action qui est produite à un instant donné est celle qui est associée à l'étape dite *active*. Le changement d'état du système est donc représenté par le changement de situation du grafcet, la situation d'un grafcet caractérisant l'ensemble des étapes qui sont actives à l'instant considéré. Ainsi, le passage d'une étape vers l'étape suivante se fait par franchissement de la transition qui les sépare.

En fait , *une transition est franchie lorsque l'étape qui la précède est active et que la réceptivité qui lui est associée est vraie.*

Par conséquent, le grafcet de la figure A-3-b doit comporter une étape *au-dessus* de la première transition pour que celle-ci puisse être franchie.

Par ailleurs, si aucune étape n'est active à la mise sous tension, aucune transition ne peut être franchie : aucune étape ne peut être activée et à fortiori, aucune action ne peut être produite. L'activation d'une étape supplémentaire (0 sur la figure A-4) peut être obtenue automatiquement dès la mise sous tension : il s'agit d'une étape initiale, représentée par un carré double.

La possibilité de répétition du cycle est symbolisée par la liaison entre les étapes 7 et 0. Les règles d'évolution du Grafcet sont détaillées au § II.1 - 1.

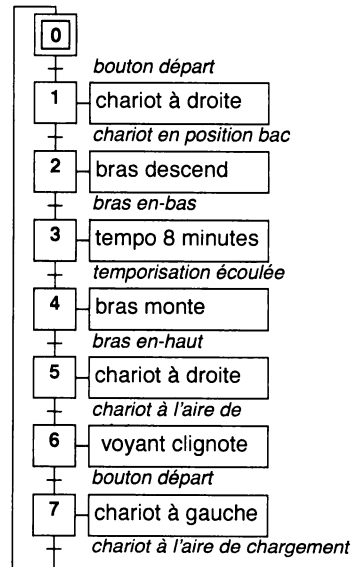


figure A-4

g) Notion d'automatismes combinatoires

Soit une action produite exclusivement par une combinaison de variables, par exemple une sonnerie déclenchée par l'un des deux boutons poussoirs situés aux deux entrées d'une maison. Le relâchement du bouton arrête la sonnerie.

L'équation logique de la sonnerie est :

$$\text{sonnerie} = \text{bouton 1 } \textit{ou} \text{ bouton 2}$$

On parle d'automatisme combinatoire car la sonnerie est produite par *combinaison* de variables.

h) Notion d'automatismes séquentiels

- Au niveau des actions

Dans l'exemple du poste de trempe, l'action *chariot à droite* ne dépend pas seulement des variables, mais dépend également de l'état courant du système : *chariot à droite* est produit par la variable *bouton départ* si le chariot se trouve en situation initiale (étape 0) ou bien par la variable *bras en-haut* si le bras vient de monter (étape 4). Cette proposition logique ne décrit toutefois que le début de l'action, ce qui peut s'écrire d'une manière symbolique :

$$\text{début chariot à droite} = [\text{bouton départ } \textit{et} \text{ étape 0}] \textit{ou} [\text{bras en-haut } \textit{et} \text{ étape 4}]$$

L'action se poursuit jusqu'à satisfaction de :

$$\text{fin de chariot à droite} = [\text{chariot en position bac } \textit{et} \text{ étape 1}] \textit{ou} [\text{chariot à l'aire de déchargement } \textit{et} \text{ étape 5}]$$

En fait, l'action dure tout le temps d'activation de l'étape 1 *ou bien* tout le temps d'activation de l'étape 5. L'équation de *chariot à droite* peut donc s'écrire plus simplement :

$$\text{chariot à droite} = [\text{étape 1}] \textit{ou} [\text{étape 5}]$$

- Au niveau des réceptivités

Toujours dans l'exemple du poste de trempe, on constate par ailleurs que le bouton *départ* n'a pas le même effet selon l'état du système.

- Si le chariot est en position de chargement, le fait d'appuyer sur le bouton *départ* provoque le déplacement du chariot vers la droite.
- S'il est en position de déchargement et qu'on appuie sur ce *même* bouton, le chariot se déplace vers la gauche.
- Dans toutes les autres situations, si le bras est au fond du bac par exemple, l'appui sur le bouton en question ne provoque *aucun* effet.

On parle d'automatismes séquentiels car les situations sont organisées en séquences : les actions sont le résultat de situations successives et le rôle des réceptivités dépend de la situation courante.

i) Ecriture des équations des sorties

On appelle *équations des sorties* les propositions logiques qui définissent la production des actions.

Dans l'exemple de la figure A-5, l'action A1 est produite si l'étape 2 *ou* si l'étape 3 est active. On écrit donc :

$$\text{A1} = [\text{étape 2}] \textit{ou} [\text{étape 3}]$$

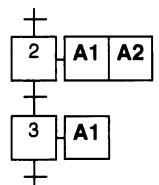


figure A-5

Il est incorrect par contre de combiner toutes les actions qui sont produites par une même étape. Exemple :

~~étape 2 = [action A1] et [action A2]~~

- En effet, les fonctions logiques permettent de combiner des variables d'entrée mais ne sont pas destinées à effectuer des affectations multiples.
- Par ailleurs, le signe = peut prêter à confusion car il n'indique pas explicitement le sens de lecture d'une équation logique, qui, comparativement à l'algèbre des nombres réels, est significatif. De ce fait, on préfère souvent le signe ← qui est sans ambiguïté :

A1 ← [étape 2] ou [étape 3]

Cette expression montre très clairement que l'activation des étapes 2 ou 3 *entraîne* la production de l'action A1. Par convention, l'écriture d'une telle équation se fait plus rarement de la gauche vers la droite.

3 - INDICATIONS POUR LA MISE EN OEUVRE

Ce paragraphe situe succinctement la place du Grafcet dans l'univers complexe que constitue un système automatisé de production afin que le lecteur puisse ensuite aborder les manuels de programmation des automates.

3 - 1. Architecture générale d'un système automatisé

Le cycle de trempe décrit plus haut ne peut se dérouler d'une manière automatique que si la machine :

- comporte un certain nombre de constituants,
- gère elle-même les interactions des constituants,
- dialogue avec son environnement.

On appelle cet ensemble *système de production automatisée* (figure A-6).

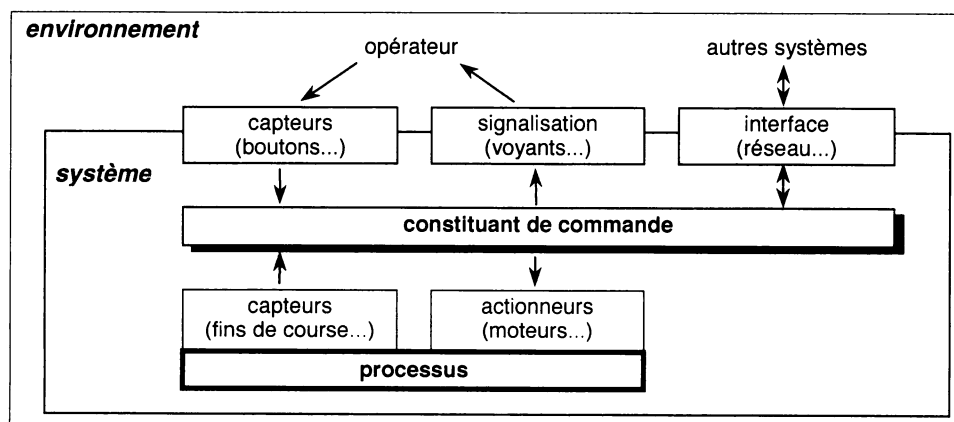


figure A-6

3 - 2. Technologie du constituant de commande

a) Solutions câblées

Les fonctions de commande peuvent être réalisées par des composants logiques élémentaires de type électrique, pneumatique, électronique ou plus rarement fluide. Ces

composants sont à relier entre-eux par des liaisons physiques (fils, tubes, circuit imprimé) selon le problème à résoudre. Chaque problème nécessite donc un schéma spécifique. Celui-ci peut être élaboré par des méthodes d'analyse diverses : tableaux de Karnaugh, méthode en cascade, etc. La reconfiguration du coffret de commande est une opération longue et délicate si des modifications ou des ajouts s'avèrent nécessaires. Composants évolués, les séquenceurs permettent de simplifier l'étude et le câblage des systèmes séquentiels en permettant en outre une meilleure évolutivité des machines. Ils sont disponibles dans les différentes technologies. Les solutions câblées ne représentent plus qu'une part infime des réalisations mais restent performantes dans des domaines particuliers : les temps de réponse sont très courts, et en ambiance explosible la sécurité est optimale pour la technologie pneumatique.

b) Solutions programmées

La souplesse de la programmation permet d'adapter très rapidement le comportement des machines aux contraintes changeantes de la production, même en phase d'exploitation. La profusion des fonctions disponibles grâce à l'évolution de la micro-informatique permet en outre d'améliorer considérablement la qualité générale des machines : dialogues évolués avec l'opérateur, horodatage des événements, dialogues avec les parties commandes d'autres machines et prise en charge du positionnement précis d'un axe de déplacement ne sont que quelques exemples. Il existe 3 familles de solutions programmées, adaptées chacune à un type particulier d'application : le micro-contrôleur, l'automate programmable et le micro-ordinateur.

- Le micro-contrôleur

D'un prix d'achat très attractif, il nécessite cependant un temps (donc un coût) de développement assez conséquent. Son intérêt réside de ce fait dans les applications de série. Sa programmation est proche de celle du micro-processeur.

- L'automate programmable

D'un prix plus élevé, l'automate programmable est cependant d'un rapport performance-prix de plus en plus intéressant. Contrairement au micro-contrôleur, il ne requiert pas de développement matériel supplémentaire. Sa programmation ne nécessite qu'une formation d'environ une semaine pour un technicien. Par ailleurs, il est robuste et est conçu pour offrir une vaste gamme d'interfaces pour communiquer avec son environnement : capteurs et actionneurs, dialogues opérateurs... Pour ces raisons, il est bien adapté au milieu industriel mais est également très utilisé dans la gestion technique des bâtiments (ascenseurs, climatisation, sécurités, etc.) ou des voies de communication (la signalisation active des autoroutes par exemple est souvent pilotée par des automates industriels).

- Le micro-ordinateur

Il offre l'avantage de pouvoir traiter et de stocker de grandes quantités d'informations avec des outils standards (plate-forme Windows par exemple). Mais sa constitution ne lui permet pas de survivre dans l'ambiance agressive d'un atelier (vibrations, pollution électromagnétique, etc.). Des modèles industriels sont disponibles, mais leur prix reste encore élevé.

- Solutions hybrides

L'interpénétration des domaines de l'automatique et de l'informatique conduit aux solutions qui combinent automate et micro-ordinateur :

- soit le micro-ordinateur est complété par des cartes d'entrées-sorties industrielles et d'une carte pour automatismes séquentiels (un automate en fait),

- soit c'est l'automate qui est complété par une unité centrale de type PC ou tout autre élément issu du monde informatique : carte graphique, carte réseau Ethernet, etc.
- certains constructeurs proposent depuis peu des solutions qui intègrent des fonctions d'automatismes et des fonctions d'informatique sans plus faire référence ni à l'automate, ni au micro-ordinateur.

3 - 3. Structure matérielle d'un automate programmable

Notre domaine d'étude étant limité aux systèmes industriels, c'est l'automate programmable qui retiendra plus particulièrement notre attention. *Voir illustrations en fin de chapitre.*

a) Les entrées-sorties

Revenons au poste de trempe décrit au chapitre précédent. L'expression des actions et des réceptivités du grafset se limitait volontairement à des observations très superficielles. Afin de pouvoir piloter le système d'une manière automatique, il est nécessaire d'une part de *recueillir les informations* qui constituent les réceptivités et d'autre part, de *transmettre des ordres* qui permettront aux moteurs, voyants... de fonctionner.

Exemples : la proposition *chariot présent à l'aire de déchargement* est une information issue d'un capteur de position installé sur le rail ; *bras descend* doit d'abord être un ordre dont l'effet est la rotation du moteur adéquat dans le sens adéquat.

L'ensemble des informations constitue le vecteur des *entrées*. L'ensemble des ordres est le vecteur des *sorties*.

Les informations transitent par le coupleur d'entrée dont le rôle est d'assurer l'adaptation des signaux, le filtrage des parasites ainsi que l'isolement de l'automate du milieu extérieur grâce à un circuit optoélectronique.

Entre deux mises à jour (voir § 3 - 4), les ordres fournis par le programme sont mémorisés par le coupleur de sortie qui comporte également un circuit d'isolement optoélectronique ainsi qu'un circuit d'amplification des signaux ou des relais électromagnétiques.

b) Les variables internes

Le cycle de trempe a, de plus, fait intervenir une temporisation, qui est une variable interne. Les variables internes sont à la fois entrées *et* sorties selon le besoin mais sont stockées dans l'unité centrale. Elles ne transitent donc jamais par les coupleurs d'entrées-sorties. Lors du démarrage de la temporisation, la variable interne associée peut être *considérée* comme une sortie. Lors du test de son état, elle est *considérée* comme une entrée. Les variables internes sont également : les compteurs, les registres, les files d'attente, etc.

c) Des périphéries supplémentaires

Les entrées-sorties et variables internes sont souvent complétées par des composants facultatifs spécialisés :

- des coupleurs de communication : dialogues avec d'autres parties commandes, avec des entrées-sorties déportées, avec un superviseur, etc.
- des cartes intelligentes qui comportent leur propre micro-processeur : comptage rapide, fonctions de régulation, imagerie, gestion d'axes numériques, etc.
- des interfaces pour disques durs, imprimantes, etc.

d) L'unité centrale

L'unité centrale est constituée de différents éléments permettant de répondre à différentes fonctions :

- *le jeu d'instructions* : c'est l'ensemble des instructions disponibles,
- *la mémoire programme utilisateur (RAM)* : elle contient le programme d'application écrit par l'utilisateur ; ce programme peut être transféré en mémoire morte reprogrammable (EEPROM par exemple) après sa mise au point,
- *la mémoire programme système (ROM)* : le programme qu'elle contient est implémenté par le constructeur et n'est pas accessible à l'utilisateur ; ce programme définit comment fonctionne l'automate,
- *la mémoire données utilisateur* : ce sont des adresses disponibles pour l'utilisateur afin qu'il puisse stocker des informations liées à la gestion du processus,
- *la mémoire données système* : adresses en liaison avec le fonctionnement interne de l'automate ; certaines sont à lecture seule pour l'aide au diagnostic, d'autres en lecture-écriture pour la configuration de l'automate,
- *des registres spécialisés* : tempos, compteurs, constantes... à paramétrer et à utiliser dans le programme d'application,
- *des registres temporaires* : le résultat logique (bit) et les accumulateurs (mots), manipulés par le programme système et quelquefois par l'utilisateur pour assurer des fonctions évoluées,
- *l'unité arithmétique et logique, A.L.U.* : elle permet au programme de réaliser des calculs arithmétiques et des combinaisons logiques,
- *les mémoires images des entrées et des sorties* : cette zone mémoire est le *quasi-reflet* des états des entrées et des sorties (d'où le nom d'*image*) ; ces mémoires sont accessibles à la fois par le programme système et par le programme utilisateur (voir § 3 - 4),
- *l'horloge interne* : elle cadence l'exécution du programme.

D'autres fonctions peuvent être disponibles comme le stockage des commentaires ou la gestion horaire des événements.

e) Les bus

- *Le bus d'entrées-sorties*, aussi appelé bus fond de panier, relie les cartes (ou coupleurs) d'entrées-sorties à l'unité centrale ; des standards, comme VME, permettent à des matériels spécialisés fournis par différents constructeurs de cohabiter dans le même châssis.
- *Le bus système* est interne à l'unité centrale : il permet au processeur d'accéder aux différentes ressources.

f) La console de programmation

La console, bien que séparée physiquement de l'automate, doit être considérée comme un de ses éléments constitutifs du fait de l'importance des fonctions qu'elle assure :

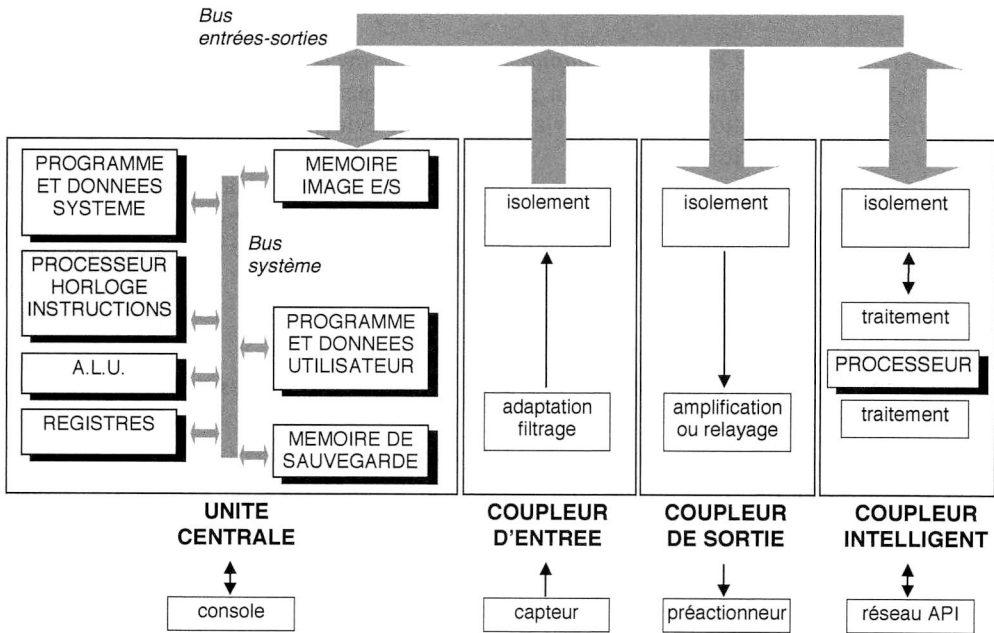
- en phase de programmation,
- en phase de réglage des paramètres,
- en phase de diagnostic et de mise au point grâce aux fonctions primordiales de visualisation et de forçage dynamique des variables, de pose de points d'arrêt, etc.
- en phase de constitution de la documentation et enfin en phase d'exploitation pour la lecture ou la modification de valeurs : durées de temporisations, seuils de comptage, paramètres divers, etc.

La console peut être du type PC. Dans ce cas, elle contient le logiciel de programmation complet, ce qui permet d'accéder à toutes les fonctionnalités. Elle sera mise en oeuvre lors

des phases de programmation et de maintenance par un personnel qualifié. Mieux adaptées aux opérateurs d'exploitation, il existe des mini-console de type « pocket » qui permettent uniquement l'accès aux fonctions non critiques.

Attention à ne pas confondre :

- *logiciel de programmation* : il réside dans la console et propose des outils pour la configuration de l'automate, la définition de la notation symbolique, le réglage des paramètres, la saisie et la modification du programme, le diagnostic...
- *langage de programmation* : il est constitué du jeu d'instructions et de sa syntaxe et est uniquement exploitable par l'automate.



Note : chaque coupleur dispose en général de plusieurs voies.
figure A-7 : exemple de configuration

g) Configuration multi-processeurs

On a vu que les cartes intelligentes disposaient de leur propre micro-processeur pour réaliser des tâches spécialisées, ce qui déleste d'autant le processeur de l'unité centrale. De plus, les modèles moyenne et haute gamme permettent en général l'installation de plusieurs unités centrales dans le même châssis pour le traitement de tâches en parallèle.

3 - 4. Principe de fonctionnement de l'automate programmable

a) Rôle du programme

La *mémoire programme* utilisateur contient la liste des instructions saisie par le programmeur. Ce programme définit le comportement du système :

- en fonction des sollicitations extérieures,
- en fonction des états des variables internes.

b) Notion de scrutation

Afin de permettre des réactions en temps réel (au temps de réponse près), le déroulement du programme repose sur le principe suivant :

- *les instructions sont automatiquement exécutées les unes après les autres*
 - si une condition logique est vraie, alors l'instruction suivante est exécutée et le déroulement du programme se poursuit
 - si une condition logique est fausse, alors l'instruction suivante n'est pas exécutée, sauf si elle est inconditionnelle, et le déroulement du programme se poursuit de toute façon
- *lorsque toutes les instructions du programme ont été lues, le programme se déroule de nouveau depuis le début*
des instructions de saut permettent de structurer l'application et d'organiser le programme afin d'en accélérer l'exécution
- *cette relecture cyclique est ininterrompue, on parle de cycle de scrutation*
 - ce mode de fonctionnement est celui par défaut de tous les automates ; certains modèles autorisent une scrutation périodique (la période doit évidemment être plus longue que la durée normale du cycle)
 - le déroulement normal du programme peut être brièvement interrompu si des tâches prioritaires s'avèrent nécessaires (cette interruption peut être générée soit par occurrence d'un événement, soit périodiquement), après traitement prioritaire le programme est repris à l'endroit où il a été interrompu

c) Durée de cycle

Il est important de se rendre compte des ordres de grandeur des durées mises en jeu dans le déroulement du programme :

- la durée d'exécution d'une instruction se chiffre en *nano* ou *microsecondes*
- selon la longueur du programme, une scrutation a une durée de l'ordre de *quelques dizaines de millisecondes*
- le traitement d'une tâche prioritaire ne dépassera pas *quelques millisecondes*.

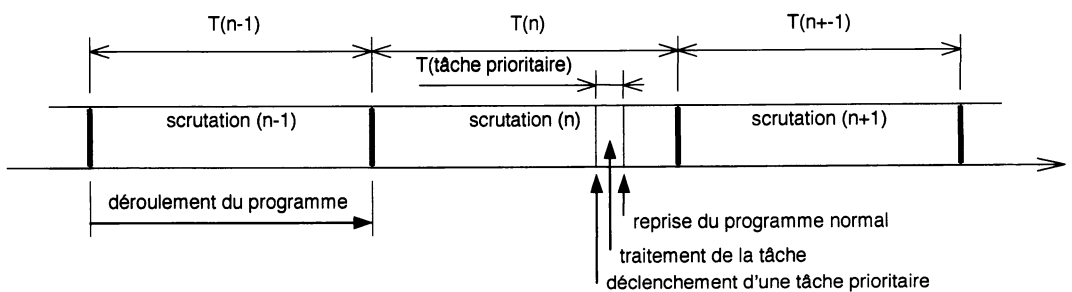


figure A-8

d) Mémoires images des entrées (notées MIE)

L'apparition et la disparition des signaux délivrés par les capteurs ont évidemment lieu indépendamment de l'automate : les états des entrées basculent tout au long du déroulement du programme. Cette instabilité peut générer des aléas de fonctionnement car lorsqu'on développe une application, une entrée est considérée comme étant soit vraie, soit fausse à un moment donné et non les deux à la fois. Pour la durée d'exécution d'un seul cycle, les états des entrées sont rendus stables par les mémoires images des entrées.

Le mécanisme en est le suivant (voir la figure A-9) :

- *en début de scrutation, les états des entrées physiques telles qu'elles sont vues par les coupleurs d'entrées sont mémorisés dans les mémoires-images correspondantes*
 - à chaque entrée correspond une mémoire-image (un bit interne en fait)
 - ce bit interne conserve son état jusqu'au prochain rafraîchissement, même si l'entrée physique correspondante change d'état pendant la scrutation
- *pendant l'exécution du programme, toute instruction qui requiert l'état d'une entrée provoque en réalité la lecture de l'image de cette entrée*
 - en général, le bit interne porte le même nom que l'entrée qu'il représente
 - de ce fait, le programmeur distrait peut penser traiter les entrées physiques alors que le programme qu'il écrit traite leurs images
- *les mémoires-images des entrées sont rafraîchies en début de chaque nouvelle scrutation*
 - si une entrée physique change d'état, son nouvel état ne sera pris en compte par le programme que lors de la scrutation suivante
 - si l'état vrai d'une entrée physique a une durée si brève qu'il apparaît puis disparaît pendant une même scrutation, cette entrée est considérée par le programme comme étant restée fausse.

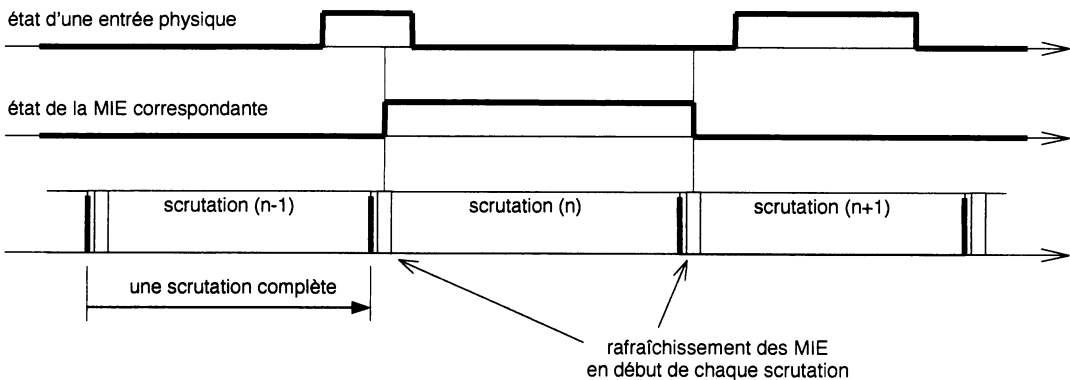


figure A-9

Dans l'exemple de la figure A-9, la première impulsion délivrée par l'entrée est prise en compte par le programme alors que la deuxième ne l'est pas malgré un temps de présence plus long.

e) Mémoires images des sorties (notées MIS)

Un mécanisme similaire met en oeuvre les mémoires images des sorties, interface logique avec les sorties physiques. Là aussi, le programmeur peut penser assigner directement les sorties, alors qu'il assigne en fait des bits internes qui portent simplement les mêmes noms que les sorties.

Le mécanisme est le suivant (voir également la figure A-10) :

- *chaque instruction du programme qui positionne une sortie dans un état déterminé a en réalité comme effet de positionner le bit interne, image de la sortie en question*
la sortie n'est pas affectée par cette instruction, le programme se poursuit

- *chaque nouvelle instruction qui positionne une sortie met la mémoire image correspondante à jour*

l'état de la mémoire-image change d'état au fur et à mesure du déroulement du programme

- *après déroulement complet du programme, les états des mémoires images sont transférés globalement vers les coupleurs de sortie*

l'une des fonctions des coupleurs de sortie est de mémoriser ces états pendant toute la durée du cycle de scrutation suivant, les sorties sont ainsi maintenues dans leur état jusqu'à la fin du prochain cycle

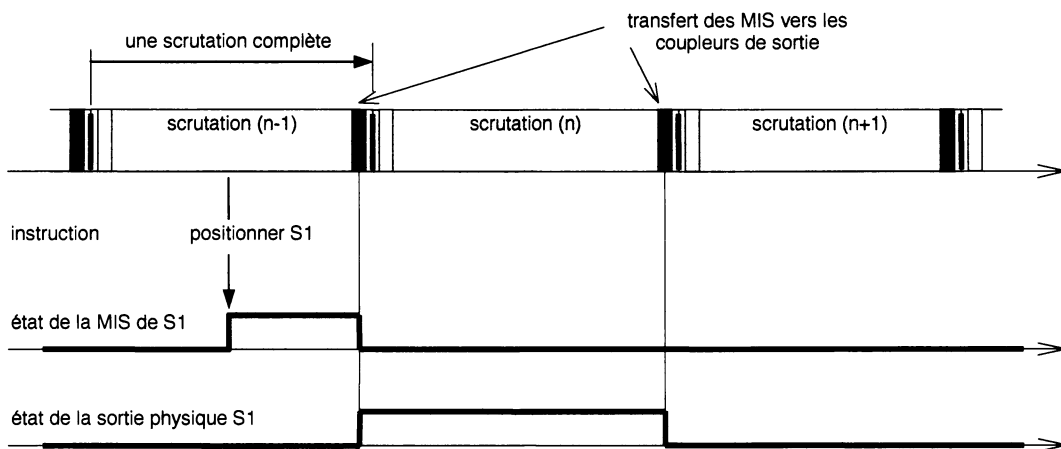


figure A-10

- Instructions monostables et bistables

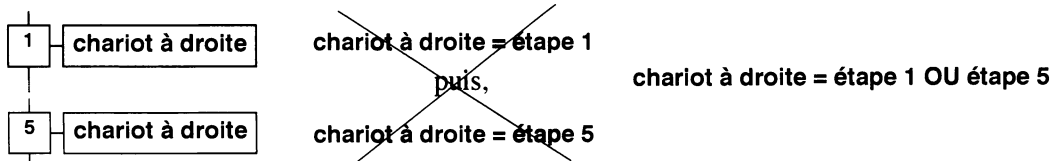
Dans l'exemple ci-dessus, l'instruction *positionner S1* est de type non mémorisée (monostable). Il existe des instructions avec mémorisation du résultat (bistables) : la mémoire image conserve alors son état pendant toutes les scrutations jusqu'à vérification de l'ordre contradictoire.

- Conséquence pour la programmation

L'étude de ce mécanisme montre qu'une sortie ne doit être positionnée que par une seule équation. Si plusieurs équations se présentaient pendant la scrutation, seule la dernière d'entre-elles serait prise en compte car l'automate traite toutes les équations avant d'effectuer la mise à jour de la MIS.

Dans l'exemple suivant, il ne faut pas écrire :

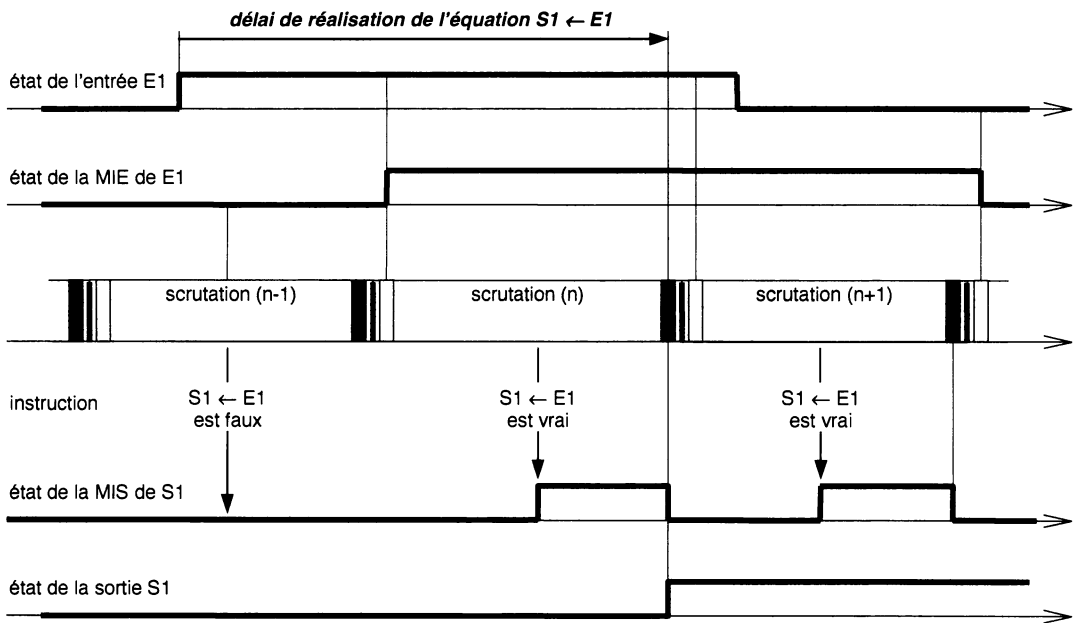
... mais :



Ceci est d'ailleurs conforme à la figure A-5, page 12.

f) Temps de réaction

La figure A-11 illustre le temps de réaction d'une équation combinatoire qui fait intervenir une entrée et une sortie. Ce délai varie de 1 à 2 fois le temps de scrutation, augmenté des temps de traitement des coupleurs, non représentés sur la figure.



Pendant la scrutation (n+2), l'équation $S1 \leftarrow E1$ sera fausse, la sortie retombera donc à zéro dès la fin de cette scrutation.

figure A-11

3 - 5. La programmation des automates

a) Organisation d'un programme

L'ensemble des instructions saisies par l'utilisateur lors de la programmation s'appelle le *programme*. Selon le choix personnel de chacun, il peut être structuré en *sous-programmes* développés séparément, ce qui facilite sa rédaction et sa mise au point. Ces instructions réalisent des *fonctions* de base : calcul logique sur bit ou sur mot, calcul arithmétique, comparaison, gestion de tableaux... Des *blocs fonctionnels* paramétrables sont habituellement disponibles dans les automates afin de répondre à des fonctions plus évoluées (compteurs, temporisateurs, files d'attente, etc.) voire complexes (régulation, asservissements, communication, etc.)

D'une importance capitale, des *variables* (ou données) sont associées au programme. Elles permettent :

- de paramétrer l'application pour une meilleure flexibilité de la machine,
- d'accéder facilement aux valeurs d'initialisation ou de réglage (sans modifier le programme lui-même),
- de stocker des valeurs issues du processus afin de les analyser *à posteriori*.

Exemples :

Le cycle de trempe fait apparaître l'utilisation des *fonctions* logiques ET et OU. Cette application peut être structurée en deux *sous-programmes*, dédiés l'un à l'évolution séquentielle du cycle, et l'autre aux équations des sorties. Le programme comprend également le *bloc fonctionnel standard* de temporisation. La durée de cette temporisation peut être mémorisée dans une *variable*.

b) Les langages de programmation

Le programme doit être saisi dans un langage compréhensible par l'automate. Plusieurs représentations peuvent simultanément être disponibles :

- on distingue d'une part la représentation Grafcet pour l'aspect séquentiel,
- d'autre part, les réceptivités et les actions ainsi que les fonctions qui ne dépendent pas du grafcet sont représentées dans l'un des langages suivants :

langages littéraux : **langages graphiques :** Si la représentation Grafcet n'est pas disponible, il est facile de transcrire les grafkets manuscrits en langage IL ou LD (voir l'exemple au § II.3)

- liste d'instructions
- littéral structuré
- schéma à contacts
- logigramme

SFC, IL, ST, LD et FBD sont les noms de ces 5 langages définis par norme CEI 61131-3

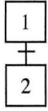
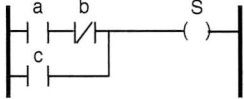
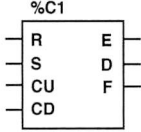
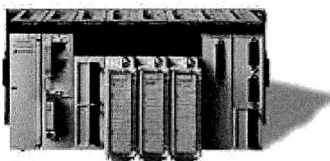
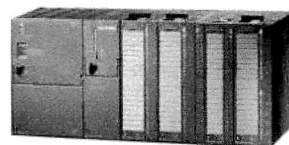
Langage	Exemple	Commentaire
SFC : Diagramme de fonctions en séquence (Sequential Function Chart)		<i>Réservé aux fonctions séquentielles – reprend le formalisme des étapes, des liaisons et des transitions du Grafcet.</i>
IL : Liste d'instructions	LD variable_a AND(variable_b OR variable_c) ST variable_sortie	<i>Langage puissant - nécessite une certaine expérience - permet l'écriture au clavier sans souris.</i>
ST : Littéral structuré	IF variable_a OR variable_b THEN consigne := saisie * 10000 ; END_IF ;	<i>Offre des instructions compactes et permet les structures habituelles des langages informatiques.</i>
LD : Schéma à contacts (Ladder)		<i>Facile à mettre en œuvre grâce à sa ressemblance avec les schémas électriques. Les blocs de programme en langage FBD ainsi que des instructions issues du langage ST sont souvent incrustés dans un tel schéma.</i>
FBD : Bloc fonctionnel		<i>Modules prêts-à-l'emploi pour réaliser des fonctions comme des compteurs, temporisateurs, etc ...Souvent, un paramétrage de chaque bloc fonctionnel sera nécessaire (valeur initiale du compteur ou durée de la tempo par exemple).</i>

figure A-12



Automate Schneider Electric
TSX Premium



Automate Siemens
S7 300

II. PRINCIPES GENERAUX

1 - LE GRAFCET : OUTIL DE MODELISATION

Au chapitre I, le Grafcet a été abordé sous 2 angles différents. Il a d'abord été le résultat d'une observation. Il a ensuite été présenté comme langage de programmation des automates. Cette distinction est de première importance car l'analyse d'un problème ne doit en aucun cas reposer sur un langage de programmation : chacun de ces outils doit être utilisé à bon escient, dans son propre contexte.

Pour éviter toute confusion, on nommera *Grafcet* l'outil de modélisation d'une manière générale (sa sémantique, ses règles d'évolution, sa syntaxe) et *grafcet* un modèle pour un système particulier (un diagramme spécifique). *SFC* désignera le langage de programmation inspiré du Grafcet.

1 - 1. L'essentiel sur le Grafcet

a) La sémantique

Ce paragraphe définit le vocabulaire lié au Grafcet ainsi que sa représentation graphique. La figure A-13 illustre les différents concepts d'une manière synthétique.

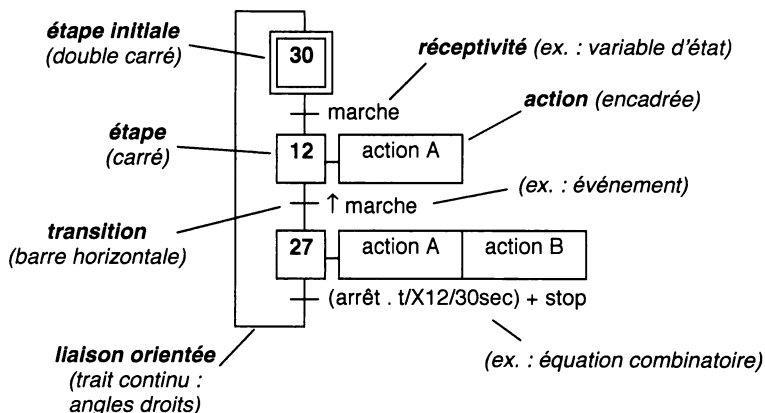


figure A-13

- Etape : active / inactive

- Une étape représente un état particulier du système à un moment donné de son cycle de fonctionnement. L'étape qui correspond à l'état du système à un instant t , est dite active.
- Un état donné peut être déterminé par plusieurs étapes qui sont actives en même temps
- Afin de faciliter le repérage, chaque étape porte son propre numéro. Lorsqu'on désigne une étape, on fait précéder ce numéro de la lettre X majuscule.
- L'ordre des numéros est quelconque et ne traduit en aucune façon le déroulement du cycle (c'est le rôle des liaisons orientées).
- La variable X_i est = 1 lorsque l'étape i est active. $X_i = 0$ lorsqu'elle est inactive.
- L'étape initiale est l'étape qui est active au moment de la mise en marche du système (à ne pas confondre avec la mise en marche du cycle). Un grafcet peut comporter plusieurs étapes initiales.

- **Situation**

- L'ensemble des étapes actives définit la situation du grafcet. La situation est représentée par un vecteur de la forme $\{i, j, \dots\}$ où i, j, \dots sont les étapes actives.
- La situation est dite *vide* lorsqu'aucune étape n'est active. Elle est dite *initiale* lorsque seule l'étape initiale (ou les étapes initiales) est (sont) active(s). La situation *courante* est définie par l'ensemble des étapes actives à l'instant considéré.

- **Action : produite / non produite**

- A chaque étape on peut associer une ou plusieurs actions.
- Il se peut également qu'aucune action ne soit associée à une étape.
- Une action donnée peut être associée à plusieurs étapes.
- Par défaut, les actions sont produites tant que les étapes auxquelles elles sont associées sont actives (ce sont les actions *continues*).
- Les actions peuvent en outre être *conditionnelles*, *mémorisées*, ou encore *limitées* dans le temps ou au contraire *retardées*.

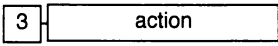
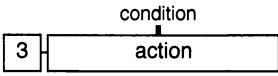
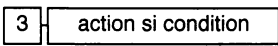
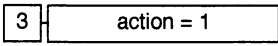
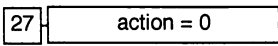
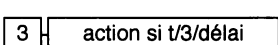
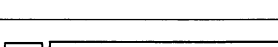
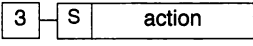
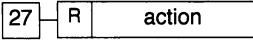
types d'actions	représentation NF C 03-190 sept 95	principe
action continue		l'action est produite tant que l'étape associée est active
action conditionnelle	 ou 	<i>2 représentations possibles</i> l'action est produite tant que l'étape associée est active et que la condition est vraie
action mémorisée	 	l'action est produite dès que l'étape X3 est activée jusqu'au moment où l'étape X27 est activée
action retardée		l'action est produite après un certain délai suite à l'activation de l'étape associée, et ensuite, tant que cette étape est active
action limitée		l'action est produite pendant une certaine durée suite à l'activation de l'étape associée

figure A-14

Les normes CEI 848 et UTE C 03-191 divergent quant à ces symboles : le rectangle qui encadre l'action comporte une ou plusieurs sections supplémentaires.

Exemple :

 pour début d'action mémorisée (Set)

 pour fin d'action mémorisée (Reset)

Différentes combinaisons de S avec les lettres D, L, et C définissent les actions retardées, limitées et conditionnelles (avec mémorisation ou non). Cette symbolique est effectivement liée au modèle Grafcet *mais peut être confondue avec celle du langage SFC, pratiquement identique.*

- Liaison orientée, entrance / sortance

- Les étapes sont reliées entre-elles par des liaisons. Ces liaisons sont orientées et définissent l'ordre d'activation des étapes. L'évolution générale du grafcet se fait du haut vers le bas, aussi ne représente-t-on pas obligatoirement de flèches sur les liaisons.
- L'entrance d'une étape (par où arrive une liaison) est *toujours* sur son côté supérieur. La sortance *toujours* sur son côté inférieur.
- On parle également de côté aval ou côté amont d'une étape.

- Transition : validée / non validée

La liaison entre deux étapes comporte une et une seule transition. Cette transition permet l'évolution du grafcet de l'étape précédente vers l'étape suivante. Lorsque l'étape précédente est active, on dit que la transition est validée.

- Réceptivité : vraie / fausse

De même que des actions sont associées aux étapes, des réceptivités sont associées aux transitions. Les réceptivités sont toujours des propositions logiques dont le résultat ne peut être que vrai ou faux. On distingue d'une part les niveaux 0 et 1, qui sont stables, et les passages d'un niveau à l'autre, qui sont par conséquent transitoires, et qu'on appelle événements. Ils sont notés de la manière suivante :

niveau		événement	
= 1	v	0 → 1	↑ v : front montant
= 0	v̄ (ou / v) : v barre	1 → 0	↓ v : front descendant

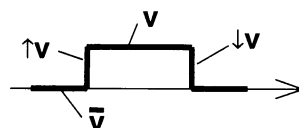


figure A-15

Le résultat logique peut également être formé d'après des opérations numériques, être la conséquence de l'évolution d'un temporisateur ou encore être toujours vrai.

résultat d'une opération	résultat d'une temporisation		résultat toujours vrai	
exemple (comparaison) : [compteur < (1000+consigne)]	exemple : t/X12/30sec	la temporisation qui a démarré à l'étape X12, et qui a duré 30 sec, prend fin (figure A-13)	=1	cette réceptivité est toujours vraie, aucune autre condition n'est attendue

figure A-16

b) Les règles d'évolution

L'évolution d'un grafcet se fait par franchissements successifs des transitions. Le modèle Grafcet définit d'une manière très précise comment s'opèrent ces franchissements.

- Franchissement d'une transition

Plusieurs phases sont mises en oeuvre. On rappelle que l'activité d'une étape a comme effet de valider la transition en aval (figure A-17-b). Cette première condition est insuffisante (figure A-17-c) : la réceptivité doit être vraie pour que la transition soit franchie. La conséquence du franchissement d'une transition est d'activer immédiatement

l'étape suivante et simultanément de désactiver l'étape précédente. Une nouvelle transition sera donc validée sitôt la transition en question dévalidée (figure A-17-d).

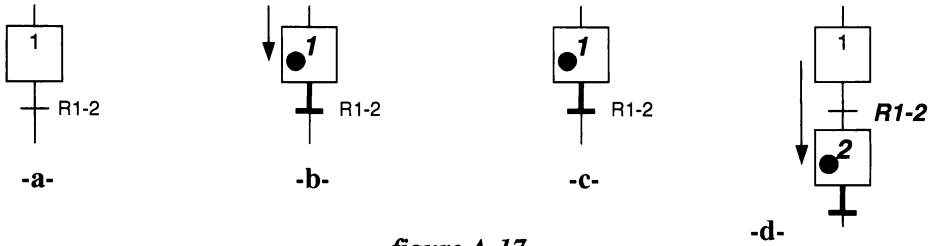
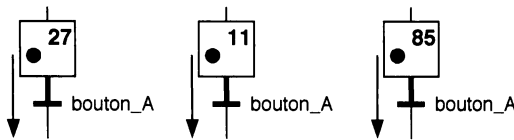


figure A-17

- Simultanéité des évolutions

Si plusieurs transitions sont franchissables à un instant donné, elles sont toutes franchies en même temps. Aucune priorité n'est donnée à l'une d'entre elles. Ce cas se produit par exemple lorsque plusieurs transitions sont validées (parce que plusieurs étapes sont actives) et que la même réceptivité leur est associée.



- 1) X27, X11 et X85 sont actives
- 2) on appuie ensuite sur le bouton A
- 3) les 3 transitions sont franchies en même temps sans aucun ordre de priorité

figure A-18

- Situation Initiale

Selon les règles déjà énoncées, au moins une transition doit être validée pour permettre une première évolution. Cette situation peut être obtenue de 2 manières : par les transitions sources (voir le paragraphe suivant), ou par activation d'au moins une étape au démarrage du système : ce sont les étapes initiales, représentées par un carré double.

c) La syntaxe

- Alternance entre étapes et transitions

Le grafset évolue d'étape en étape à chaque fois qu'une transition est franchie. De ce fait, les liaisons orientées doivent comporter *une et une seule* transition entre 2 étapes.

- Source et puits

Il arrive qu'une transition doit pouvoir être franchissable si la réceptivité associée est vraie *quelle que soit la situation du grafset*. La validation de cette transition est alors permanente et ne dépend donc d'aucune étape précédente. C'est une transition source (comme une source de montagne, elle est toujours alimentée). Elle est utilisée lorsque l'apparition de la réceptivité qui lui est associée doit avoir un effet prioritaire sur tout le reste de l'application (appui sur le bouton arrêt d'urgence, perte de pression dans le circuit pneumatique, ouverture inopinée du capot de protection, etc.)

A l'inverse, le franchissement d'une transition peut ne pas activer d'étape suivante. L'étape précédente est cependant désactivée. La marque que véhicule le grafset (le point noir qui symbolise l'activité des étapes) est alors perdue (comme dans un puits, d'où le nom de cette

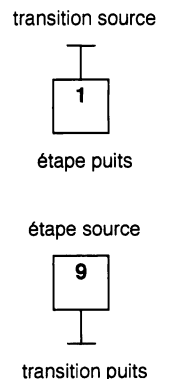


figure A-19

transition particulière). De la même manière, les étapes peuvent être source ou puits.

- Structures de base

Les structures sont les différentes manières d'organiser les liaisons orientées. Elles respectent toutes l'alternance entre les étapes et les transitions, ce qui explique les places occupées par les transitions. Ce point mérite une attention particulière. La figure A-20 illustre les structures de base. Le grafcet de la figure A-20-a comporte un aiguillage vers 2 séquences distinctes. L'une d'elles seulement pourra être exécutée. L'aiguillage peut être étendu à plus de 2 séquences. On parle de structure à sélection de séquence ou de divergence en *ou*. La figure A-20-b représente la convergence de telles séquences.

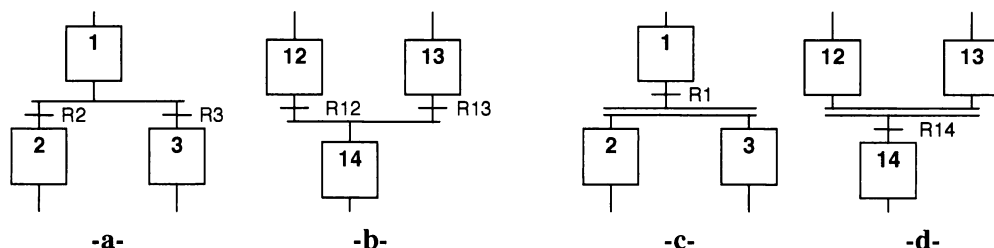


figure A-20

Les figures c et d représentent divergence et convergence (en *et*) de séquences simultanées. L'étape X1 valide la transition en aval. Si la réceptivité R1 est vraie, ce sont *toutes* les étapes suivantes qui sont activées (X2 *et* X3 dans ce cas). Par ailleurs, la transition en amont de X14 (figure d) n'est validée que si *toutes* les étapes qui la précèdent sont actives (X12 *et* X13 dans ce cas). Celles-ci seront *toutes* désactivées lors du franchissement de cette transition.

Dès la fin de chaque séquence simultanée, on ajoute une étape d'attente. Ainsi, chaque séquence évolue à son propre rythme et atteint son étape d'attente. La *resynchronisation* a lieu lorsque toutes les séquences sont terminées. Plus aucune autre condition n'est nécessaire pour poursuivre le cycle, c'est pourquoi la réceptivité en aval de la convergence est une réceptivité toujours vraie. Notons que ceci est le seul cas où cette réceptivité particulière est employée sans ambiguïté pour l'évolution du grafcet (ce point est développé au chapitre II.3 -).

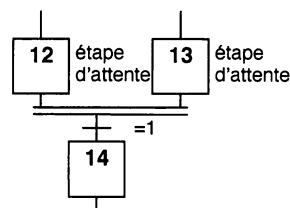


figure A-21

- Structures habituelles

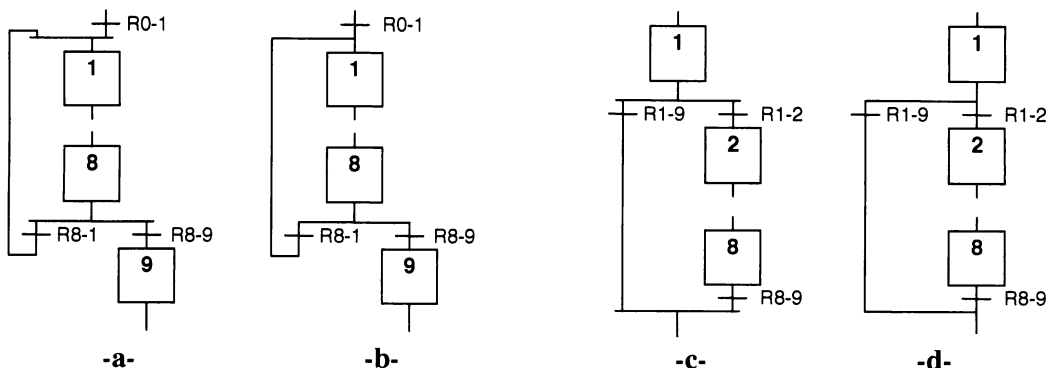


figure A-22

Les structures de base donnent lieu à des configurations souvent rencontrées (figure A-22). Il s'agit de la reprise de séquence (a et b) et du saut d'étapes (c et d).

Signalons la possibilité de graphismes simplifiés (figures b et d), que l'on rencontre la plupart du temps dans les logiciels d'édition.

- Gestion de priorités

La structure à sélection de séquence pose des questions particulières. Dans le cas de la figure A-20-a par exemple, les réceptivités R2 et R3 peuvent très bien être vraies *avant* l'activation de l'étape X1. Cette activation valide les 2 transitions en question. Les réceptivités étant vraies, et, *les transitions simultanément franchissables devant être franchies simultanément*, les séquences qui commencent par les étapes X2 et X3 sont exécutées toutes les deux.

A propos :

Réponse :

Est-il exact que les 2 séquences seront réellement exécutées ?

Pour le modèle Grafcet : oui. Au niveau de la réalisation : c'est moins certain. En effet, les possibilités de réalisation d'un grafcet sont nombreuses. Le plus souvent, il est traduit en programme automate. Selon la marque de l'automate utilisé, son langage et la manière de programmer, il se peut que ce grafcet évolue effectivement vers les 2 séquences. Il se peut également qu'une seule séquence soit exécutée.

Si cette représentation peut créer un malentendu, le concepteur doit obligatoirement choisir une solution plus explicite. Dans le cas où R2 et R3 peuvent effectivement être vraies avant l'activation de X1, plusieurs possibilités se présentent :

- soit les 2 séquences doivent être exécutées,
- soit aucune séquence ne doit être exécutée,
- soit l'une d'elles seulement doit être exécutée.

Les grafquets de la figure A-23 illustrent ces divers fonctionnements sans équivoque.

$R2 = 1$ et $R3 = 0 \Rightarrow$ séquence X2
 $R3 = 1$ et $R2 = 0 \Rightarrow$ séquence X3
 $R2 = 1$ et $R3 = 1 \Rightarrow$ séquences X2 et X3
 $R2 = 1$ et $R3 = 0 \Rightarrow$ séquence X2
 $R3 = 1$ et $R2 = 0 \Rightarrow$ séquence X3
 $R2 = 1$ et $R3 = 1 \Rightarrow$ arrêt sur X1
 $R2 = 1$ et $\forall R3 \Rightarrow$ séquence X2 (priorité à la séquence X2)
 $R3 = 1$ et $\forall R2 \Rightarrow$ séquence X3 (priorité à la séquence X3)

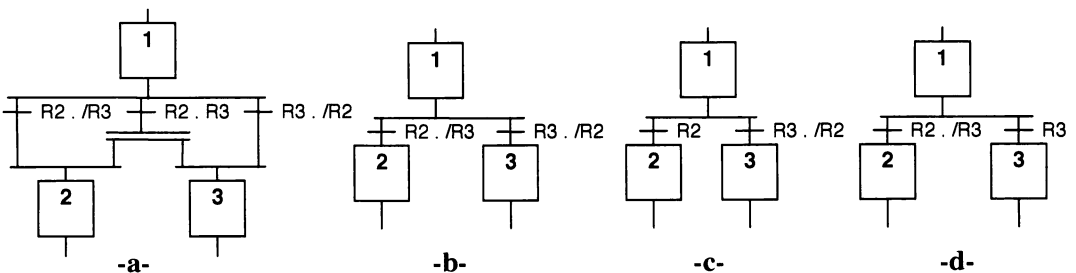


figure A-23

1 - 2. Dialogues entre grafquets

Une étude complète se décompose en général en plusieurs fonctions décrites chacune par un grafcet spécifique. Les modes de marches et d'arrêts se traduisent également par de multiples grafquets. Des dialogues entre eux devront donc s'établir.

a) Niveaux hiérarchiques

L'organisation de tous ces grafquets se fait sur plusieurs niveaux hiérarchiques (figure A-24). Au niveau supérieur, le *grafcet de surveillance* est chargé d'autoriser ou non le

fonctionnement du cycle de production. Le *grafcet de conduite* gère les modes de marches et d'arrêts et assure l'unicité du mode (un seul mode à la fois doit être actif). Chaque mode peut ensuite être décrit par un *grafcet spécifique*, le *grafcet de commande* du mode en question. Des tâches subalternes peuvent être définies par des *grafcets de tâches* de niveaux inférieurs.

b) Niveaux hiérarchiques

En principe, les dialogues entre ces *grafcets* ne se font que verticalement. Du haut vers le bas, ce sont des *ordres*, du bas vers le haut, des *comptes-rendus*.

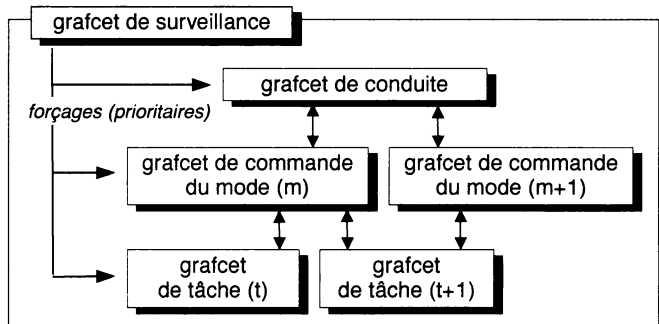


figure A-24 : exemple de grafcets hiérarchisés

c) Représentation et modes de fonctionnement

Plusieurs possibilités de représentations sont offertes. En premier lieu, il faut distinguer les dialogues liés à la gestion des tâches des ordres généralisés pour assurer la surveillance. Les ordres qui concernent cette seconde catégorie sont prioritaires sur les autres dialogues : ce sont des *forçages* de situations (figure A-24).

- Gestion des tâches

Le tableau ci-dessous montre différentes versions d'une étape qui émet un ordre d'exécution vers un *grafcet* de niveau hiérarchique inférieur.

Tâche	Séquence	« Sous-programme »	Macro-étape
proposé à l'origine par le Grepa 	CEI 848 / UTE C 03-191 	NF C 03-190 	René David / UTE C 03-191
Ce concept fait partie du Grafcet et n'introduit donc pas d'extention supplémentaire. Il permet toutes sortes de combinaisons en jouant sur les seules règles du Grafcet et est adapté à tous les langages d'automates. (voir ci-dessous)	Le symbole du rectangle, déjà attribué aux actions, peut de ce fait prêter à confusion, notamment lors de la programmation. Le concept est identique à celui de la tâche.	Ce terme est impropre car il fait référence à <i>quelque chose de semblable</i> dans le domaine informatique. Il est d'ailleurs systématiquement placé entre guillemets dans la norme. Les sous-programmes effectivement rencontrés lors de la programmation sont un concept différent, attention aux confusions possibles	Les macro-étapes permettent également de décomposer un <i>grafcet</i> en plusieurs parties. Elles s'ajoutent aux règles de base du Grafcet.

figure A-25

L'expansion d'une *macro-étape* se distingue des *grafcets* tâche, séquence et « sous-programme » par une formalisation particulière (figure A-26) :

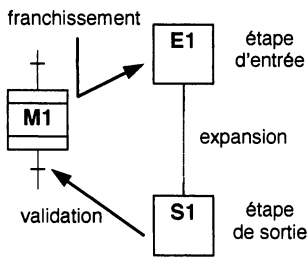


figure A-26

- si la macro-étape (du grafcet appelant) doit être activée, c'est en réalité l'étape d'entrée de son expansion (grafcet appelé) qui est activée,
 - la transition en aval de la macro-étape est validée par activation de l'étape de sortie de son expansion.
- L'expansion peut en fait s'insérer en lieu et place de la macro-étape. Elle n'a pas d'étape initiale.

Quant aux *tâches*, *séquences* et « *sous-programmes* », les grafquets appelés répondent à la structure de la figure A-27-a. Le grafcet appelé nécessite une étape initiale et

en fin de séquence, une étape de compte-rendu : sur l'exemple de cette figure, l'activation de l'étape X1 suffit à faire évoluer le grafcet de tâche car sa première transition est *validée* dès le démarrage du système. On voit qu'aucune nouvelle règle n'est nécessaire. De même, l'activation de l'étape de compte-rendu (X19) permet l'évolution du grafcet appelant. Dès lors, le grafcet de tâche est réinitialisé pour un usage ultérieur (réceptivité /X1).

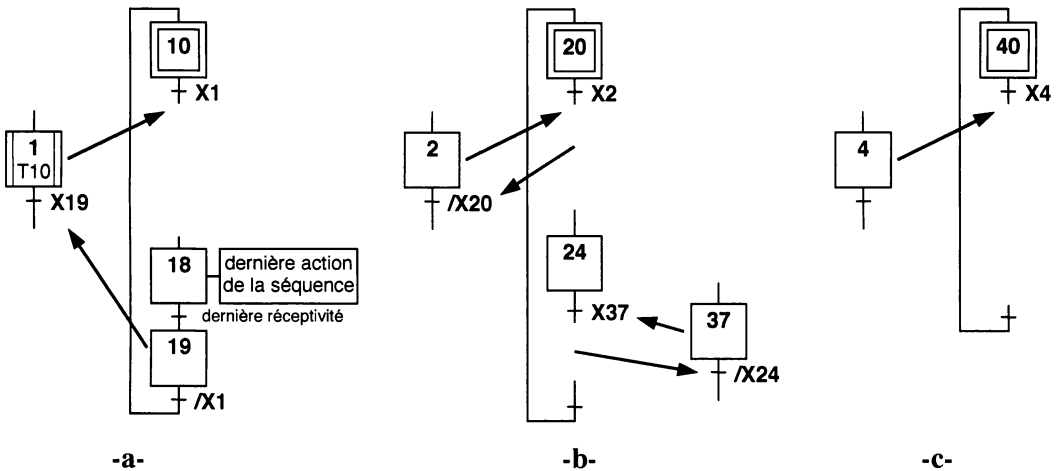


figure A-27

La synchronisation complète qui vient d'être présentée est la plus courante (le grafcet appelant n'évolue que si la séquence appelée est terminée). Mais basées sur ce principe général, d'autres formes de dialogues sont possibles.

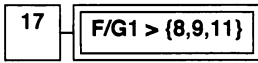
- Le grafcet de tâche de la figure A-27-b est exécuté sur activation de l'étape X2. Les évolutions des deux grafquets sont ensuite indépendantes l'une de l'autre.
- Le grafcet de tâche de la figure A-27-c exécute sa séquence en boucle tant que X4 est active.

On rappelle que dans aucun de ces cas une quelconque action n'a été associée à l'étape appelante. Le seul fait de son activation suffit à exécuter la séquence.

- Forçages de situations

Le forçage consiste à imposer une situation déterminée à tous les grafquets d'une application quelle que soit leur situation courante. Du fait que ces ordres sont prioritaires par rapport aux autres dialogues, il est conseillé de les réserver au niveau *surveillance*. Des conflits de priorité risquent sinon de remettre en cause la sécurité. Par conséquent, un grafcet forcé

dans une situation donnée ne peut pas évoluer tant que l'ordre de forçage est présent. Le symbole défini par la norme UTE C 03-191 est le suivant :



Tant que l'étape X17 est active, les étapes X8, X9 et X11 du grafçet G1 sont actives et les autres étapes de ce grafçet sont inactives.

figure A-28

Cette notion permet de nombreux dérivés : figeage dans la situation courante, forçage des états des sorties, etc. Pour éviter d'encombrer le modèle Grafçet avec une iconographie supplémentaire, la norme se limite au symbole présenté ci-dessus. Les intentions particulières du concepteur peuvent donc être basées sur ce symbole et complétées *en clair*. En réponse aux cas souvent rencontrés, on peut admettre la symbolisation suivante :

- F/G1 > { } → forcer la situation vide de G1 : aucune étape n'est active
- F/G1 > { * } → figer la situation courante de G1 : les étapes conservent leur état
- F/G1 > { INIT } → forcer la situation initiale de G1 : seules les étapes initiales sont actives

2 - NIVEAU DE PRECISION D'UN GRAFÇET

Les notions de quantité et de qualité d'une information peuvent être illustrées par l'exemple suivant : quand on dit qu'il pleut, l'information *il pleut* exprime la même quantité d'informations pour un promeneur que pour un météorologue. Mais la qualité de cette information n'est pas la même pour tous : pour le promeneur, elle signifie *ne pas oublier de prendre le parapluie*, pour le météorologue, *mettre à jour les statistiques*. Ce qui est mis en évidence par cette représentation multiple d'un même phénomène est la notion de point de vue.

2 - 1. Structure d'une chaîne fonctionnelle

a) Notion de chaîne fonctionnelle

La figure A-29 montre la structure générale des liens à la fois physiques et informationnels qui mettent en relation la partie commande et la partie opérative pour la réalisation d'une seule fonction du système. La partie commande peut être considérée comme étant le *cerveau* du système, la partie opérative les *muscles* et les chaînes qui les relient seraient alors les *nerfs* : la chaîne d'action transmet les *ordres* alors que la chaîne d'acquisition véhicule les *comptes-rendus*. Ensemble elles constituent une chaîne fonctionnelle destinée à remplir une fonction. Un système complet fait intervenir plusieurs chaînes semblables.

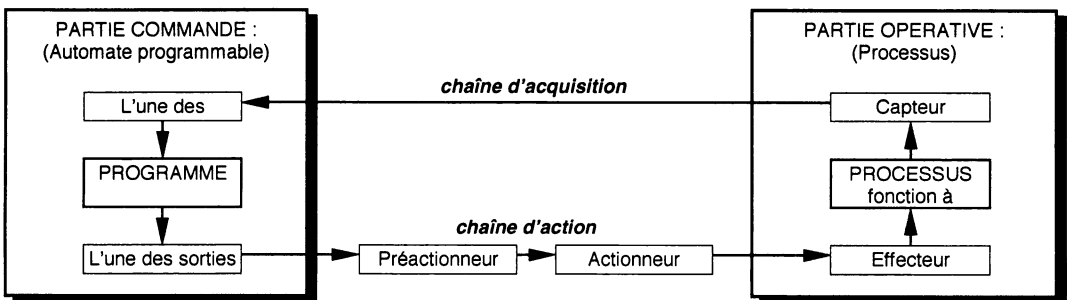


figure A-29

b) Chaîne d'acquisition

Le type et le niveau d'énergie de nombreux capteurs sont directement compatibles avec les cartes d'entrées des automates (par exemple 24V=). La liaison entre ces deux composants ne nécessite donc en général pas d'interfaçage particulier.

c) Chaîne d'action

Les actionneurs par contre utilisent des énergies de type et de niveau différents de ceux des cartes de sorties des automates (moteur 380V triphasé, vérin hydraulique, etc.) La conversion du type et/ou du niveau d'énergie doit donc obligatoirement être prise en charge par un composant supplémentaire : le préactionneur (contacteur, distributeur, etc.) En amont du préactionneur, on parle de *circuit de commande*, et en aval, de *circuit de puissance*.

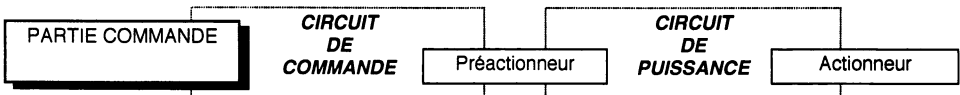


figure A-30

2 - 2. Etude détaillée d'une chaîne d'action

Soit l'exemple d'une caisse qu'il faut déplacer à un moment donné dans un cycle automatique de palettisation.

- Des phénomènes liés à ce déplacement se produisent en plusieurs endroits de la chaîne d'action (d'où les *niveaux* a, b, c, d et e ci-dessous).
- En chaque endroit, on peut porter un regard extérieur sur le système (c'est le *point de vue de l'observateur* dont les outils sont autant le voltmètre que la vue).
- On peut également considérer ces phénomènes du *point de vue du système* ; on se place alors « à l'intérieur » des composants.

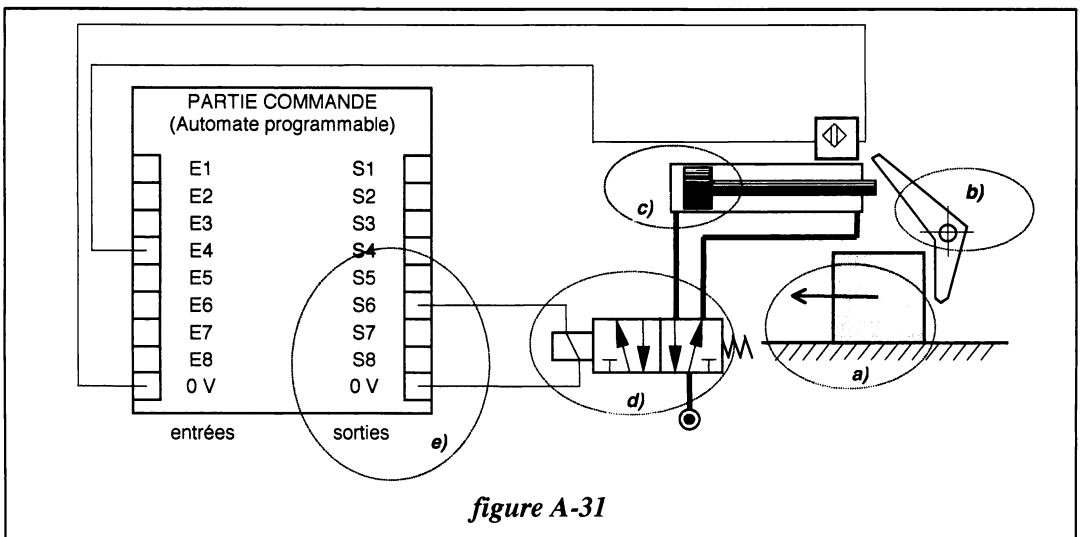
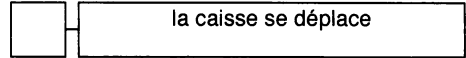


figure A-31

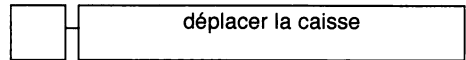
a) Au niveau de la fonction à réaliser

L'endroit observé est la caisse elle-même car le seul objectif de cette chaîne fonctionnelle est de la déplacer. Les 2 points de vue sont notés différemment : un verbe au présent pour exprimer que le phénomène est en train de se produire, un verbe à l'infinitif pour exprimer que l'action doit être produite par le système.

- du point de vue de l'observateur extérieur



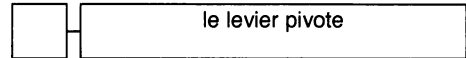
- du point de vue du système



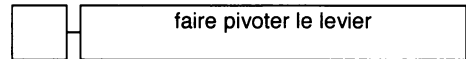
Pour l'observateur, la proposition *la caisse se déplace* est une *action* réalisée par le système. Par contre, pour qu'elle puisse se déplacer, le système doit donner un *ordre*, qui est *déplacer la caisse*. A ce niveau, la technologie employée n'est pas évoquée. La caisse pourrait aussi bien se déplacer sous l'effet de l'inclinaison de son plan d'appui que par la mise en marche d'un convoyeur à rouleaux.

b) Au niveau de l'effecteur

- du point de vue de l'observateur extérieur :



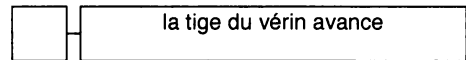
- du point de vue du système :



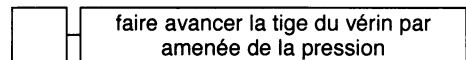
Là aussi, l'observateur constate une action alors que le système émet l'ordre qui doit la provoquer. On a défini la solution technique qui permet d'agir sur la caisse : un levier.

c) Au niveau de l'actionneur

- du point de vue de l'observateur extérieur :



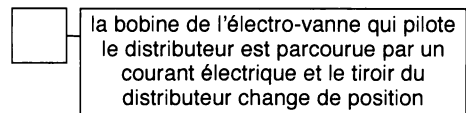
- du point de vue du système :



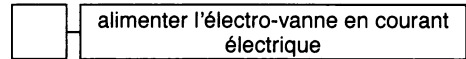
A ce niveau de représentation, c'est la technologie de l'actionneur qui est définie, un vérin pneumatique à double-effet par exemple.

d) Au niveau du préactionneur

- du point de vue de l'observateur extérieur :



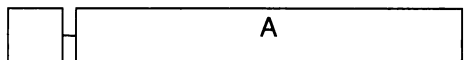
- du point de vue du système :



L'observateur peut mesurer la tension qui apparaît aux bornes de l'électrovanne. Le système doit fournir cette tension en établissant un circuit électrique. Le choix technologique concerne le préactionneur : par exemple un distributeur monostable à commande électrique.

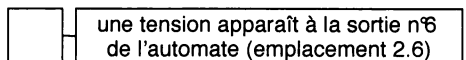
L'écriture de cet ordre est souvent simplifiée en définissant un symbole équivalent.

L'étape du grafcet s'écrit alors comme ceci :

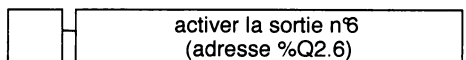


e) Au niveau de la sortie de l'automate

- du point de vue de l'observateur extérieur :

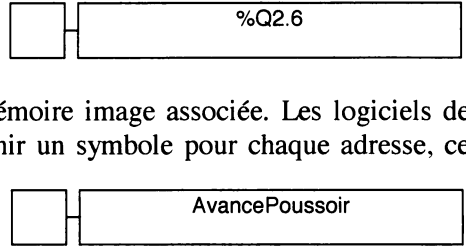


- du point de vue du système :



Cette fois, on peut mesurer la tension sur la borne de la sortie n°6 de l'automate. Du point de vue du système, c'est un ordre qui est généré par le programme. A ce niveau, on définit le type d'automate et le câblage de ses sorties.

Dans le grafcet destiné à l'écriture du programme d'automate, on note le numéro de l'emplacement défini pour le câblage. Ce numéro est également celui de l'adresse de la mémoire image associée. Les logiciels de programmation des automates permettent de définir un symbole pour chaque adresse, ce qui facilite la maintenance du programme (à condition que les symboles choisis soient explicites). L'étape ci-dessus devient alors :



figures A-32 a-e

2 - 3. Relations de cause à effet

Les observations faites successivement mettent en évidence la notion de chaîne dont les maillons sont des relations de cause à effet.

- *Toute action observable est l'effet d'un ordre émis par un élément du système.*
- *Tout ordre émis par un élément du système provoque une action observable.*

Sur la figure A-33, le rectangle 1 est la cause qui provoque l'effet décrit par le rectangle 2, qui est lui-même cause de l'effet du rectangle 3 et ainsi de suite jusqu'au rectangle 10. Le grafcet peut être écrit à l'un quelconque de ces niveaux et points de vue.

Si une confusion sur la nature des déclarations peut subsister, on précise le point de vue et le niveau des grafkets.

Il faut également garder à l'esprit que le programme n'agit pas sur la caisse, ni sur le vérin, ni même sur le distributeur : seulement sur la sortie de l'automate.

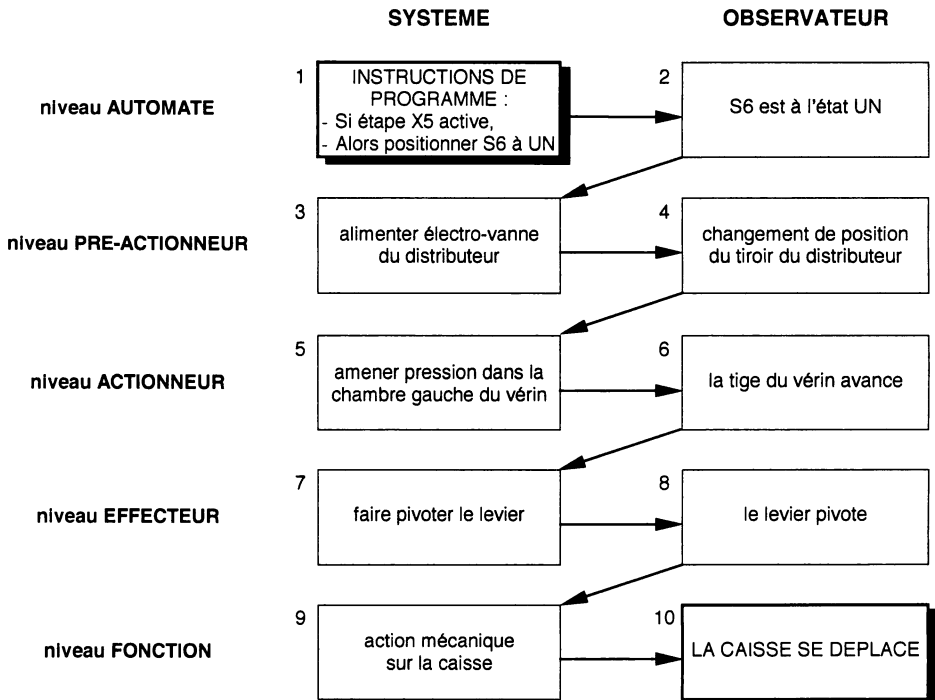


figure A-33

3 - GRAFCETS AU NIVEAU AUTOMATE

Sur certains modèles d'automates, l'algorithme de programmation est laissé à l'initiative de l'utilisateur. (L'algorithme est la manière d'organiser le programme, l'ordre dans lequel les instructions sont exécutées). Il doit être élaboré dans le respect des règles du Grafcet dont certaines sont difficiles à mettre en oeuvre d'une manière tout à fait rigoureuse. Il en est ainsi de la simultanéité de franchissement des transitions franchissables. En effet, le temps interne de traitement de l'automate, aussi court soit-il, ne peut jamais être réellement nul, ce qui empêche cette simultanéité. Une bonne maîtrise de ces phénomènes est primordiale, soit pour produire soit-même un algorithme correct, soit pour utiliser à bon escient celui implémenté par le constructeur le cas échéant.

3 - 1. Structure générale d'un programme

a) Cas d'un grafcet unique

Un grafcet complet peut être décomposé en 2 parties comme le montre la figure A-34. On a d'une part le grafcet proprement dit, à savoir la succession d'étapes-transitions (réceptivités comprises) et d'autre part les actions qui sont associées aux étapes. Les actions, ou les ordres qui les génèrent, dépendent directement des états des étapes. D'une manière générale, le programme doit donc mettre à jour les étapes du grafcet dans un premier temps, et dans un deuxième temps seulement, produire les actions en fonction de la nouvelle situation du grafcet.

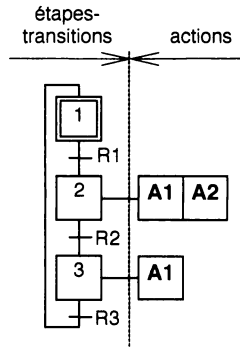


figure A-34

- Partie grafcet

La mise à jour du grafcet se fait par analyse à la fois des étapes actives (situation courante) et des réceptivités qui sont vraies. Cette évolution se fait par franchissement des transitions franchissables. Pour *chaque transition* du grafcet, on réalise l'algorithme général suivant : (*)

Si étape précédente active
 Et si réceptivité vraie
 Alors activer l'étape suivante
 Alors désactiver l'étape précédente

(ordre quelconque)

1 module par transition dans un ordre quelconque

Si le langage de l'automate n'intègre pas la représentation graphique du grafcet, cet algorithme peut facilement être codé sous forme de liste d'instructions ou sous forme de schéma à contacts.

- Partie actions

Le grafcet ayant évolué, la 2° partie du programme doit produire les actions associées aux étapes actives. Dans l'exemple de la figure A-34, l'équation logique de l'action A1 s'écrit :

$$A1 \leftarrow X2 + X3$$

Si l'action A1 est une sortie qui agit sur la partie opérative, son équation est souvent enrichie de conditions supplémentaires : sécurité, modes de marches, affinement du fonctionnement, etc.

(*) D'autres techniques de programmation peuvent être envisagées. L'objet de ce chapitre étant de présenter les principes généraux, elles ne seront pas abordées.

b) Partie préliminaire

La formation de la réponse à un front ou le positionnement du résultat d'une équation logique particulièrement ardue peuvent être traités dans une partie supplémentaire du programme (exemple de la figure A-35). Cette partie doit être exécutée *avant toute autre* afin que ses résultats puissent être exploités ensuite. Le regroupement de ces calculs permet une meilleure structuration du programme.

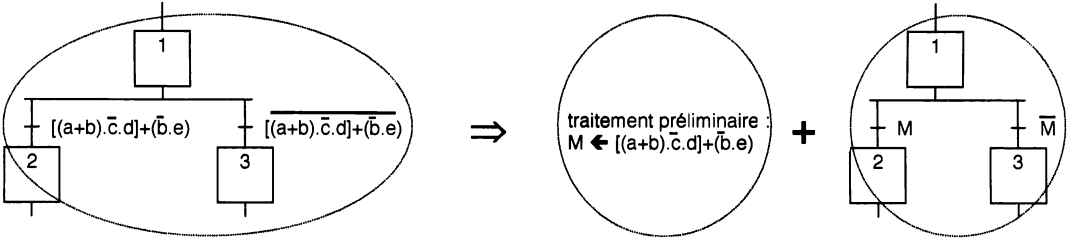


figure A-35

c) Cas d'un programme qui comprend plusieurs grafquets

Une sortie peut très bien être produite par des étapes qui appartiennent à des grafquets différents. Une sortie ne devant par ailleurs être positionnée que par une seule équation, on préfère d'une manière générale structurer le programme comme suit :

Traitement préliminaire
 Traitement du grafquet g-1
 Traitement du grafquet g
 Traitement du grafquet g+1
 Traitement des sorties

}] dans un ordre quelconque

Cette généralisation n'est pas utile pour toutes les sorties lors de l'étude initiale du système, mais elle garantit la pérennité des instructions existantes en cas de modifications ultérieures (ajout de grafquets en particulier). La mise à jour du programme est plus sûre du fait de sa modularité.

3 - 2. Programmation des actions

a) Stabilité d'une étape

Soit la figure A-36. Supposons que la réceptivité $r2$ soit déjà vraie au moment de l'activation de l'étape X1. La transition associée est validée dès cette activation et donc *immédiatement* franchie. Le temps d'activation de l'étape X1 est proche de zéro : l'étape X1 est dite *instable*.

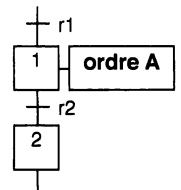


figure A-36

Ce qui compte dans le grafquet au niveau automate, c'est moins le temps de cette activation que la possibilité pour le programme de produire l'ordre qui est associé à l'étape X1.

b) Conséquence

- Si l'on considère le modèle Grafcet

Le modèle Grafcet précise que la durée d'activation d'une étape et la durée de franchissement d'une transition ne peuvent jamais être tout à fait nulles : les ordres associés à des étapes instables doivent donc être exécutés.

- **Si l'on considère l'algorithme général de programmation ...**

... qui consiste à ne traiter les actions qu'après *complète* mise à jour des étapes du grafcet, le résultat est le suivant : le traitement du grafcet a comme effet d'activer directement l'étape X2 dès vérification de la réceptivité **r1** ; au moment du traitement de l'équation de A (qui est $A \leftarrow X1$), cet ordre n'est pas produit car X1 *n'est plus* active. *Les ordres associés à des étapes instables ne sont donc pas exécutés.*

- **Première conclusion**

S'il s'agit d'un grafcet descriptif ou de spécification, des ordres peuvent être associés à des étapes instables. Plus de prudence est de rigueur si le grafcet est destiné à la programmation. Pour simplifier, la règle suivante s'applique à tous les cas de figure :

En règle générale, on ne peut associer aucune action à une étape instable.

- **Si l'on considère les solutions des constructeurs**

En principe, la norme CEI 61131-3 standardise les démarches, mais le renouvellement du parc des automates s'étendra encore sur plusieurs années car la lourdeur des investissements en matière de maintenance freine souvent l'acquisition de nouveaux modèles. L'utilisateur doit donc prendre en compte la variété des principes de programmation, d'autant que la connaissance d'un seul type d'automate risque de le conduire à des habitudes fâcheuses.

c) Exemple 1 : gérer des données

Soit à gérer des données lorsque X12 est active (figure A-37). Envisageons les deux types de mises en oeuvre : sur étape instable, et sur étape stable.

- **Sur étape instable**

La réceptivité toujours vraie en aval de X12 autorise l'évolution immédiate du grafcet dès activation de X12. Pour que l'ordre de gestion des données soit pris en compte, l'algorithme général de programmation doit être aménagé comme suit :

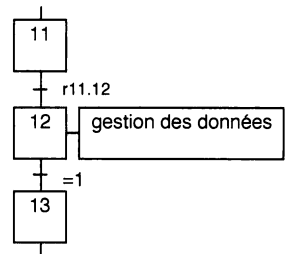


figure A-37

Si étape X11 active

Et si [r11.12] vraie

Alors activer l'étape X12

Alors désactiver l'étape X11

Alors gérer les données ← insertion du traitement interne

Si étape X12 active

Alors activer l'étape X13

Alors désactiver l'étape X12

Notons que la désactivation de X12 a lieu seulement après la gestion effective des données. De plus, les données n'auront été traitées *qu'une seule fois*, ce qui est suffisant d'une manière générale, et même obligatoire dans certains cas (empilement d'une information dans une file d'attente par exemple).

Cette insertion se traduit de manière différente selon le langage automate utilisé.

A titre d'exemple, le grafcet de la figure A-37 est transcrit ci-dessous en langage IL et en langage SFC :

Une mémoire d'étape est matérialisée par un bit interne de type %Mi, avec *i* le numéro d'étape. En langage SFC, une procédure de fonctionnement complète qui traduit les règles du Grafcet est implantée dans l'automate, et une mémoire d'étape est alors notée %Xi.

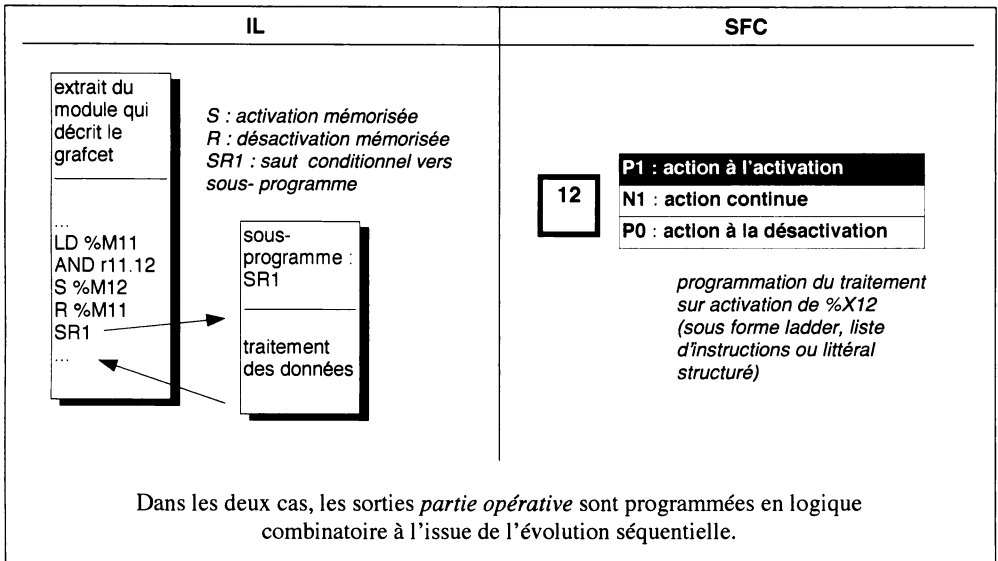


figure A-38

- Sur étape stable

Dans ce cas, on applique l'algorithme général : la gestion des données est traitée après évolution complète du grafcet, c'est-à-dire dans la partie postérieure du programme, celle qui positionne également les sorties liées à la partie opérative (X12 doit être stable dans ce cas). L'étape n'est active que pendant une seule scrutation, le traitement n'a donc lieu qu'une seule fois. Cette technique s'applique d'une manière universelle, quel que soit l'automate (figure A-39).

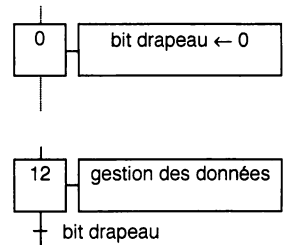
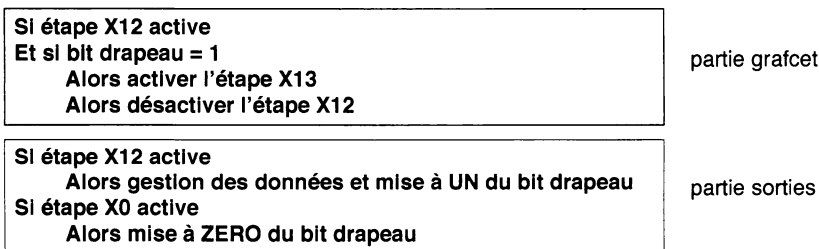


figure A-39



Scrutation (n)

Partie grafcet : lors de l'activation de X12, la transition suivante ne peut pas être franchie car le bit drapeau est = 0. Le grafcet n'évolue pas : X12 est stable.

Partie sortie : X12 étant active ⇒ gestion des données et mise à UN du bit drapeau.

Scrutation (n+1)

Partie grafcet : X12 est active et le bit drapeau est =1 ⇒ évolution du grafcet.

Partie sortie : X12 n'est pas active, les données ne sont plus traitées.

- Remarque

Le modèle Grafcet ne définit pas la représentation à ce niveau de finesse car sa vocation première est de définir uniquement les fonctions. La réceptivité [=1] notamment doit être utilisée avec précaution. On rappelle que la seule utilisation non ambiguë de cette réceptivité particulière est la resynchronisation après des séquences simultanées (convergence en ET).

d) Exemple 2 : incrémenter un compteur

Soit à positionner un compteur à une valeur déterminée (par exemple 0) sur l'étape X10 et à incrémenter ce compteur à chaque activation de l'étape X15.

- Le compteur est un mot quelconque

Attribuons à ce compteur, dans un premier temps, l'adresse d'un mot quelconque : %MW4 par exemple.

Sous peine de voir le compteur augmenter d'un pas à chaque scrutation tant que l'étape X15 est active, le programmeur doit prendre l'une des précautions suivantes.

Sur étape instable :

- insérer l'incrémentation du compteur dans le traitement du grafcet (en IL sur la figure A-40)

Sur étape stable :

- ajouter une étape supplémentaire et un bit drapeau (non illustré),
- ou former un front sur activation (ou désactivation) de X15, avant incrémentation (en LD sur la figure A-40).

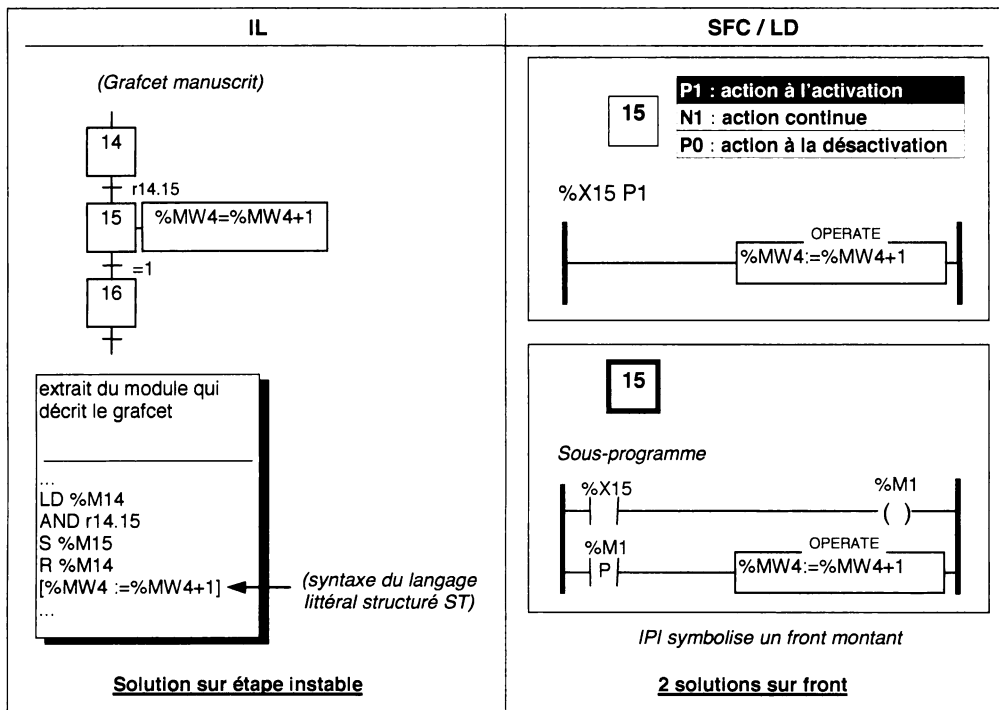


figure A-40

- Le compteur est un bloc fonctionnel

Il est également possible de matérialiser ce compteur grâce à un bloc fonctionnel. Le paramétrage se fait hors programme, dans une fenêtre de configuration (par exemple la valeur initiale du compteur, C.P., au moment de l'activation de l'entrée S). Les blocs fonctionnels sont pré-programmés afin que l'évolution des compteurs ne se fasse que sur front de la condition, ce qui simplifie la tâche du programmeur.

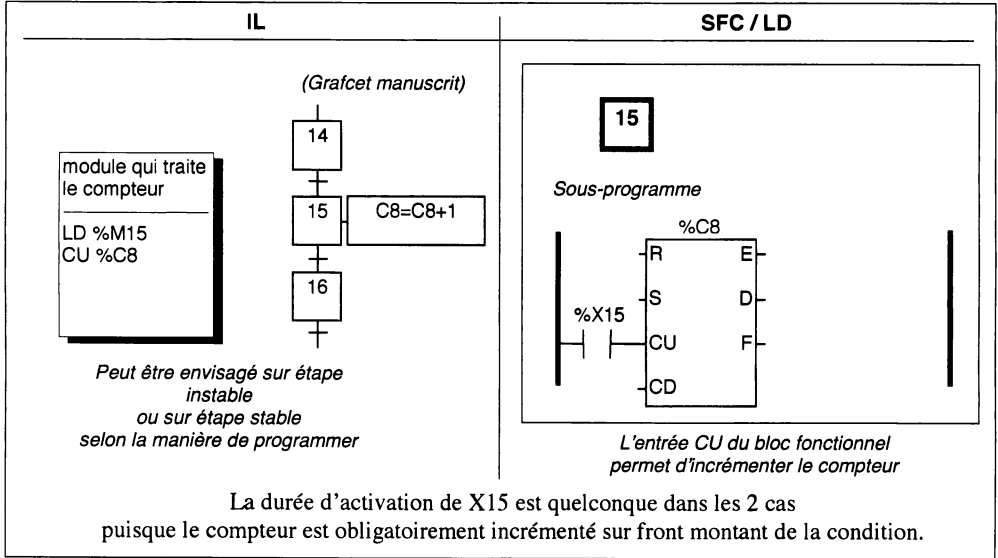


figure A-41

3 - 3. Application des fronts

Soit à allumer un voyant différent à chaque nouvelle impulsion sur un bouton poussoir. Les voyants sont au nombre de 2. La troisième impulsion a comme effet d'éteindre le dernier voyant. Ce cycle peut redémarrer à volonté.

On peut se satisfaire du grafcet au *niveau fonction* de la figure A-42-a.

Le grafcet au *niveau automate* doit cependant être aménagé. Une impulsion sur Bp est figée pour au moins une scrutation complète dans la MIE. L'étape X1 est alors activée et la transition suivante validée.

L'information Bp provenant de la MIE étant toujours =1, l'étape X2

est immédiatement activée et ainsi de suite jusqu'à épuisement de tous les modules de programme qui décrivent l'évolution du grafcet. Lorsque les équations des sorties sont traitées, il est probable que les étapes X1 et X2 ne sont plus actives : aucun voyant ne s'allume. Le grafcet n'a en réalité pas répondu au cahier des charges.

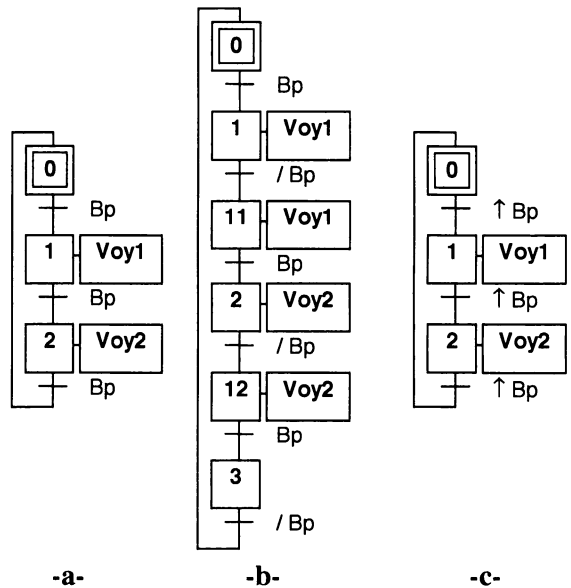


figure A-42

Les étapes doivent en fait être rendues stables. Cette situation se produit à chaque fois qu'une réceptivité est déjà vraie au moment de la validation de la transition à laquelle elle est associée et que seule l'occurrence (l'apparition ou la disparition) de cette réceptivité doit être prise en compte.

Le grafcet développé peut s'écrire sous 2 formes : soit en détaillant l'évolution des variables d'état (0 ou 1 seulement, figure A-42-b), soit en prenant en compte les états transitoires du signal (front montant ou front descendant, figure A-42-c).

4 - LES MODES DE MARCHES ET D'ARRÊTS

4 - 1. Les arrêts

a) Les arrêts pour causes normales

Ils peuvent être :

- demandés par l'opérateur : bouton d'arrêt
- émis par une autre partie commande
- générés automatiquement suite à épuisement du stock de pièces, etc.

Il sont traités au même titre que les modes de marches (voir § 4 - 2).

b) Les arrêts de sécurité

Il s'agit :

- de l'arrêt d'urgence : bouton coup de poing
- de l'ouverture du dispositif matériel de protection : capot, barrière, etc.
- des dysfonctionnements : procédures de surveillance automatiques, etc.

Les arrêts de sécurité doivent toujours être prioritaires par rapport aux marches.

L'étude des normes et réglementations liées à la sécurité dépasse largement le cadre de cet ouvrage. Une analyse de risques permet de dégager toute une palette de mesures à mettre en œuvre en fonction du niveau de dangerosité de la machine. A titre d'exemple, voici le principe de câblage qui met en œuvre un relai de sécurité ainsi que la notion de redondance (AU câblé en NO et en NF, mise en série de KAU et de KPO).

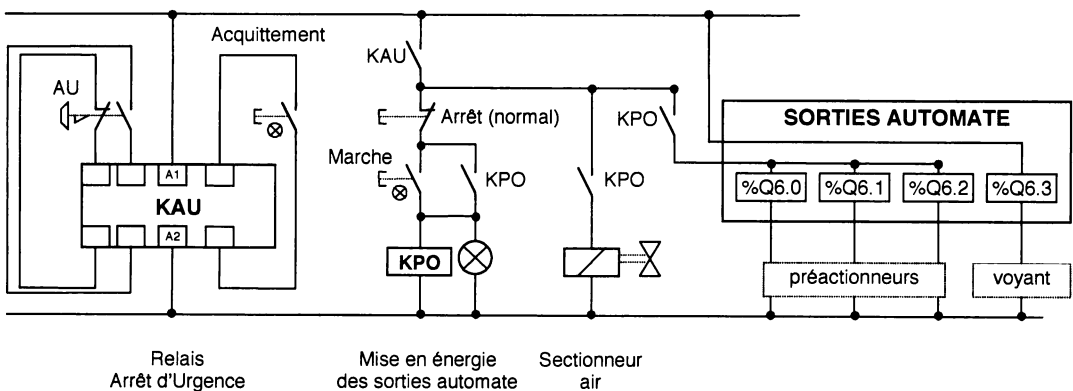


figure A-43

Le schéma de principe de la figure A-43 montre que si l'on appuie sur le bouton d'arrêt d'urgence, les sorties de l'automate ne sont plus alimentées. Les préactionneurs sont ainsi privés de leur alimentation *sans qu'aucune intervention du programme ne soit nécessaire*.

Notons que les voyants conservent leur alimentation afin de permettre la signalisation des défauts par programme.

Le rôle du programme, en cas d'arrêt d'urgence, est en outre de forcer le grafctet dans une situation conforme à la réalité, c'est-à-dire de garantir la concordance entre le modèle et le système modélisé.

4 - 2. Les marches

a) Modes de démarrage de l'automate

Le démarrage de l'automate peut avoir lieu dans des conditions très différentes :

- un nouveau programme vient d'être chargé dans l'automate : les variables prennent alors leurs valeurs initiales \Rightarrow on parle de *démarrage à froid*,
- par ailleurs, plusieurs cas de *reprise à chaud* sont possibles : soit l'automate a été arrêté puis redémarré avec le même programme, soit une coupure puis un retour du secteur se sont produits alors que l'automate était en fonctionnement, etc. Dans ces cas les variables peuvent conserver ou non leurs états, les étapes de grafctets peuvent être maintenues ou non. Les conditions de redémarrage sont paramétrables et/ou programmables selon les choix du constructeur de l'automate.

b) Exemple de modes de marches et Gemma

Après démarrage, plusieurs modes de marches peuvent en général être exécutés, soit sur demande de l'opérateur (passage en mode manuel par exemple), soit d'une manière automatique, illustrée par l'exemple de remplissage de pots de yaourts de la figure A-44.

Lorsqu'une nouvelle bande vient d'être introduite dans la machine, seul le formage doit avoir lieu dans un premier temps. En cours de production, les trois postes agissent simultanément après avance d'un pas de la bande.

Lorsque la production doit se terminer, le formage et le remplissage s'arrêtent *avant* le poste de bouchage-découpe. On a donc 3 modes de marche : la préparation, la production normale et la clôture. Chacun d'eux peut être décrit par un grafctet indépendant (figure A-45).

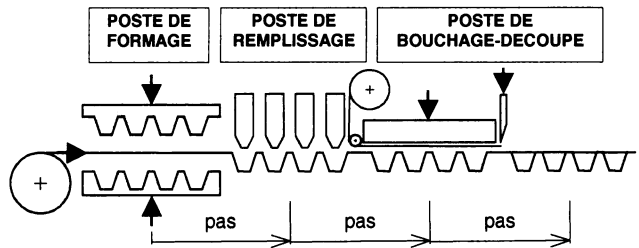


figure A-44

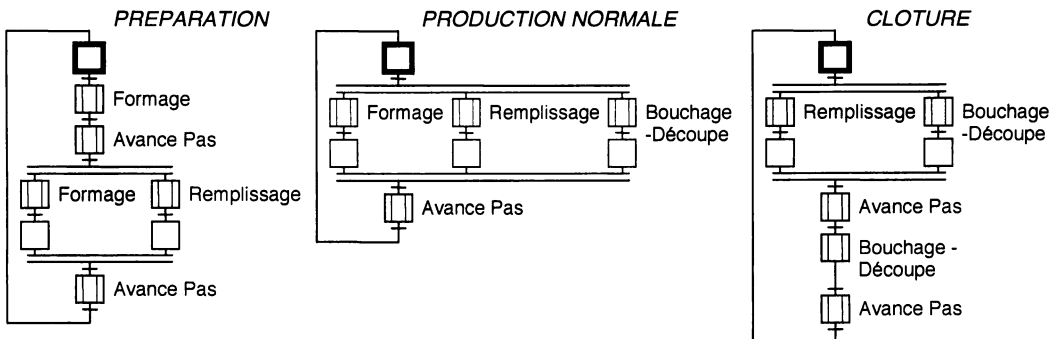
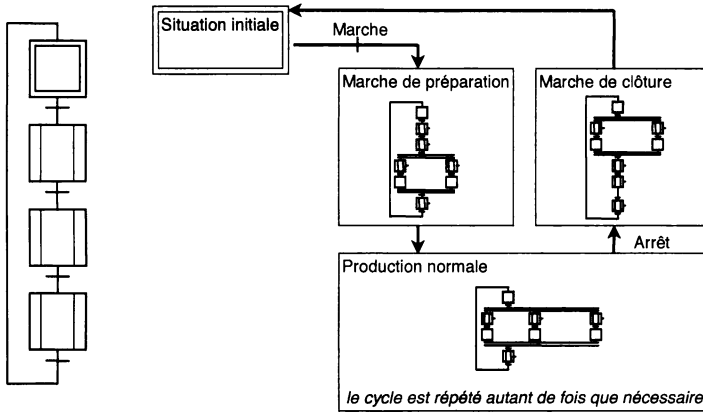


figure A-45

L'enchaînement de ces modes est organisé de la manière suivante :

Préparation → Production Normale tant que nécessaire → Clôture. Un grafcet de gestion



des modes de marches peut décrire ce cycle. On parle souvent de *grafcet de conduite* ou de *Gmma* (*grafcet de gestion des modes de marches et d'arrêts*). La figure A-46 représente ce grafcet ainsi que les correspondances avec les grafcets des différents modes. Ceux-ci sont schématisés à l'intérieur de *rectangles-états* qui forment avec les liaisons un diagramme. Il

figure A-46

n'est pas obligatoire de représenter les grafcets : un simple texte explicatif peut suffire. L'ADEPA a généralisé ce type de diagramme à tous les modes habituellement rencontrés sur les machines industrielles : le *Gemma* (*Guide d'étude des modes de marches et d'arrêts*). Sa version la plus récente est représentée figure A-47. Ce document doit être complété par le concepteur : son but est de proposer, d'une manière similaire à une checklist, un canevas pour l'étude des modes de fonctionnement d'une machine. Le vocabulaire adopté est précis et le concepteur est guidé dans l'approche du problème.

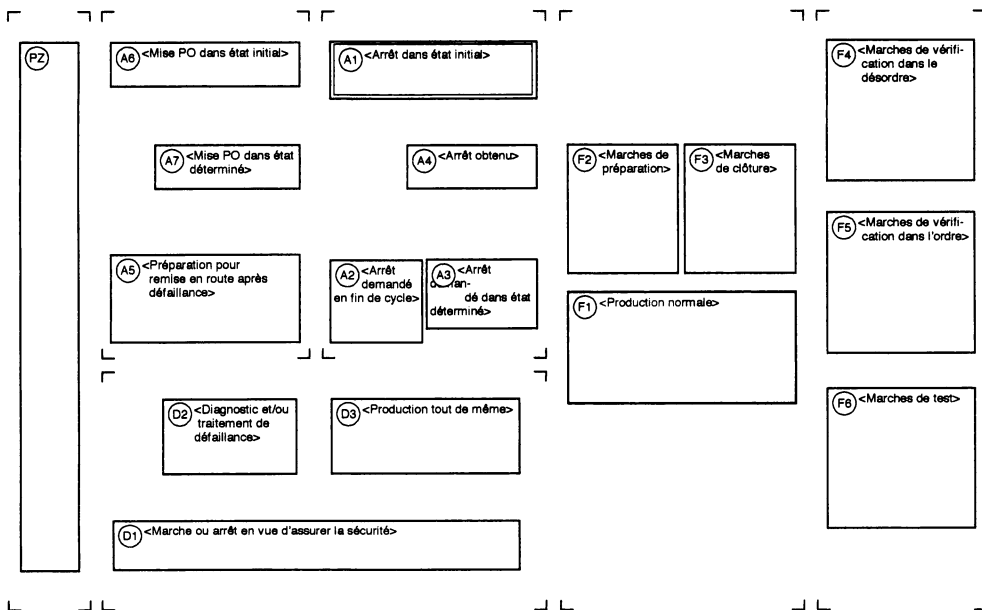


figure A-47

Les concepts du Gemma sont cependant bien moins rigoureux que ceux du Grafcet. Les liaisons entre les rectangles-états par exemple peuvent ne pas comporter de « transition ». Par ailleurs, des rectangles peuvent nécessiter soit aucune, soit plusieurs étapes. La transcription du Gemma en grafcet de conduite n'est de ce fait pas tout à fait systématique.

4 - 3. Conduite de la machine

Le Gemma ayant d'abord guidé le concepteur dans son travail, il sert ensuite à définir les interfaces utilisateurs liés aux modes de fonctionnement et éventuellement à tracer le synoptique de commande par simple analogie. Lors des phases d'exploitation et de maintenance, le Gemma facilitera ainsi la conduite de la machine. Le panneau de commande de la machine peut combiner différentes technologies : des boutons, des commutateurs, des voyants, un terminal opérateur programmable du commerce (voir la figure A-48). Ces appareils disposent d'un certain nombre de touches et de voyants ainsi que d'un afficheur textuel ou graphique. Tous ces éléments sont paramétrables en fonction de l'application.

Il est très important de prévoir toutes les signalisations utiles à la bonne prise en main de la machine. L'ajout de cette fonction ne nécessite que des frais limités et peu de programmation mais augmente considérablement la qualité générale du système. La signalisation doit cependant rester pertinente car la surabondance de voyants et autres dispositifs nuit à la perception de *chaque* signal.

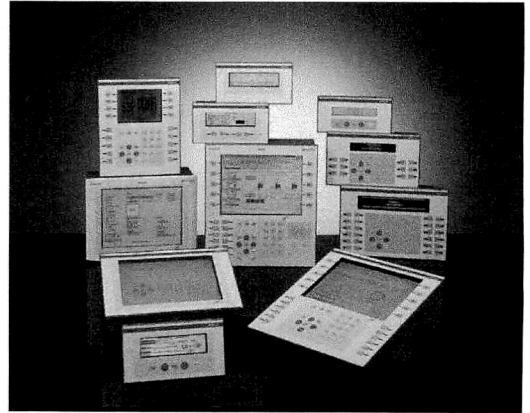


figure A-48

4 - 4. Programmes spécifiques

Quelquefois, certains modes ne sont utilisés que très épisodiquement, voire une seule fois, par exemple pendant les tests préliminaires avant mise en exploitation de la machine. Dans ces cas, plutôt que de prévoir une pléthore de modes différents dans un même programme dont la densité serait mal exploitée, on préfère réaliser des programmes spécifiques et les charger dans l'automate aux moments opportuns.

III . EXERCICES D'INITIATION

1 - CYCLE EN TRIANGLE

1 - 1. Présentation du sujet

L'étude porte sur le cycle qui permet au point P d'effectuer les mouvements ci-dessous lorsqu'un opérateur appuie sur Dcy :

- P se déplace en ligne droite de A en B,
- puis de B en C,
- puis de C en A.

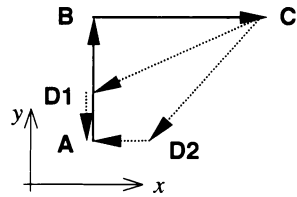


figure A-49

La dernière trajectoire *n'est pas connue a priori*. Selon les vitesses de déplacement d'une part, et les longueurs des courses d'autre part, le point P peut passer aussi bien :

- par D1 : le déplacement suivant x prend fin avant le déplacement suivant y
- que par D2 : c'est le déplacement suivant y qui prend fin avant le déplacement suivant x .

Le grafcet doit prendre en compte cette incertitude et permettre un fonctionnement correct quelles que soient les vitesses et les courses suivant les deux axes.

1 - 2. Enoncé du sujet

On propose le grafcet au niveau préactionneur ci-dessous.

- La technologie est de type électropneumatique : distributeurs bistables à commande électrique et vérins double-effet.
- Les fins de courses sont notés a_0 , a_1 pour le vérin A et b_0 , b_1 pour le vérin B.

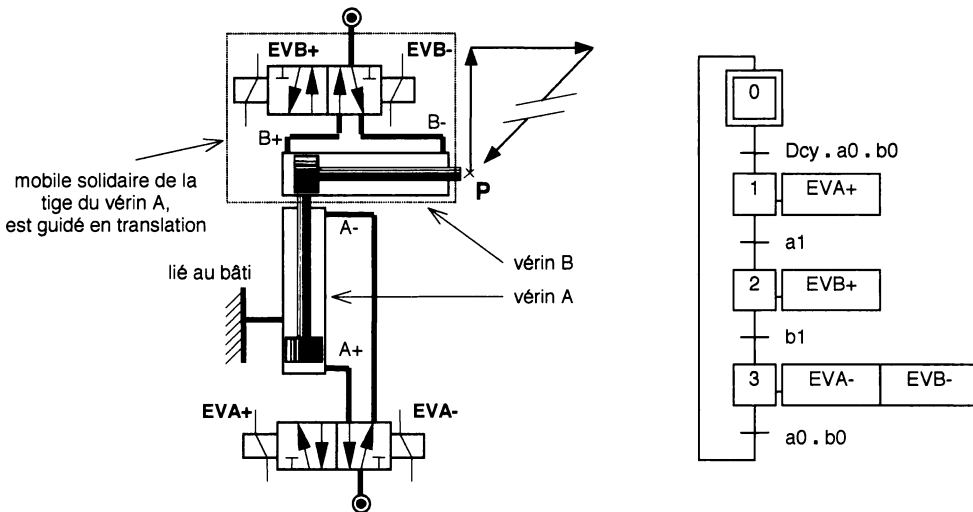


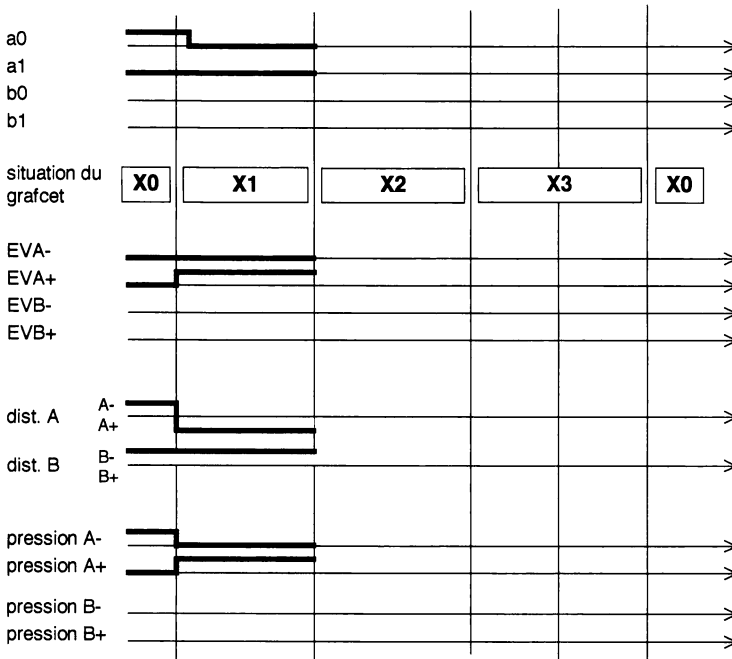
figure A-50

- Question 1.

Analyser d'une manière très précise les enchaînements des événements du point de vue des éléments de la chaîne fonctionnelle.

A cet effet, compléter le tableau et le chronogramme ci-dessous.

récept.	étapes	électrovannes	tiroirs distributeurs	vérins
	X0=1	aucune EV sous tension	positions initiales : A- et B-	A et B : tiges rentrées



figures A-51a-b

- Question 2.

Exprimer une conclusion sur la correspondance :

- a) entre la distribution de la pression dans les vérins, la position des tiroirs des EV et la présence de tension aux EV,
- b) entre la présence de tension aux EV et les mouvements des vérins.

- Question 3.

On envisage de réaliser le même cycle par une technologie électrique (schéma simplifié ci-contre). Déduire de la question 2 pour quelle raison le grafcet proposé pour la technologie électropneumatique ne peut être directement transposé en technologie électrique, en changeant simplement les noms des symboles. Ecrire le grafcet au niveau préactionneur.

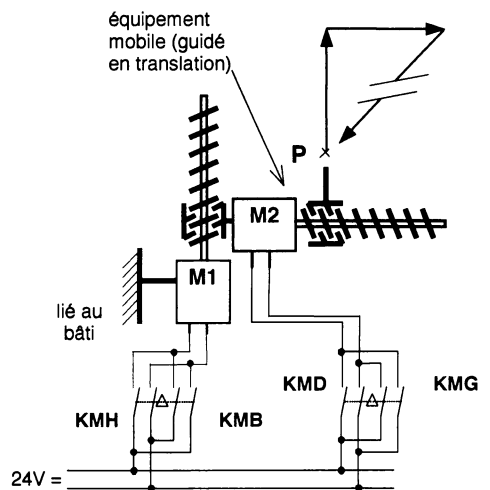
- Question 4.

Tracer un chronogramme et mettre en évidence la correspondance entre :

- la présence de tension aux contacteurs
- et la rotation des moteurs.

- Question 5.

Proposer 2 autres grafcets pour la version en technologie électrique.



Fins de course : h, b (moteur M1) et d,g (moteur M2)
figure A-52

1 - 3. Corrigé et commentaires

a) Analyse du grafcet (Q.1)

- Sous forme de tableau

réc.	étapes		électrovannes	tiroirs distributeurs	vérins
		X0=1	aucune EV sous tension	positions initiales : A- et B-	A et B : tiges rentrées
Dcy	X0=0	X1=1	EVA+ sous tension	A se déplace en A+	pression en A+ : tige de A sort
					<i>A arrive en fin de course</i>
a1	X1=0	X2=1	EVA+ n'est plus sous tension EVB+ sous tension	A reste en position A+ B se déplace en B+	A+ reste sous pression pression en B+ : tige de B sort
					<i>B arrive en fin de course</i>
b1	X2=0	X3=1	EVB+ n'est plus sous tension EVA- sous tension EVB- sous tension	A et B reviennent en position initiale	pression en A- : tige de A rentre pression en B- : tige de B rentre
a0 ou b0			EVA- reste sous tension EVB- reste sous tension		pression maintenue dans les 2 vérins côté -
<i>attente de la fin du mouvement du vérin qui n' a pas fini sa course</i>					
a0 et b0	X3=0	X0=1	plus aucune EV sous tension		pression maintenue dans les 2 vérins côté -

figure A-53

- Sous forme de chronogramme

Remarque : on suppose que la trajectoire de retour passe par D1

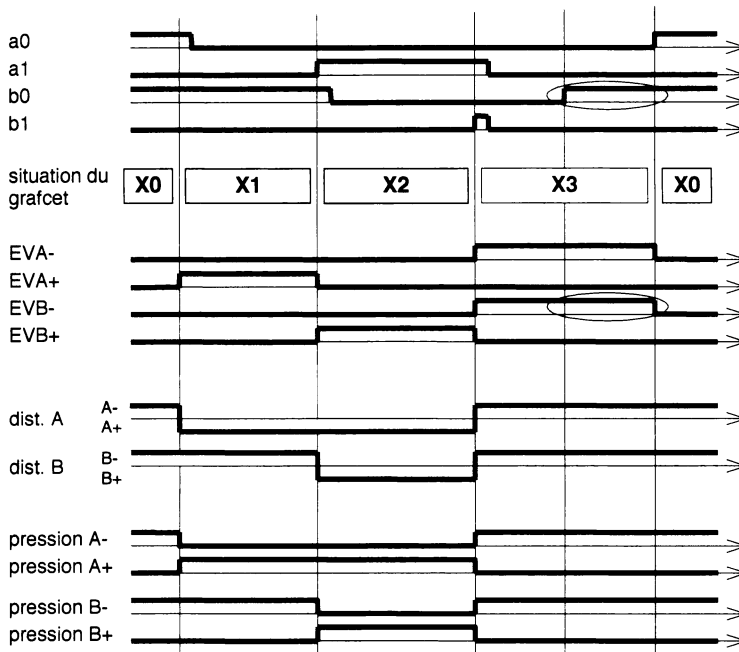


figure A-54

b) Conclusion sur l'analyse (Q.2)

a) La position du tiroir du distributeur détermine de quel côté du vérin est établie la pression (énergie de puissance).

Par ailleurs, le distributeur étant bistable, son tiroir conserve sa position malgré l'absence de tension sur l'électrovanne correspondante.

En principe, la présence de cette tension n'est donc pas nécessaire à tout instant pour maintenir le vérin dans sa position.

b) A l'inverse, on constate dans l'exemple du chronogramme, que l'électrovanne B- reste sous tension alors que le vérin qu'elle pilote est déjà arrivé en fin de course (zones entourées).

Ceci n'empêche en rien le bon fonctionnement du système, mais on s'aperçoit qu'une impulsion sur les électrovannes suffit pour agir correctement sur les vérins. La consommation en courant électrique peut être optimisée en utilisant des ordres limités dans le temps. De plus, des électrovannes à faible consommation sont disponibles sur le marché.

c) Grafcet pour la technologie électrique et commentaire (Q.3)

Alors qu'en technologie électropneumatique, il est envisageable de maintenir un ordre sur le pré-actionneur d'un actionneur qui a déjà atteint sa fin de course, il est exclu de procéder de la même façon en technologie électrique. En effet, le moteur qui agit sur un mobile doit être coupé lorsque celui-ci atteint sa fin de course. Une écriture appropriée du grafcet permet de résoudre cette question sans frais supplémentaires. La solution de la figure A-55 met deux séquences simultanées en oeuvre. Les étapes X5 et X7 sont des étapes d'attente qui permettent la resynchronisation du cycle lorsque les deux séquences X4 → X5 et X6 → X7 sont terminées. La seule activation des étapes X5 et X7 suffit à faire évoluer le grafcet : la réceptivité en aval de ces étapes ne nécessite donc aucune condition supplémentaire (elle est toujours vraie). Remarquons que certaines situations exigent un traitement particulier : limiteur de couple pour assurer la pression du mobile contre la butée de fin de course, etc.

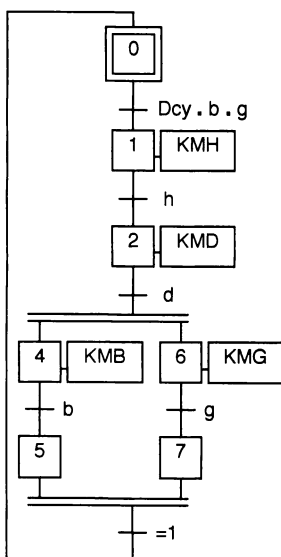


figure A-55

d) Chronogramme (Q.4)

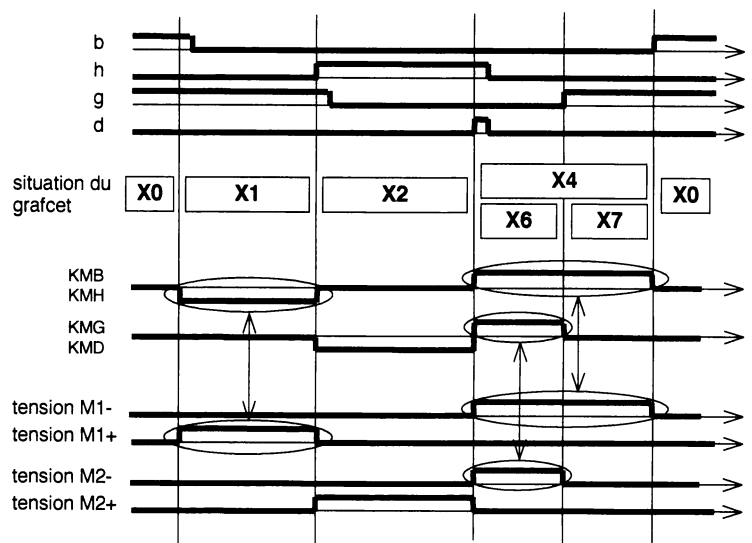


figure A-56

Ce chronogramme montre clairement la correspondance directe entre la tension au niveau des contacteurs et au niveau des moteurs, c'est-à-dire de leur rotation : KMH et M1+ pendant la durée d'activation de l'étape X1, KMB et M1- ainsi que KMG et M2- pendant la durée d'activation de l'étape X4 et X6 par exemple.

e) Variantes (Q.5)

En amont de l'étape X2, il n'y a pas de différence fondamentale entre les différentes versions. Mais une attention particulière doit être apportée dans l'écriture du grafcet en aval de l'étape X2. Ci-dessous sont représentées 2 solutions supplémentaires écrites pour la technologie électrique. Ces solutions sont également adaptables à la technologie électropneumatique.

Remarque : les 3 grafquets de *niveau préactionneur* proposés pour la technologie électrique sont en réalité équivalents aux grafquets de *niveau actionneur*, aux symboles près, y compris pour la technologie électropneumatique.

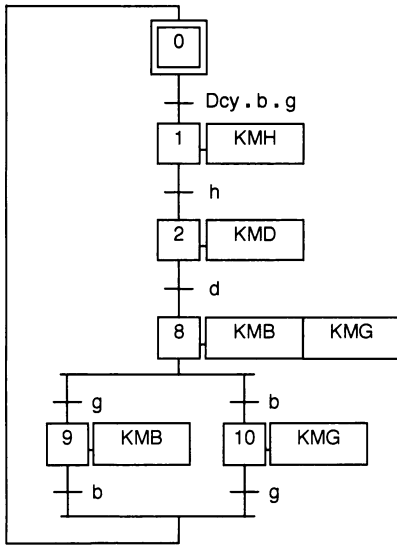


figure A-57

Tant que l'étape X8 est active, les deux contacteurs KMB et KMG sont alimentés, de même que les moteurs correspondants. Dès que *l'un* des mobiles arrive en fin de course, celui-ci s'arrête car son moteur est coupé tandis que l'autre mobile continue sa course. En aval de l'étape X8, c'est la première *ou bien* la seconde des deux séquences finales qui est exécutée, d'où la divergence en OU. Il suffit ensuite que le capteur correspondant au dernier mouvement soit présent pour que le cycle se termine.

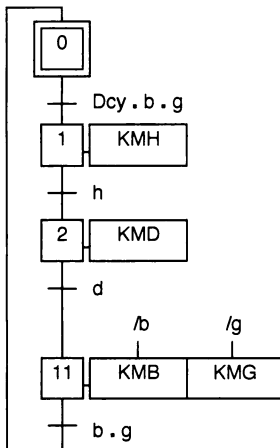


figure A-58

KMB est alimenté tant que X11 est active *et* que b est absent. L'équation s'écrit :

$$KMB = X11 . /b$$

KMG est traité de la même manière.

Dès l'activation de l'étape X11, les contacteurs KMB et KMG sont donc alimentés puisque *ni b ni g* ne sont encore présents.

Lorsque seulement l'un des fins de course b *ou* g est enclenché, le grafcet n'évolue pas. L'alimentation du contacteur en question est cependant coupée, au vu de son équation formulée telle que ci-dessus.

Le grafcet n'évolue que si les deux mouvements sont terminés : b *et* g sont alors présents.

2 - POSITIONNEMENT D'UN CHARIOT

2 - 1. Partie a : positionnement par roue crantée et détecteur TOR

a) Présentatin du sujet

Un chariot effectue un cycle aller-retour entre deux points dont l'un est fixe et l'autre variable.

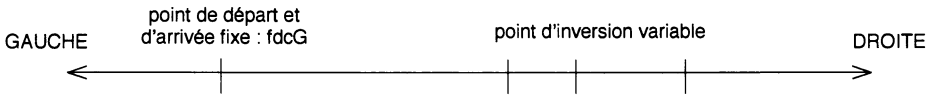


figure A-59

- Le point fixe est déterminé par la position d'un fin de course TOR noté fdcG.
- Le point d'inversion est déterminé par le nombre d'impulsions délivré par un détecteur inductif à proximité d'une roue crantée. Cette roue est solidaire de l'arbre moteur du chariot.

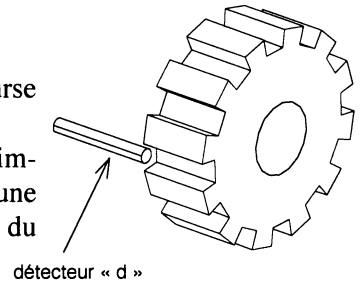


figure A-60

b) Fonctionnement du système

On propose de traiter ce problème en le décomposant en 3 grafquets distincts (hiérarchisés et numérotés comme suit) :

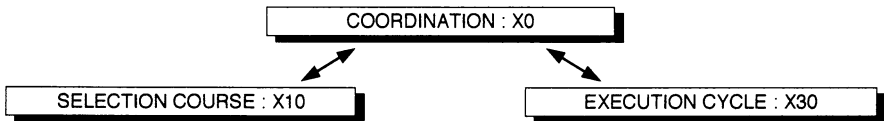


figure A-61

- Grafquet de coordination

Une impulsion sur le bouton Dcy autorise la sélection de la longueur de course. La validation de ce choix se fait par une nouvelle impulsion sur Dcy. Le cycle aller-retour est alors exécuté.

- Grafquet de sélection de course

Une impulsion sur l'un des boutons C1, C2 ou C3 provoque la mémorisation du nombre de crans à compter avant l'arrêt du chariot, ce qui détermine sa course (choisir 3 valeurs quelconques).

- Grafquet d'exécution du cycle

La longueur du trajet *aller* dépend de la sélection. Une temporisation de 0,5 secondes intervient avant l'inversion du sens de marche. Le trajet *retour* se fait jusqu'à la position initiale (fdcG).

c) Travail demandé

- Ecrire les grafquets au niveau automate (représentation symbolique) et justifier les choix.
- Compléter les grafquets en ajoutant la fonction de signalisation. Un voyant *VoyArr* est allumé en situation initiale. Un voyant *VoyMar* clignote lorsque l'opérateur doit sélectionner la course. Ce même voyant restera ensuite fixe pendant le déroulement du cycle. Un voyant *VoyOK* signale à l'opérateur que la sélection a été faite avec succès et qu'il peut demander l'exécution du cycle.

d) Corrigé et commentaires

- Coordination des tâches

- Le grafctet de coordination est de type linéaire : les 2 tâches T1 et T3 s'enchaînent l'une à la suite de l'autre.
- La condition *fdcG* ne se justifie que si l'on néglige la gestion des modes de marches et des sécurités dans l'esprit du Gemma (voir l'exercice *Poste de perçage*).
- Dans la mesure du possible, on préfère regrouper les voyants sur un même niveau hiérarchique. De même, il est souhaitable que chaque bouton, compteur, etc soit affecté à un seul niveau : ici, *Dcy* a une fonction de gestion et les boutons C_i une fonction de sélection. On évite par exemple d'utiliser *Dcy* à la fois dans le grafctet de coordination *et* dans le grafctet de sélection. Il est bien entendu qu'il ne s'agit pas là d'une règle absolue.

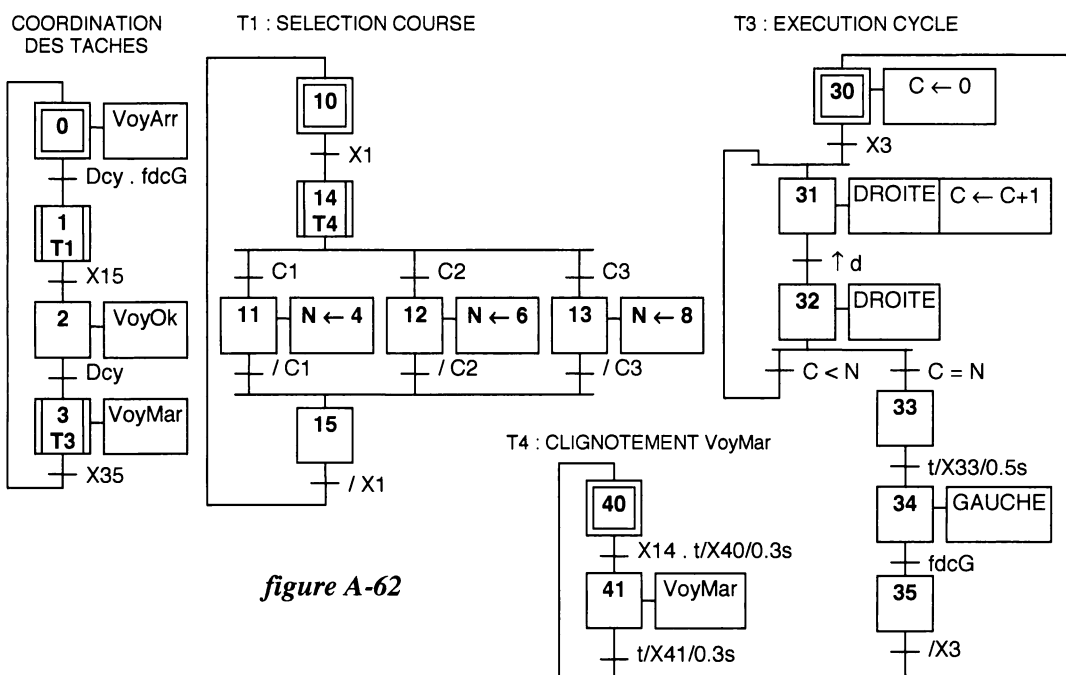


figure A-62

- Tâche de sélection

- La réceptivité X1 est la condition nécessaire et suffisante pour que le grafctet de sélection évolue. X14 reste active tant que l'opérateur n'appuie sur aucun des boutons C_i .
- Pendant tout ce temps, la sous-tâche T4 génère le clignotement du voyant *VoyMar* grâce au grafctet non synchronisé X40 (ce non-synchronisme est mis en évidence du fait de l'absence d'une étape de compte-rendu). Si X14 est désactivée dès le début de la temporisation $t/X41/0.3s$, le voyant reste allumé un court instant. Afin d'éviter cet aléa, la réceptivité en aval de X41 peut éventuellement être complétée ainsi : $[t/X41/0.3s + /X14]$. Notons que certains automates proposent des instructions ou des adresses système directement exploitables pour créer un clignotement, auquel cas ce type de grafctet peut être superflu.
- Selon le mécanisme des tâches synchronisées, X15 est une étape de compte-rendu qui permet, lorsqu'elle est activée, de faire évoluer le grafctet appelant. Dès que cette évolution est accomplie, le grafctet de tâche peut être réinitialisé ($X15 \rightarrow X10$).
- Les transitions en aval des étapes X11, X12 et X13 peuvent poser quelque problème du fait que les réceptivités associées ne sont pas évidentes :
 - il est inutile par exemple de vérifier la nouvelle valeur de N (réceptivité du type $[N=4]$),
 - il est fortement déconseillé d'employer des temporisations, leur fonction ne correspondant en rien à l'effet recherché,

- la réceptivité du type [=1] doit être utilisée avec précaution au moment du codage en langage automate (voir le problème des étapes instables au chapitre II),
- en plus d'être simple, la solution proposée n'autorise l'évolution du cycle que lorsque l'opérateur a effectivement relâché le bouton et elle s'applique aisément à toutes les techniques de programmation.

• N est une variable interne qui varie dans cet exemple de 4 à 8. Un quartet est suffisant pour mémoriser ces valeurs, mais l'entité minimum en général accessible par programme est soit l'octet, soit le mot (16 bits). Le programmeur choisira librement l'adresse de cette variable.

- Tâche d'exécution

- C est un compteur ; il est intéressant d'utiliser un bloc fonctionnel de comptage (ou une adresse spécifique) afin de bénéficier du confort de programmation qui en résulte.
- Ce grafctet présente une forme classique de comptage. Le compteur est positionné à 0 sur l'étape initiale, puis incrémenté à chaque cran, la comparaison se faisant alors par rapport au nombre de crans demandés. Une solution équivalente consiste à initialiser le compteur à la valeur N , puis à le décrémenter jusqu'à 0.

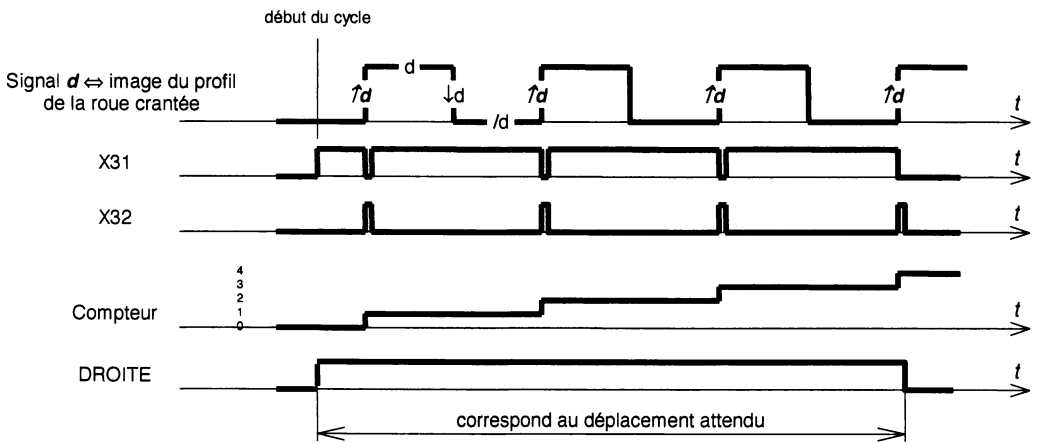


figure A-63 (prise en compte de $\uparrow d$)

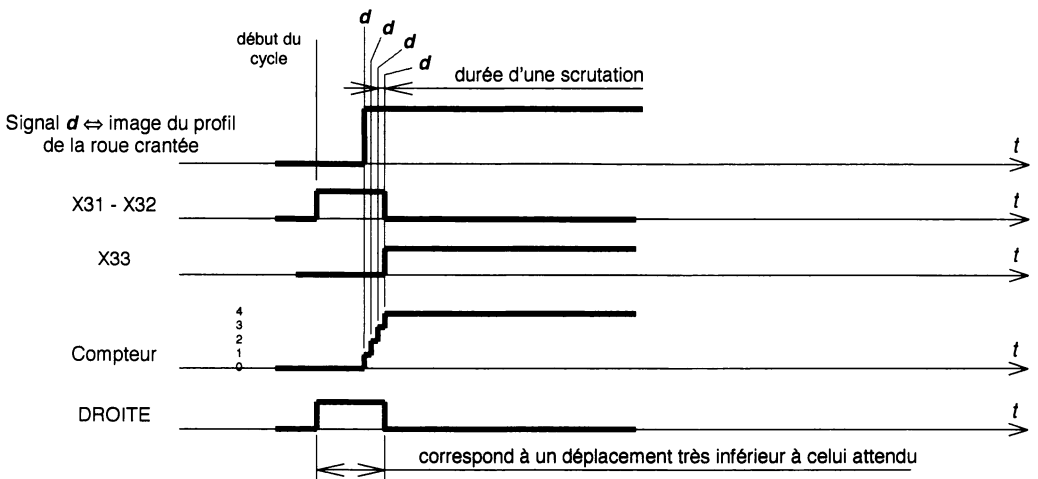


figure A-64 (prise en compte de d)

- Une attention particulière doit être portée sur les points suivants (voir le chapitre II.3) :
 - l'incréméntation doit *pouvoir* avoir lieu (stabilité ou instabilité de l'étape associée),

- l'incrémation ne doit s'effectuer *qu'une seule fois* pour chaque cran (front de l'étape associée).
- Par ailleurs, on évite en général de boucler une étape sur elle-même (solution sans X32).
- L'ordre *DROITE* doit éventuellement être répété sur l'étape X32 pour éviter de brèves interruptions de l'alimentation du moteur à chaque détection de cran.
- La première série de tracés du diagramme temporel de la figure A-63 montre le fonctionnement correct du cycle tel qu'il est prévu sur le grafcet (avec prise en compte du front de *d*). La situation au départ du tracé est : */d* et $N=4$. Sur la deuxième série de tracés (figure A-64), on a supposé l'utilisation du signal à niveau *d* en lieu et place du front de *d*. On voit que dans ce cas le cycle se terminerait *sans* que le chariot se soit déplacé de la distance voulue une fois le premier cran atteint.

- Remarque sur la pertinence de la limitation du nombre d'étapes

Chercher à réduire le nombre d'étapes des grafquets n'a de sens que si la réalisation est câblée : le coût des composants est ainsi plus faible. Le dimensionnement correct d'un automate doit au contraire permettre d'en utiliser un grand nombre. L'intérêt majeur *d'étirer un grafcet en longueur* est d'en faciliter le diagnostic en cas de défaut. Prenons par exemple l'évolution X32 → X33 → X34.

Si lors du test du programme, l'évolution du grafcet est bloquée dans la situation {X32}, il faut chercher l'erreur au niveau du comptage ; si elle est bloquée en {X33}, le défaut se situe au niveau de la temporisation.

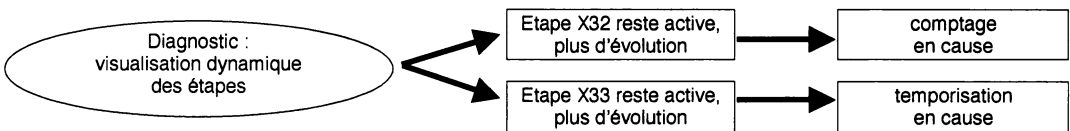


figure A-65

L'étape X33 peut être supprimée comme sur la figure A-66. Dans ce cas, l'erreur peut provenir de causes variées qu'il faut toutes envisager (comptage *et* temporisation). La recherche du défaut est moins immédiate.

En outre, la décomposition en étapes multiples a permis d'insérer facilement le clignotement du voyant sur l'étape X14 (le passage de X10 à X11 aurait pu se faire avec [X1.C1]).

De même, l'étape X2 se prête naturellement à y associer le voyant *VoyOK*.

Egalement, l'évolution de X1 directement vers X3 par la réceptivité [X15 . Dcy] empêcherait la réinitialisation du grafcet X10 dès la fin de sa tâche.

Le résultat de ces deux approches est évidemment le même du point de vue du fonctionnement. L'expérience guidera le programmeur dans le choix de l'une et/ou de l'autre selon la cas.

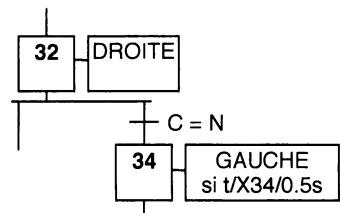


figure A-66

2 - 2. Partie b : précision du positionnement

a) Présentation du sujet

Soit le système étudié en partie *a* de cet exercice. On cherche à en déterminer la précision.

b) Travail demandé

- Calculer suivant l'axe de déplacement du chariot la distance qui correspond à l'incrément de programmation ; calculer la dispersion du positionnement :

vitesse d'avance du chariot = 0.2 m/s

diamètre roue motrice = 60 mm

temps de scrutation du programme = 50 ms

- Quelle solution, évoquée au chapitre I, permet de réduire la dispersion ?

c) Corrigé et commentaires

- La distance qui correspond à l'incrément de programmation est directement liée au nombre de crans que comporte la roue et au diamètre de la roue motrice :

$$\text{distance}_{(\text{mm})} = \{ \pi * D_{(\text{mm})} / Q_{(\text{nbre de crans})} \}$$

On peut donc atteindre un point tous les 15.7 mm.

Dispersion : à cause du traitement des entrées et des sorties via les mémoires images, le temps de réaction d'un automate est au minimum de 1 scrutation et au maximum de 2 scrutations. En 1 scrutation, le chariot se déplace de $200 * 50 / 1000$ mm. Cette distance de 10 mm correspond à la dispersion (position maxi - position mini). Pour $N=1$, le point d'arrêt réel se trouve aléatoirement entre la coordonnée 15.7 et la coordonnée 25.7 mm.

Cette dispersion peut être acceptable pour un transbordeur dont le positionnement final est ajusté par un dispositif supplémentaire. Par contre, elle est bien trop grande pour la plupart des mécanismes.

- Une première solution pour réduire considérablement cet écart est de s'affranchir du temps de scrutation de l'automate en déclenchant un traitement prioritaire dès l'apparition de la condition d'arrêt du chariot et dont le but est de provoquer instantanément l'arrêt du chariot. Le temps de réaction étant alors négligeable, la dispersion sera très faible. Pour que ce type de traitement puisse être validé, les automates imposent en général que le détecteur soit câblé à une adresse bien définie ou bien qu'on le raccorde à une carte d'interruption spécialisée.

2 - 3. Partie c : positionnement par codeur incrémental

a) Présentation du sujet

Afin d'augmenter la flexibilité du chariot, le nombre de points par tour de roue doit être plus élevé. A cet effet, la roue crantée, qui génère 12 fronts montants par tour, est remplacée par un codeur incrémental, dont le principe est identique, mais qui délivre 1000 impulsions par tour.

PRINCIPE DU CODEUR INCREMENTAL ROTATIF :

Une matrice de capteurs optoélectronique est illuminée par une source lumineuse à LED infrarouge au travers d'un disque marqué angulairement de zones opaques et de zones transparentes. La rotation du disque génère ainsi une suite de signaux, convertis en impulsions logiques.

Un tel codeur peut être assemblé directement sur l'arbre d'un moteur ou être entraîné grâce à un système pignon-crémaillère.

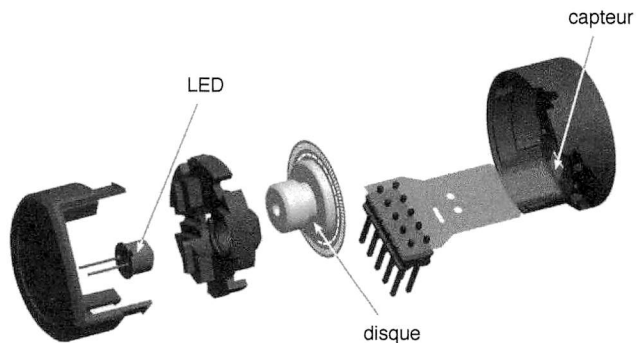


figure A-67 (doc. Portescap)

b) Travail demandé

- Calculer suivant l'axe de déplacement du chariot la distance qui correspond à l'incrément de programmation.
- Formuler des commentaires sur la dispersion.
- Calculer les nouvelles valeurs de N relatives aux boutons C1, C2 et C3 si on veut obtenir les mêmes déplacements qu'en partie a.
- Recenser l'ensemble des contraintes que ce dispositif requiert. En chiffrer les critères pour le cas des hypothèses de la partie b.

c) Corrigé et commentaires

- La relation est :

$$\text{distance}_{(mm)} = \{ \pi * D_{(mm)} / P_{(nbre\ pts\ du\ codeur)} \}$$

Cette fois, on peut atteindre un point tous les 0.19 mm. La précision est env. 83 fois meilleure qu'avec la roue crantée.

- Sans la mise en oeuvre du traitement prioritaire, la dispersion reste pourtant de 10 mm car l'utilisation du codeur n'améliore en rien le temps de réaction de l'automate. Elle peut être rendue négligeable grâce au traitement prioritaire. Dans tous les cas, le codeur permet une plus grande flexibilité.

- Solution de la roue crantée, pour N=4 : $4 * 15.7\text{ mm} = 62.8\text{ mm}$
 Solution du codeur pour le même déplacement : $62.8 / 0.19\text{ mm} = 330\text{ points}$

C1	N = 4	Nombre de points (consigne) = 330
C2	N = 6	Nombre de points (consigne) = 496
C3	N = 8	Nombre de points (consigne) = 661

- Un premier critère est la fréquence du train d'impulsion. Cette fréquence doit rester compatible avec celle de la scrutation du programme pour garantir la prise en compte de *toutes* les impulsions.

Le temps d'une scrutation étant de 50 ms, sa fréquence est de $1 / 0.05$; c'est-à-dire : **20 Hz**

La vitesse de rotation du codeur est : $S_{(tr/s)} = V_{(mm/s)} / (\pi * D_{(mm)})$

Le nombre d'impulsions par seconde est de $(S_{(tr/s)} * P_{(nbre\ points\ du\ codeur)})$

d'où la fréquence du train d'impulsions = **1061 Hz**

Conclusion : seule 1 impulsion sur 53 est prise en compte par le programme. La solution est inacceptable.

Solution : Le codeur doit être relié à une carte (ou une entrée) spécialisée dite de *comptage rapide* dont les fréquences de traitement peuvent être variables selon le type (500 Hz, 10000 Hz, 250000 Hz par exemple). Un processeur indépendant gère ce comptage et dialogue avec l'unité centrale pour prendre en compte la consigne à atteindre et pour traiter le résultat en fonction du programme.

Autrement dit, ce n'est plus le programme de l'unité centrale qui gère le comptage : le grafcet doit être modifié en conséquence comme l'indique la figure A-68. Le signe > de la réceptivité en aval de l'étape X31 vient du fait qu'il est statistiquement improbable que l'égalité stricte puisse être observée par l'unité centrale, donc par le grafcet. Dans le cas où l'on met en oeuvre un traitement prioritaire pour l'arrêt du chariot lorsque la valeur courante est égale à la consigne, il faut veiller à ne pas maintenir la sortie *DROITE* plus longtemps, car l'évolution du grafcet sera en retard par rapport à cet instant précis. On conseille au lecteur de tracer un chronogramme pour bien assimiler ce mécanisme.

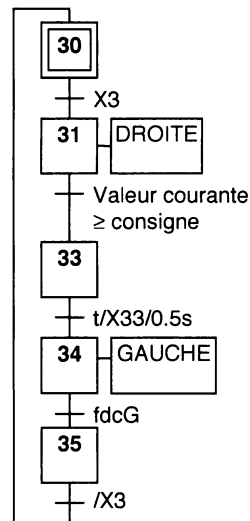


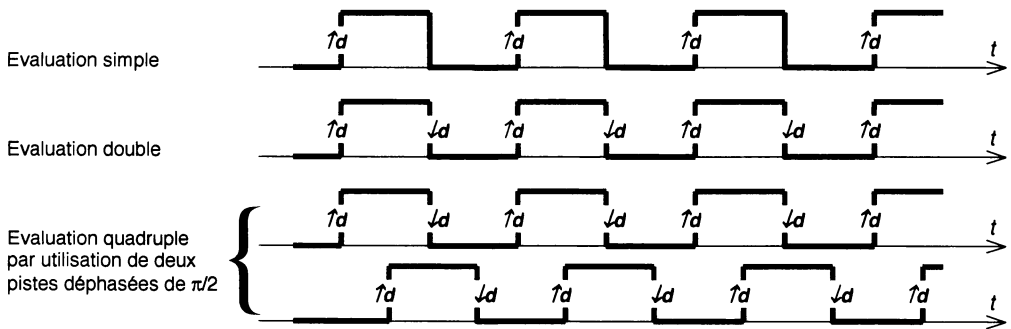
figure A-68

— Un deuxième critère est la taille du registre de comptage. En effet, le comptage des impulsions se fait dans un registre particulier dont le format est déterminé par le constructeur (16 bits, 32 bits). Plus la fréquence est élevée, plus le registre arrive rapidement à saturation, ce qui a une incidence sur la course utile du chariot. Si le registre est par exemple de 16 bits et si on utilise seulement les valeurs positives (1 bit étant cependant réservé pour le signe), la limite avant saturation est de :

2^{15} , à savoir 32768 points, ce qui correspond à une course de :

$$\text{course utile}_{(mm)} = \pi * D_{(mm)} * 32768_{(pts)} / P_{(nbre\ pts\ du\ codeur)} = \mathbf{6173\ mm}$$

— La précision du codeur peut être doublée en évaluant également les fronts descendants du train d'impulsions. La course utile sera évidemment réduite de moitié et la fréquence doublée. La note technique du codeur montre qu'il peut exister une seconde piste : elle est en général déphasée de $\pi/2$ par rapport à la première. L'évaluation de tous les fronts des deux pistes multiplie ainsi la résolution du codeur par 4 (figure A-69).



Note : l'échelle de temps est différente de celle de la figure A-63

figure A-69

— Le déphasage des deux pistes a une seconde utilité : la discrimination du sens de rotation du codeur. Selon l'ordre d'arrivée des fronts, l'une des deux fonctions *comptage* ou *décomptage* est alors automatiquement validée.

— Un dernier critère concerne la prise de référence de l'axe numérique ainsi réalisé. A cet effet, la troisième piste du codeur définit une position unique : le *top zéro*. Ce signal apparaît à chaque tour du codeur aussi en faut-il déterminer celui qui doit effectivement initialiser l'axe (c'est-à-dire positionner le compteur à la valeur initiale : 0 ou une valeur appropriée) : un détecteur TOR permet cette sélection (figure A-70)

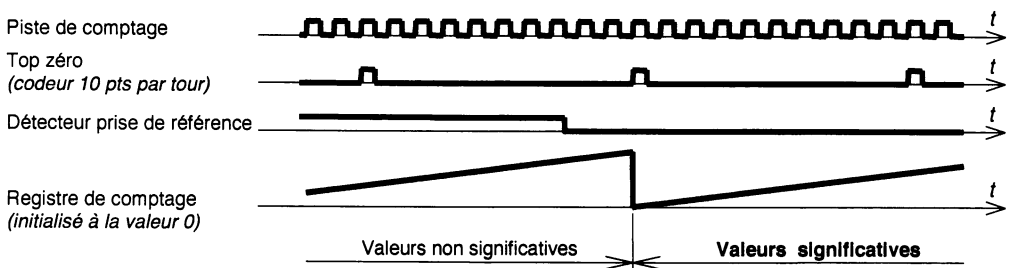


figure A-70

• Conclusion : Le choix du nombre de points d'un codeur, du facteur d'évaluation et de la fréquence de comptage maxi doit être compatible avec la précision recherchée, la taille du registre de comptage, la course du mobile et sa vitesse d'avance.

- **Remarques :**

- La configuration des paramètres ci-dessus se fait selon le type d'automate par switches et/ou par logiciel.
- Les codeurs existent également en technologie analogique (mesure d'une tension au lieu d'impulsions), peuvent être de type absolu (à chaque position angulaire du codeur correspond une valeur déterminée) et peuvent se présenter sous forme linéaire (règles).
- La gestion des informations et des actionneurs peut également être faite d'une manière plus complète par des cartes d'axes.

3 - POSTE DE PERCAGE

3 - 1. Présentation du sujet

Il s'agit d'un poste de perçage automatique qui permet de traiter 6 types de pièces sans qu'aucun réglage sur la partie opérative ne soit nécessaire lorsqu'on change de série. La machine est représentée schématiquement ci-dessous.

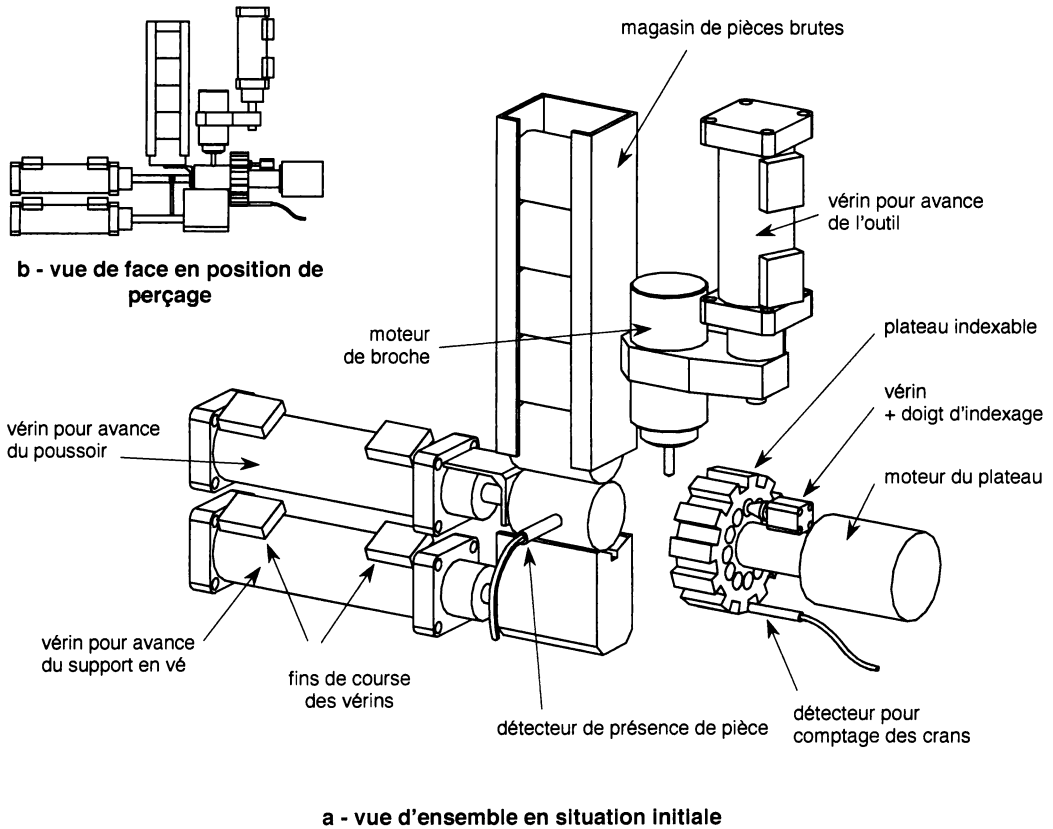


figure A-71

Les différents types de pièces à traiter appartiennent à la même famille : les trous à percer sont tous du même diamètre, de la même profondeur et disposés dans un même plan transversal, seul leur nombre varie d'un type à l'autre. Ce nombre peut être de 1, 2, 3, 4, 6 et 12. Dans le cas où 2 trous doivent être percés, ils sont espacés de 180°. 3 trous sont espacés de 2 fois 120°, 4 trous sont à 90°, 6 trous à 60° et 12 trous à 30°. De ce fait, l'indexage du plateau se fait par une couronne qui est crantée tous les 30°, l'incrément α entre deux trous étant un multiple de 30°.

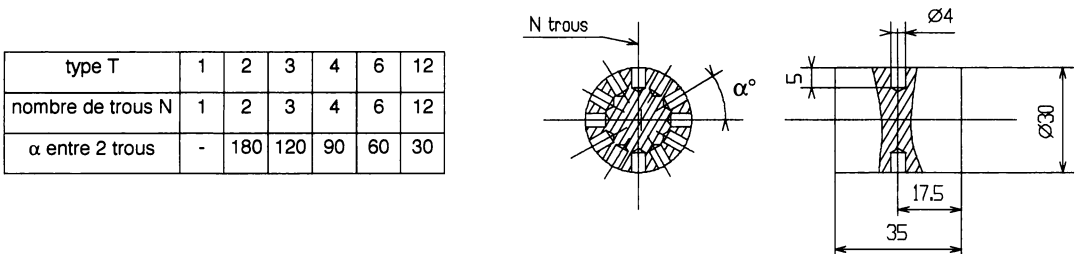


figure A-72 : les 6 types de pièces à percer

3 - 2. Fonctionnement du système

La figure A-71-a montre le système en situation initiale : toutes les tiges de vérin sont rentrées, les moteurs à l'arrêt et une pièce peut éventuellement être présente sur le support en vé. Le plateau est dans une position correcte.

- Le cycle démarre par l'alimentation de la pièce : le poussoir et le support en vé avancent simultanément jusqu'en fin de course. Le poussoir est muni d'un accessoire qui maintient en place les éventuelles pièces présentes dans la goulotte.
- Un premier perçage a lieu : la broche se met en rotation en même temps que l'indexeur se met en place, l'outil avance ensuite jusqu'en fin de course puis recule immédiatement en position initiale.
- Si un autre trou doit être percé, la broche reste en rotation, l'indexeur se rétracte, le plateau tourne de l'angle α nécessaire, l'indexeur se remet en position de verrouillage et un nouveau perçage a lieu.
- Si tous les trous ont été percés, l'indexeur ainsi que le support en vé reculent. La pièce reste bloquée entre le poussoir et le plateau. Dès que le support est rentré, le poussoir recule et la pièce tombe dans une goulotte d'évacuation (non représentée sur les figures).
- Un cycle pour une nouvelle pièce peut commencer. La broche s'arrête lorsque toutes les pièces de la série ont été percées.

3 - 3. Pupitre opérateur

Le pilotage de la machine se fait à l'aide du pupitre représenté figure A-73.

Il est complété par une partie gestion des énergies qui commande la mise en marche et la mise à l'arrêt des parties commande et opérative : boutons mPC, aPC, mPO et aPO et voyants PCenService et POenService.

Acqu. défaut est un bouton lumineux. Sa signalisation est assurée par câblage.

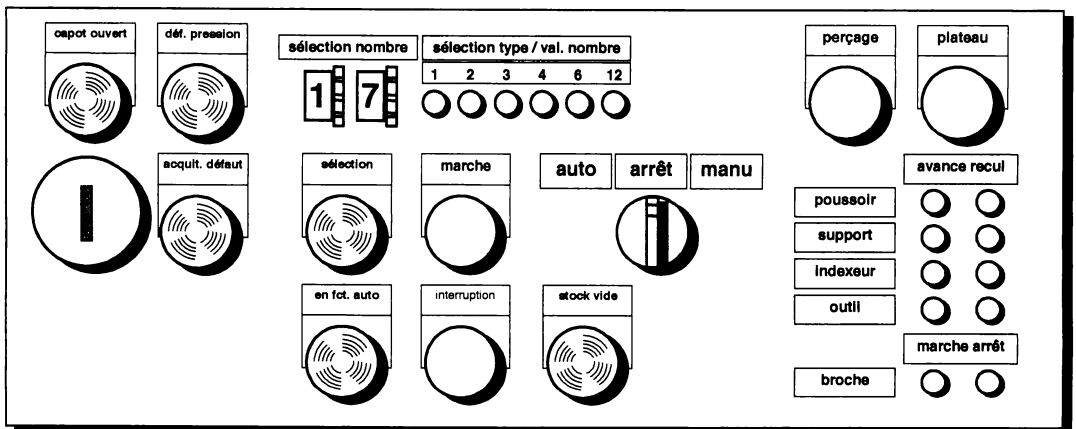


figure A-73

3 - 4. Commande du système

Le sélecteur à 3 positions *auto-arrêt-manu* permet de sélectionner le mode de marche.

a) Calque marche normale

- En mode *auto*, l'opérateur appuie sur le bouton poussoir *marche* pour faire les sélections suivantes :

- le nombre de pièces à réaliser dans la série (cette sélection se fait par les 2 roues codeuses, le nombre peut varier de 0 à 99),
- le type de pièce à réaliser (6 boutons poussoirs sont prévus à cet effet ; ils sont notés *t1*, *t2*, *t3*, *t4*, *t6*, *t12* selon le type).

Remarque : au moment de la sélection du type, le nombre de pièces à réaliser qui est mémorisé est celui qui est affiché sur les roues codeuses.

- Lorsque l'opérateur appuie de nouveau sur le bouton *marche*, le cycle de production est démarré et se termine automatiquement lorsque le nombre total de pièces demandé est atteint. A tout instant, l'opérateur peut sélectionner la position *arrêt* du sélecteur de mode. Dans ce cas, la machine s'arrête lorsque le cycle qui est en cours est terminé
- Vérification de la présence pièce : pour éviter d'endommager la machine, le cycle ne doit démarrer que si une pièce est présente. Dans le cas où le magasin serait vide, l'opérateur installe au moins 1 pièce brute et doit ensuite valider le redémarrage par une action positive sur le bouton *marche*.

Note : il est conseillé de traiter dès maintenant les questions 1, 2 et 3.

b) Calque initialisation PO (ou mise en référence)

Au démarrage, ou en cas d'arrêt par défaillance, ou après réglage, la partie opérative ne se trouve pas forcément en situation initiale. Lorsqu'on demande le (re)démarrage, sa mise en situation initiale doit se faire de manière automatique, dans de bonnes conditions de sécurité.

c) Calque défaillances

Le poste est dangereux du fait des mouvements des vérins et de la rotation de la broche et du plateau : un capot de protection - non représenté sur les figures - agit sur un interrupteur de sécurité. Par ailleurs un bouton coup de poing *arrêt d'urgence* permet à tout instant d'arrêter immédiatement la machine. Le fonctionnement ne doit être possible que si le capot est fermé et le bouton AU déverrouillé. En cas de redémarrage, l'opérateur doit en outre valider son intention en appuyant sur le bouton *marche*.

d) Calque réglages

- Vérins et broche en mode manu

Afin de régler les positions exactes des fins de courses, notamment lors du remplacement du foret, l'opérateur accède individuellement à chaque mouvement de vérin par un bouton poussoir. La broche est mise en rotation et arrêtée de la même manière.

- Plateau en mode manu

Le plateau est mis en rotation par une impulsion sur le bouton poussoir *plateau* et s'arrête dès l'arrivée du prochain signal au niveau du détecteur de crans.

Afin de tester le bon fonctionnement de l'usinage, le bouton *perçage* permet de lancer un cycle unique de perçage.

e) Interruption du cycle

Lorsque le cycle est *en cours d'évolution*, l'appui sur le bouton *interruption* provoque l'arrêt immédiat du cycle. Les vérins terminent néanmoins les courses qui ont été entamées. Le système reste en l'état jusqu'à ce que l'opérateur appuie sur le bouton *marche*, auquel cas le cycle se poursuit là où il a été interrompu. Pour quitter définitivement la production,

on bascule le sélecteur de mode en position *arrêt*. Dans ce cas, l'état de la partie opérative n'est pas modifié.

f) Gestion du nombre de pièces

Le compteur qui est associé au nombre de pièces à réaliser ne doit pas être perturbé par les différents arrêts intermédiaires : après les redémarrages, la production se poursuit pour obtenir le nombre total de pièces demandé au départ.

g) Signalisation

Les voyants proposés sur le pupitre doivent signaler les états successifs du système.

3 - 5. Enoncé du sujet

a) Première partie

Question 1. Faire l'inventaire des variables (paramètres et compteurs).

Question 2. Tracer le Gemma du calque *marche normale*. En déduire, pour ce calque seulement, le grafctet de conduite (gestion des modes de marches et d'arrêts).

Question 3. Ecrire le grafctet de coordination des tâches du cycle de production ainsi que les grafctets de toutes les tâches : *niveau effecteur, point de vue de l'observateur (voir la figure A-33, page 34)*. Compléter les dialogues sur le grafctet de conduite.

Veiller à respecter la recommandation suivante : chaque bouton, chaque voyant, chaque compteur, etc. ne doit être géré que sur un seul niveau hiérarchique.

b) Deuxième partie

Question 4. Choisir la technologie des vérins et de leur préactionneurs en réfléchissant au comportement du système dans les cas suivants :

- ouverture inopinée du capot,
- arrêt d'urgence,
- chute de pression.

Question 5. Compléter le Gemma par les autres modes. Compléter le grafctet de conduite, tracer les grafctets supplémentaires (surveillance et autres modes).

Question 6. Tracer un diagramme qui résume la structure hiérarchisée des grafctets.

Question 7. Au cas où le foret se brise en cours d'usinage, on estime que le mouvement du vérin qui provoque son avance est ralenti voire bloqué. Proposer une solution simple pour générer un arrêt approprié de la machine.

3 - 6. Corrigé et commentaires

a) Inventaire des variables (Q.1)

Les variables définissent d'une part le nombre de pièces à réaliser dans la série et d'autre part les caractéristiques de chaque type de pièce, à savoir le nombre de trous, ce qui implique également l'angle entre les trous. On a donc 3 niveaux de paramètres et de compteurs. Les symboles retenus sont composés des lettres C et N (respectivement pour compteur et nombre) et des indices P, T et C (respectivement pour pièces, trous et crans). Afin de structurer correctement l'analyse, chaque compteur doit être géré sur un seul niveau hiérarchique : le nombre de pièces au niveau *conduite*,

le nombre de trous au niveau commande de la tâche *production* et l'incrément d'angle au niveau de la tâche *positionnement plateau*. Il en résulte le tableau suivant.

désignation	variables		niveau hiérarchique
	paramètre	compteur	
nombre de pièces de la série	Np	Cp	conduite
nombre de trous	Nt	Ct	tâche <i>production</i>
nombre de crans (incréments)	Nc	Cc	tâche <i>positionnement</i>

figure A-74

b) Mode : marche normale (Q.2)

- **Gemma**

Le Gemma proposé en figure A-75 présente certaines particularités.

- Le Gemma peut être tracé sur feuille libre. La grille proposée par l'ADEPA sert alors de référence, en particulier pour l'identification des rectangles-états (F1, F2, etc.).
- De nombreuses liaisons interconnectent les rectangles, notamment pour arrêter la machine depuis les modes F2 et A2. Elles peuvent paraître surabondantes *a priori*, mais elles ne nécessitent que très peu de programmation supplémentaire et surtout, confèrent au système une bien plus grande souplesse d'emploi, donc une meilleure qualité d'ensemble.

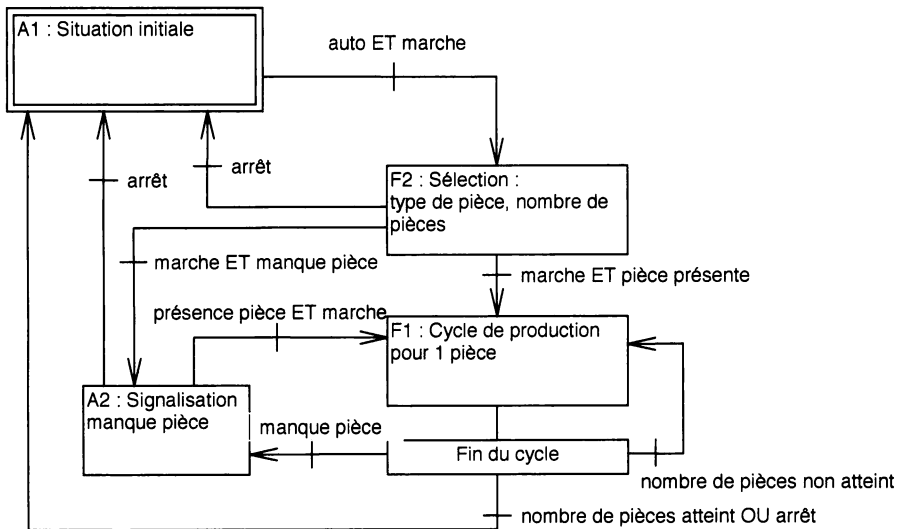


figure A-75

- Le rectangle *fin du cycle* permet d'expliciter très clairement le fonctionnement pour des cycles répétés de manière automatique. Ce rectangle ne peut en effet être comparé à aucun mode proposé par la grille habituelle : ce n'est ni un arrêt, ni un fonctionnement, ni une défaillance. La fonction qu'il réalise peut aussi être traduite directement par grafcet à l'intérieur du rectangle F1.
- Remarquons encore que l'écriture des conditions logiques n'utilise pas le formalisme des automatismes : [ET] ne s'écrit pas [.] , [OU] ne s'écrit pas [+] et [pas] ne s'écrit pas [/] car ce document est destiné à toutes les personnes en contact avec la machine. Pour la même raison, ces conditions ne sont pas exprimées de manière exhaustive. Elles ne font apparaître que les paramètres les plus immédiats, sans souci de gestion de priorité par exemple (elles doivent être comparées avec les réceptivités équivalentes du grafcet associé, figure A-76).

- Grafcet de conduite

Ce grafcet est la transcription du Gemma. Les réceptivités sont complétées de façon à gérer les sélections de séquences sans ambiguïté.

La transcription ne peut pas se faire selon une méthode systématique. Par exemple, les 3 liaisons qui quittent F2 sont de même nature sur le Gemma mais sont aménagées différemment sur le grafcet pour permettre la sortie de ce mode sans obliger l'opérateur à faire une sélection. Cette possibilité traduit le souci de produire une machine proche de la *commande intuitive*, principe d'ailleurs prévu par le code du travail, article R.233-89.

Le grafcet de conduite gère les tâches *sélection* et *production* ainsi que les voyants liés à la conduite du système et le compteur du nombre de pièces. Et c'est à ce niveau que le pilotage de la rotation de la broche s'avère le plus simple : elle est équivalente à la tâche *production* du point de vue temporel.

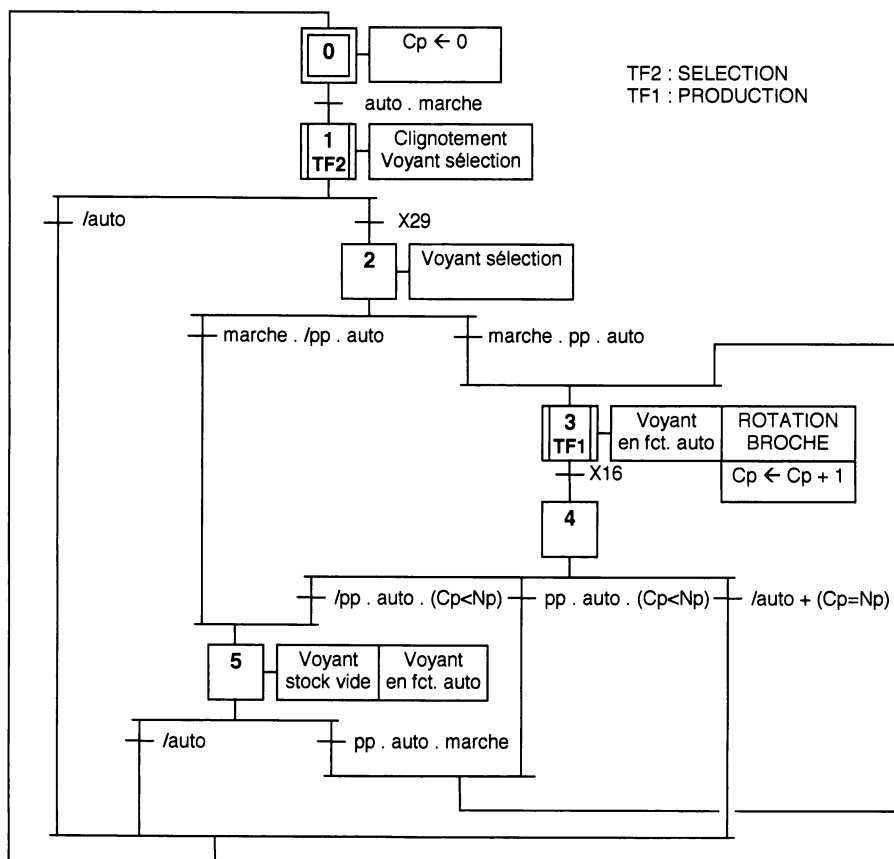


figure A-76

c) Grafquets : marche normale (Q.3)

- Tâche sélection TF2

Le grafcet de la figure A-77 est le résultat de l'analyse à la fois des variables utiles et de leur mode de mémorisation.

Au sujet de l'ordre de l'étape X28 : les roues codeuses délivrent en général un code BCD sur 4 bornes pour chaque digit. Quatre entrées TOR par digit sont donc nécessaires sur l'automate (pour lire les chiffres 0 à 9) ainsi qu'un programme de décodage.

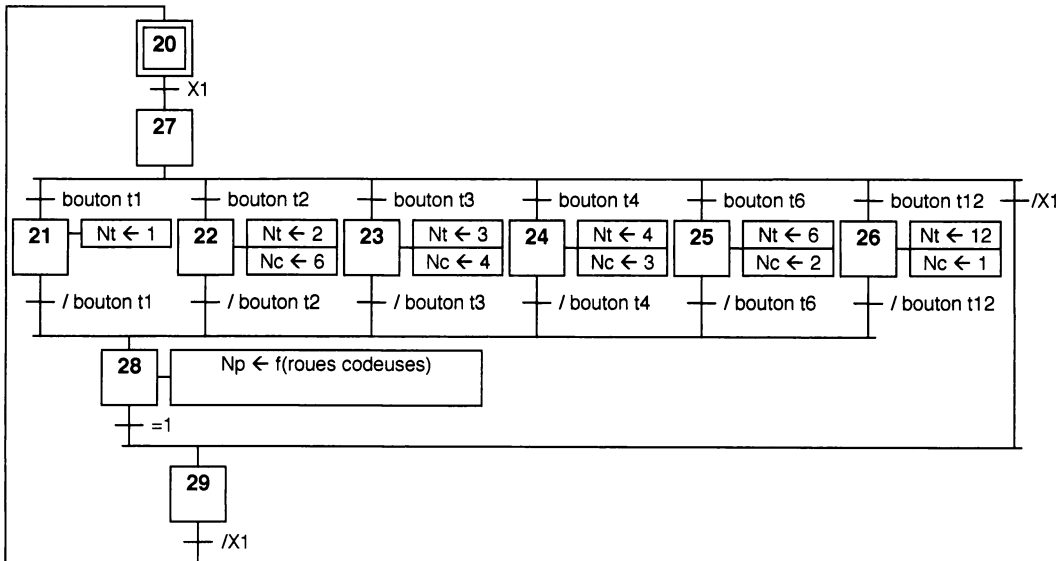


figure A-77

- Tâche production TF1

Ce grafcet (figure A-78) est chargé de la coordination des tâches de niveau hiérarchique inférieur : l'alimentation, l'évacuation, la trajectoire de l'outil et le positionnement du plateau. Notons que les questions déjà traitées (une pièce est-elle présente ? la partie opérative est-elle initialisée ? etc.) ne sont plus à l'ordre du jour. C'est justement pour cette raison que la décomposition en grafquets hiérarchisés est intéressante : l'analyse porte à des moments différents sur des problèmes restreints et isolés.

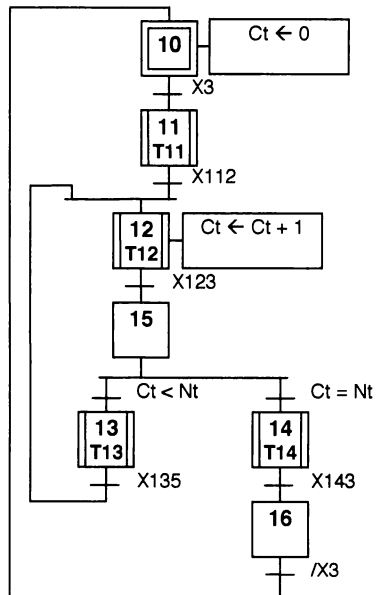
Les comptages effectués par les grafquets X10 et X130 appellent les mêmes commentaires que le corrigé de l'exercice *Positionnement d'un chariot*.

- Tâches de niveau inférieur

Voir la figure A-79, page suivante.

d) Technologies des préactionneurs et des actionneurs (Q.4)

La sécurité doit obligatoirement être traitée au premier niveau par le choix judicieux des préactionneurs, des actionneurs et du schéma de câblage. La programmation ne permet quant à elle de ne gérer que les situations de repli et de redémarrage. Dans le cas habituel d'un pilotage par automate programmable, les préactionneurs sont à commande électrique (même type et même niveau d'énergie que les sorties). Le câblage est réalisé de sorte qu'en cas de danger - qui survient ici soit par ouverture du capot, soit par appui sur le bouton AU, soit par une chute de pression -



- T11 : ALIMENTATION
- T12 : PERCAGE
- T13 : POSITIONNEMENT PLATEAU
- T14 : EVACUATION

figure A-78

l'alimentation en courant des préactionneurs est interrompue. S'ils sont monostables, ils reviennent en position initiale ; s'ils sont bistables, ils conservent leur état. L'automate continue à fonctionner.

En cas de chute de pression, les tiges des vérins à simple effet rentrent grâce à l'effet de leur ressort intégré alors que les vérins à double effet conservent leur position. Si le mouvement d'un vérin doit être immédiatement bloqué, il convient d'installer des bloqueurs à proximité de leurs orifices.

Ces considérations permettent de retenir les solutions ci-dessous. La sécurité doit être optimale et mérite toujours d'être analysée avec beaucoup d'attention, cas par cas.

effecteur	actionneur	préactionneur	motivation
poussoir	vérin double effet	distributeur bistable	bloquer la pièce dans tous les cas
support en vé	vérin double effet	distributeur bistable	accompagner le poussoir
indexeur	vérin double effet	distributeur monostable	par défaut, libérer le mouvement du plateau
avance de l'outil	vérin double effet avec régulateur hydraulique	distributeur monostable	extraire l'outil dans tous les cas réguler la vitesse d'avance

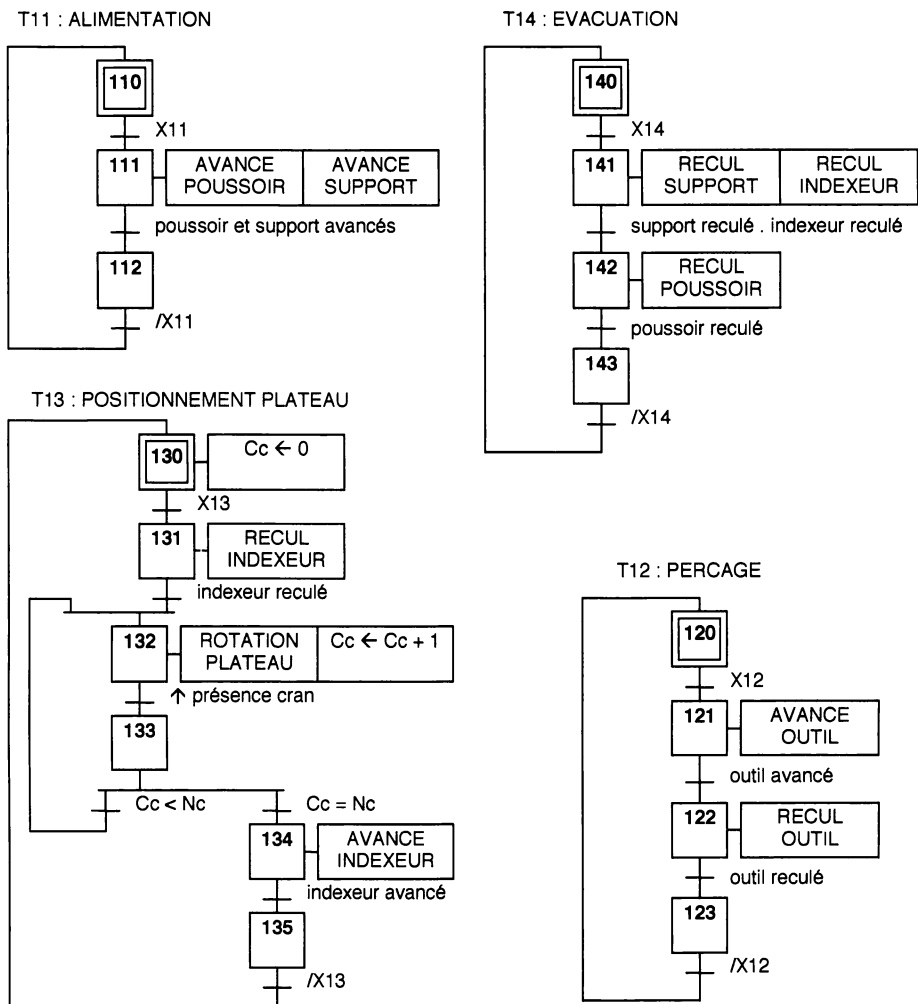


figure A-79

e) Modes de marches et d'arrêts : tous les modes (Q.5)

- Gemma complété

La figure A-80 représente en traits interrompus le Gemma de la marche normale (identique à celui de la figure A-75) augmenté, en traits pleins, des modes de marches et d'arrêts supplémentaires. Les rectangles-états font autant l'objet de modes de fonctionnement que de situations intermédiaires. Les modes sont les suivants : *réglage*, *initialisation de la partie opérative* et *arrêt de défaillance*. Les situations intermédiaires sont les suivantes : *interruption du cycle* et *redémarrage autorisé après intervention*.

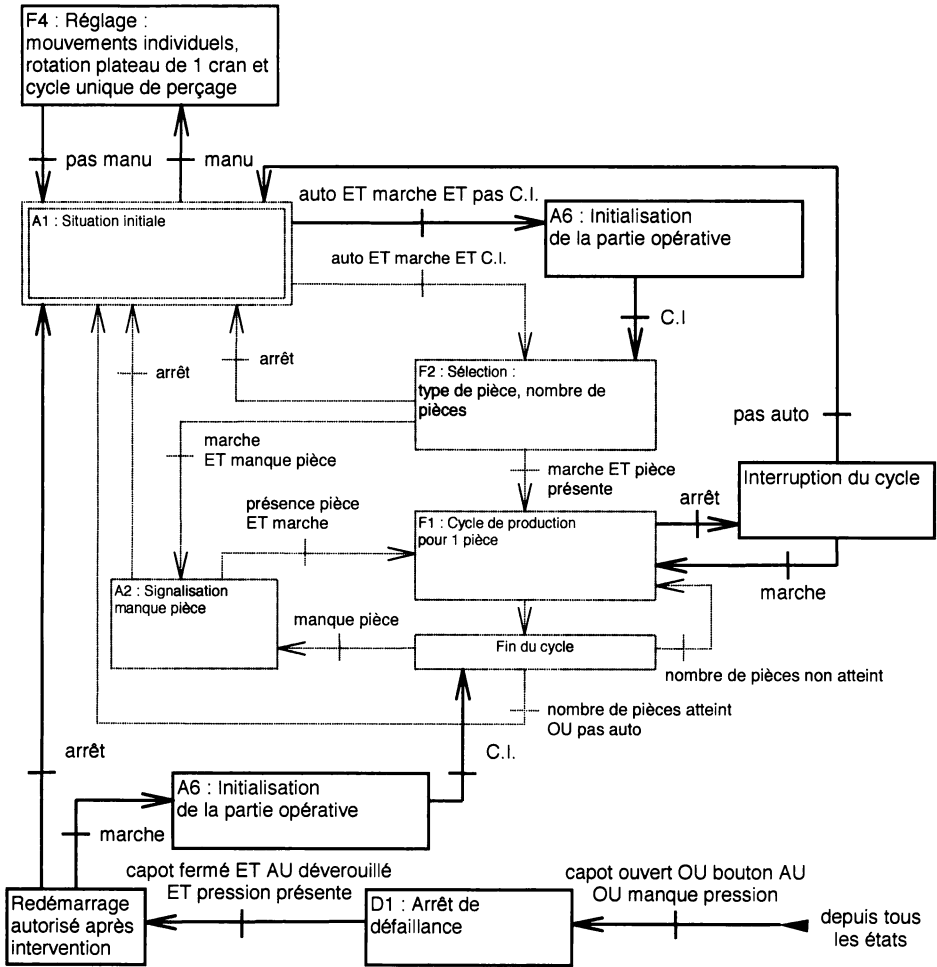


figure A-80

Le mode *Initialisation de la PO* est représenté par 2 rectangles distincts du fait que les liaisons en aval ne sont pas les mêmes selon que ce mode est obtenu à partir de la situation initiale ou à partir de la défaillance.

L'arrêt de défaillance est obtenu depuis tous les états. On rappelle que la fonction de ce mode *n'est pas* d'agir sur la distribution des énergies, fonction réalisée physiquement par câblage, mais simplement de mettre la partie commande en cohérence avec la réalité de la partie opérative. La qualité et la sécurité des redémarrages sont conditionnées par cette précaution.

- Grafcet de conduite complété

Là encore, la transcription du Gemma n'est pas tout à fait immédiate. Le mode *réglage* est obtenu grâce au bouton *manu*. Le retour en situation initiale est obtenu en tournant le commutateur en

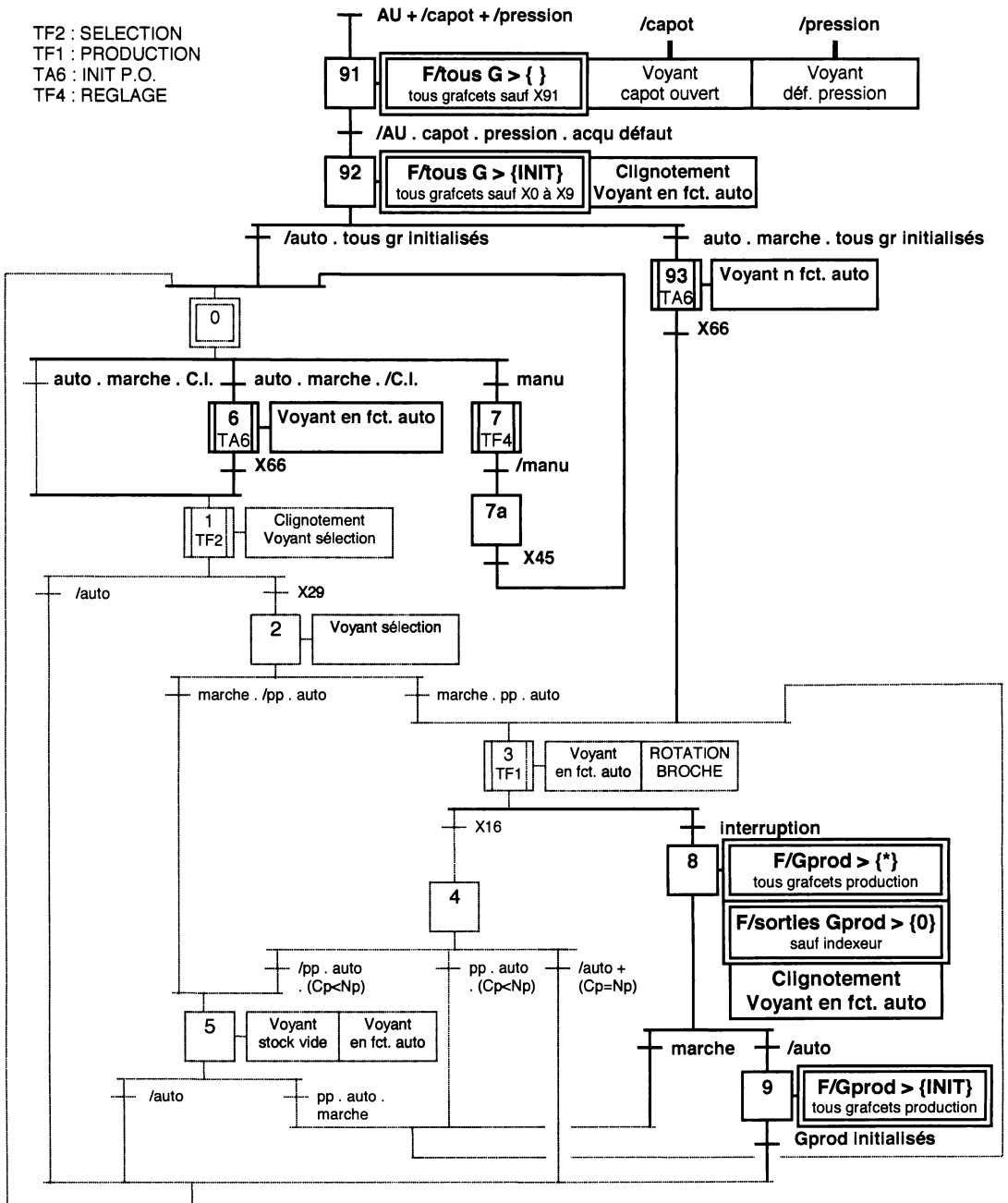


figure A-81

position neutre. Ceci peut cependant créer un problème par non respect du principe d'*unicité du mode*. En effet, si aucune précaution n'est prise, l'opérateur peut quitter le réglage et immédiatement engager le mode *production* alors qu'un cycle unique de perçage, lancé en mode *réglage*, est encore en cours d'exécution. Le Grafcet est un outil qui permet cette rigueur. L'étape

X7a assure l'unicité du mode : la situation initiale (étape X0) ne peut être réactivée que sous condition de la fin effective du mode réglage (compte-rendu délivré par l'étape X45).

Le grafctet de conduite est également enrichi de la possibilité d'interruption immédiate du cycle. Le symbole ***F/Gprod>{*}*** indique que les étapes de tous les grafctets de la tâche production (coordination et tâches subalternes) doivent être figées en l'état pendant toute la durée d'activation de l'étape X8.

Si les étapes sont figées, et qu'aucune autre disposition n'est envisagée, les sorties associées aux étapes actives restent à l'état 1. Les moteurs continuent à tourner, etc. ce qui n'est certainement pas l'effet recherché. Les sorties doivent donc être inhibées. On peut soit les forcer à zéro par des instructions qui sont prioritaires par rapport à celles de leurs équations habituelles, soit compléter leurs équations comme ci-dessous (Xf étant l'étape qui fige la situation courante) :

exemple d'équation : SORTIE \leftarrow X_m + X_n
 équation complétée : SORTIE \leftarrow (X_m + X_n) . /Xf

Après figeage, deux possibilités sont offertes.

Continuer le cycle à partir de la situation courante nécessite le défigeage des étapes des grafctets du mode *production* ainsi que la réactivation des sorties adéquates. Ces effets sont obtenus par la séquence [X8 → X3]. Remarque : si un perçage était en cours de perçage lors de l'interruption du cycle, ce trou ne sera pas terminé (dans le cas où cette situation serait gênante, il convient de compléter le grafctet par des tests et séquences adéquats).

Quitter le mode *production* nécessite la mise en situation initiale des grafctets de ce mode afin de permettre des évolutions ultérieures (***F/Gprod > {INIT}***) réalisé par l'étape X9) . La mise en situation initiale de la partie opérative n'est pas nécessairement prévue dès la sortie de l'état d'interruption, car elle est traitée de toute manière en aval de l'étape X0.

Si les instructions explicites de figeage n'existent pas dans le langage de l'automate, il est possible de réaliser la procédure décrite ci-dessous par opérations logiques sur mots.

- Sur activation de l'étape X8, copier l'état des étapes des grafctets du mode *production* dans des mémoires de sauvegarde réservées à cet effet.
- Désactiver toutes les étapes de ces grafctets.
- Sur activation d'une étape X8a, en aval de X8 dans le cas de la reprise du cycle, copier les mémoires de sauvegarde dans la zone des étapes des grafctets. Pour le cas de l'arrêt, l'étape X9 est inchangée.

Grâce à la désactivation des étapes, les sorties sont automatiquement mises à jour : elles sont mises à l'état zéro, simplement du fait de leurs équations normales.

Si l'opérateur demande le démarrage alors que la partie opérative n'est pas en position initiale, un cycle spécifique s'en charge automatiquement (étape X6).

- Grafctet de surveillance

Les étapes X91 à X93 composent le grafctet de surveillance. La transition en amont de l'étape X91 est une transition source, toujours validée. Quelle que soit la situation de l'ensemble des grafctets, l'action sur le bouton AU ou l'ouverture du capot ou la chute de pression (détectée par un pressostat installé en sortie du bloc de conditionnement d'air) provoque l'activation de l'étape X91.

[***F/tousG>{ }***] force toutes les étapes de tous les grafctets à l'état zéro. Afin de permettre l'évolution ultérieure des grafctets, l'étape X91 doit cependant rester active. Les sorties sont déjà inhibées du fait de leur câblage (comparativement à l'étape X8, le forçage à zéro des sorties est cette fois inutile). La situation ne peut ensuite évoluer que si les trois conditions de sécurité sont obtenues. Les grafctets sont alors tous réinitialisés.

- Si l'incident n'est pas fâcheux, et après intervention, l'opérateur peut immédiatement demander le retour en production. La partie opérative étant dans un état quelconque, un cycle automatique se charge d'abord de sa mise en référence (étape X93). La tâche *sélection* n'est pas requise : la production continue pour le même type de pièces et poursuit le comptage depuis le seuil actuel.
- Le retour à l'étape X0 permet l'arrêt du fonctionnement, en vue notamment d'appeler le mode réglage.

Il peut être intéressant de dissocier les grafquets de conduite et de surveillance. La figure A-82 illustre les modifications à apporter en aval de l'étape X92.

L'étape X3 devient alors une *étape initialisable*. Les transitions en aval des étapes X94 et X95 sont des transitions puits : après leur franchissement les étapes précédentes sont désactivées, aucune autre étape n'est activée.

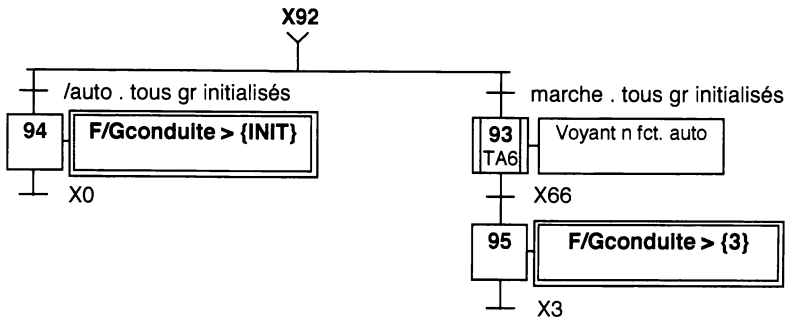


figure A-82

- Signalisation

Tous les voyants sont gérés au niveau des grafquets de conduite et de surveillance pour une meilleure maintenance du programme. Si le voyant *en fct. auto* est allumé fixe, il signale que la partie opérative est en cours d'initialisation ou que la production est en cours, même lorsque la partie opérative est au repos (stock vide) ; s'il clignote, il signale que l'arrêt d'urgence a été acquitté et que l'opérateur peut demander le redémarrage.

Le voyant du bouton *acqu. défaut* est câblé dans le circuit d'arrêt d'urgence, il ne figure donc sur aucun grafquet. Les voyants supplémentaires *capot ouvert* et *déf. pression* permettent un diagnostic rapide en cas de défaillance.

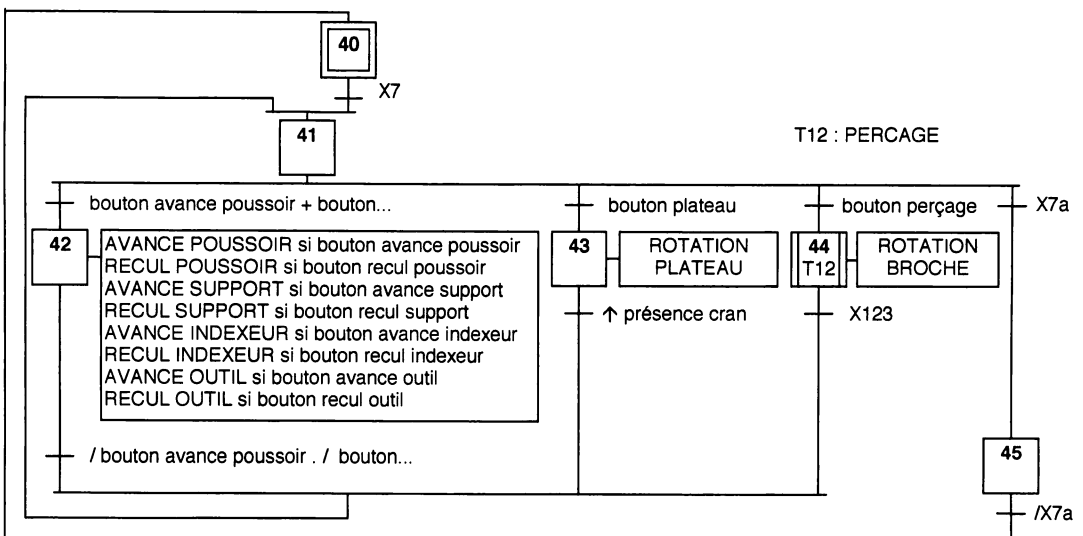


figure A-83

f) Graficets : autres modes

- Tâche réglage TF4 (figure A-83)

L'activation de l'étape X7 du graficet de conduite permet d'exécuter les différents mouvements individuels dans le désordre, et simultanément. Le relâchement de tous les boutons active l'étape X41. De nouveaux mouvements individuels peuvent être demandés, ou encore le positionnement du plateau jusqu'au cran suivant, ou encore un cycle unique de perçage. La sortie de ce mode se fait par le sélecteur de mode (sur le graficet de conduite) à condition que toute action manuelle soit arrêtée (séquence X41 → X45).

Les actions pilotées manuellement sont en outre conditionnées par des sécurités. Par exemple :

- avance indexeur si bouton avance indexeur et plateau arrêté

De même les transitions en amont des étapes X43 et X44 sont complétées comme suit :

- bouton plateau et indexeur rentré
- bouton perçage et plateau arrêté

La rotation de la broche peut être pilotée directement par câblage, en prenant la précaution que les boutons *marche/arrêt broche* ne soient validés qu'en position *manu* du sélecteur de mode.

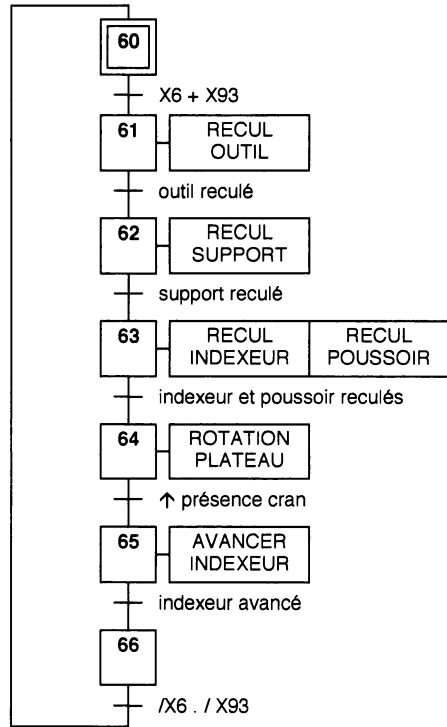


figure A-84

- Tâche Initialisation partie opérative TA6

Le cycle de ce graficet doit être étudié afin qu'il ne provoque aucun incident (collision...) et ce quel que soit l'état d'origine de la partie opérative (figure A-84).

g) Structure hiérarchisée (Q.6)

Le diagramme de la figure A-85 montre la structure hiérarchisée sur plusieurs niveaux ainsi que les liens informationnels entre les graficets. Les ordres issus du graficet de surveillance ne sont pas représentés : ils agissent sur l'ensemble de l'application. La numérotation des étapes et des tâches, si elle est judicieuse, facilite l'analyse des graficets.

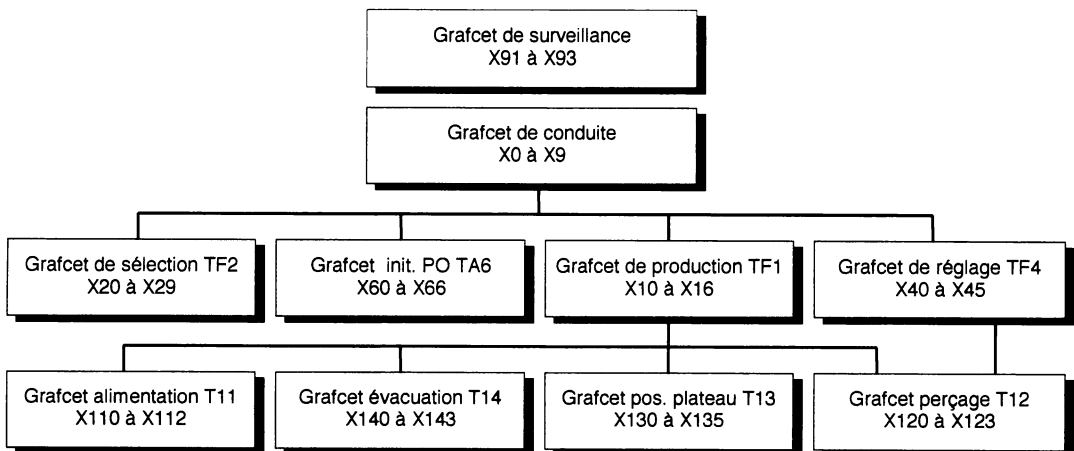


figure A-85

h) Vérification du bris foret (Q.7)

L'effet provoqué par le bris du foret est similaire à un arrêt d'urgence sauf que cette sécurité n'est pas à inclure dans le câblage. La procédure logicielle mise en route par activation de l'étape X91 peut donc être réutilisée, mais cette fois, c'est bien la désactivation de toutes les étapes qui positionne les sorties à l'état zéro. Quant au test de l'état du foret, il peut se faire d'une manière simple en vérifiant la concordance temporelle du mouvement d'avance : si la fin de course est obtenue dans le délai imparti, le foret est en bon état. Le grafcet de surveillance est complété par les étapes X94 et X95 comme l'illustre la figure A-86. Les transitions de début et de fin de ce grafcet partiel sont des transitions source et puits. A chaque activation de l'étape X121 (début de l'avance de l'outil) la durée du mouvement est surveillée. Si le capteur de fin de course délivre un signal avant la fin de la tempo, le cycle continue, X94 est désactivée. Dans le cas contraire, X95 permet d'enclencher la procédure d'arrêt. Le voyant *défaut foret* est ajouté au pupitre.

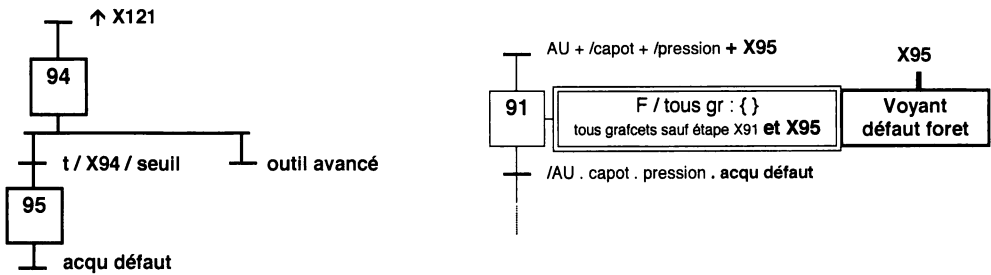


figure A-86

PARTIE B : STRUCTURE DES GRAFCETS

IV. NECESSITE D'UNE DEMARCHE

La partie A, consacrée à la connaissance de base du Grafcet et des éléments technologiques indispensables pour le mettre en oeuvre, a notamment défini sa syntaxe. Selon le niveau de précision recherché, selon qu'il s'agit du modèle ou du langage, plusieurs notations pour un même comportement sont requises. L'un des objectifs de cette deuxième partie est de montrer qu'en plus de la syntaxe, c'est la structure même des grafcets qui peut avantageusement être considérée sous des points de vue différents.

1 - L'EXPRESSION DU BESOIN

Les biens de consommation sont souvent fabriqués par des machines automatiques selon le besoin exprimé par l'utilisateur (ou stimulé si le besoin n'apparaît pas d'une manière spontanée). On distingue d'une part les producteurs de biens de consommation et d'autre part les producteurs des machines qui servent à fabriquer ces biens. Les premiers sont des entreprises spécialisées dans les procédés mais ne réalisent en général pas leurs machines. Celles-ci sont en outre souvent trop spécialisées pour pouvoir être trouvées toutes faites dans des catalogues. Les fabricants des machines spéciales répondent donc également à un besoin, celui exprimé par l'entreprise qui produit les biens de consommation (figure B-1). Nous ne nous intéresserons qu'à la construction de ces machines.

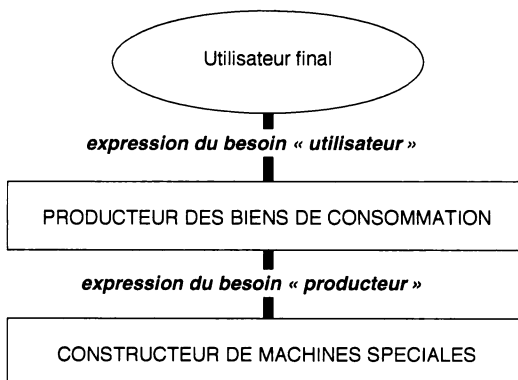


figure B-1

Sur la figure B-2, l'axe des abscisses représente l'évolution du temps. L'axe des ordonnées représente le niveau de finesse de l'étude : vers le haut, on considère le système le plus globalement possible, c'est à dire en terme de fonction à réaliser et non en terme de technologie. Vers le bas, cet axe précise de plus en plus les solutions technologiques choisies. Lors de la phase de conception, l'analyse du système se fait en partant de l'aspect global et se poursuit progressivement selon les choix retenus (on parle d'analyse descendante), c'est la procédure C. La procédure R montre que la réalisation se fait quant à elle en commençant par des éléments de détail et se poursuit en intégrant progressivement ces éléments unitaires pour aboutir au système complet, prêt à la mise en exploitation.

2 - LE CYCLE DE VIE D'UN SYSTEME

2 - 1. Conception et réalisation

Sur la figure B-2, l'axe des abscisses représente l'évolution du temps. L'axe des ordonnées représente le niveau de finesse de l'étude : vers le haut, on considère le système le plus globalement possible, c'est à dire en terme de fonction à réaliser et non en terme de technologie. Vers le bas, cet axe précise de plus en plus les solutions technologiques choisies. Lors de la phase de conception, l'analyse du système se fait en partant de l'aspect global et se poursuit progressivement selon les choix retenus (on parle d'analyse descendante), c'est la procédure C. La procédure R montre que la réalisation se fait quant à elle en commençant par des éléments de détail et se poursuit en intégrant progressivement ces éléments unitaires pour aboutir au système complet, prêt à la mise en exploitation.

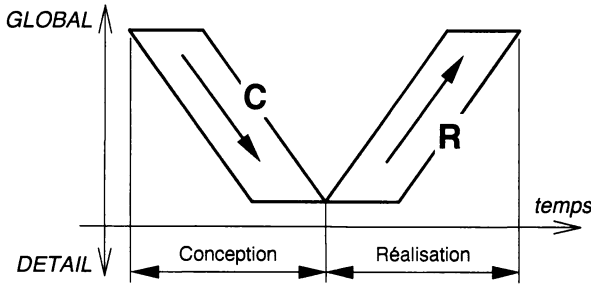


figure B-2

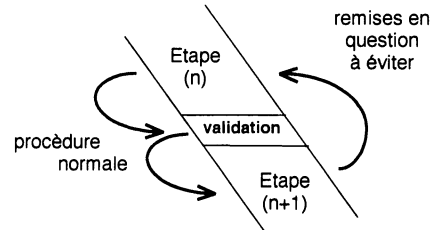


figure B-3

2 - 2. Les validations successives

La phase de conception est régulièrement ponctuée de validations qui consistent à faire le point sur l'état d'avancement du projet (figure B-3). Des décisions théoriquement irrévocables sont alors discutées, formulées et adoptées par les personnes, services ou entreprises qui collaborent au projet. Le principe de cette démarche consiste à ne pas remettre en cause ces décisions par la suite. En effet, les hésitations qui peuvent se produire lors d'un projet mal géré entraîneront inévitablement des modifications dans les travaux déjà entrepris et donc une forte et inutile augmentation du coût de la réalisation. Par ailleurs, la modularité en termes de structure et de tâche permet de préparer puis de tester séparément des sous-ensembles avant de les assembler.

2 - 3. La cohérence du projet

Le travail réalisé puis validé à l'étape $R_{(n)}$ (figure B-4) doit correspondre aux décisions prises à l'issue de l'étape $C_{(n)}$ avant d'être poursuivi. Ainsi de suite jusqu'en $R_{(finale)}$. Lors de la mise en exploitation du système, si cette démarche a été suivie, le fonctionnement global et complet est conforme aux prescriptions du cahier des charges.

En pratique, les impératifs quotidiens (délais en particulier) ne permettent pas toujours de respecter cette démarche. Par ailleurs, la création intellectuelle ne fonctionne pas en ligne droite. Elle suit un cours labyrinthique, où l'intuition joue un rôle prédominant. Il est d'autant plus important d'essayer de suivre, au moins dans les grandes lignes, une démarche rigoureuse. Les étapes de validations doivent donner au projet toute sa cohérence.

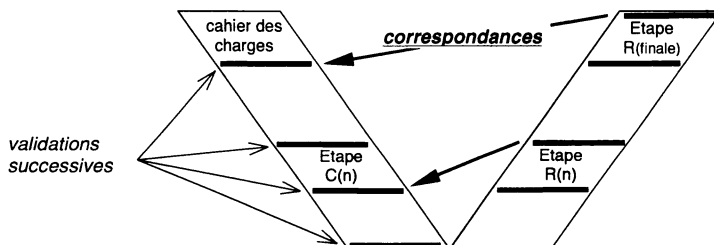


figure B-4

2 - 4. Synchronisation des métiers

La conception et la réalisation du système concernent plusieurs familles de métiers, en particulier ceux qui traitent la partie opérative et ceux qui traitent la partie commande. L'avancement harmonieux du projet ne peut se faire qu'en menant ces deux études de front, aussi représente-t-on souvent le cycle de vie d'un système comme sur la figure B-5.

On intègre également dans ce cycle la phase essentielle d'*exploitation*, qui consiste en l'utilisation pour laquelle le système a été construit, ainsi que les modifications et évolutions qui ne manqueront pas d'être indispensables. De même, la maintenance préventive fait partie intégrante de la vie du système et doit être parfaitement planifiée dès sa conception.

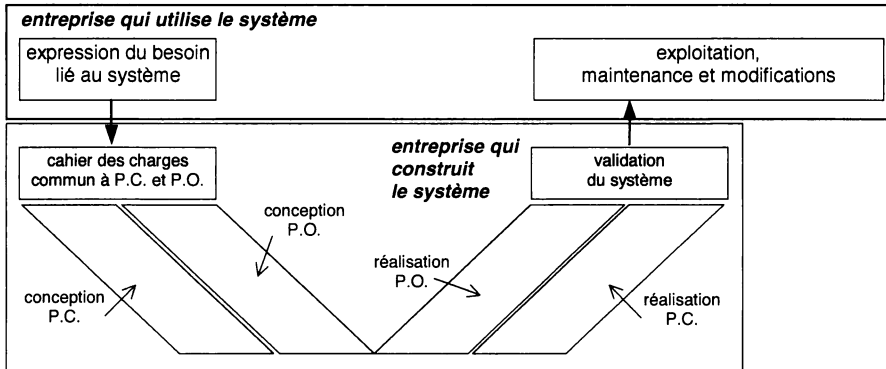


figure B-5

3 - CONSTRUCTION DE GRAFCETS

3 - 1. Avec ou sans méthode

Il est parfaitement possible de construire les grafjets d'une manière intuitive en commençant par représenter la ou les étapes initiales. Les étapes sont tracées au fur et à mesure du déroulement de la séquence que l'on souhaite modéliser. Si l'enchaînement des actions est facile à prévoir, le résultat est probablement correct et exploitable. Un problème plus ardu nécessitera plusieurs essais, des ratures, la modification de certains passages, mais le résultat peut rester bon. Il arrive cependant qu'on souhaite que le système puisse enchaîner des actions dans un ordre moins facilement prévisible, ou que l'on attende du système une très grande souplesse dans son comportement. La maîtrise d'une méthode de travail est alors souhaitable. Elle doit nous conduire à des grafjets de qualité, ayant certaines propriétés pour être efficaces :

- *permettre évidemment un fonctionnement sans ambiguïté,*
- *être de lecture claire,*
- *avoir des équations simples au niveau des réceptivités,*
- *garantir une programmation sûre et aisée.*

3 - 2. Fonctions

L'énoncé du cahier des charges technique reflète en premier lieu le besoin du fabricant en terme de finalité. Il exprime la fonction principale à assurer par le système (poinçonner et cisailer les flans, assembler les interrupteurs, trier les roues selon leur diamètre, etc.)

Des fonctions contraintes doivent également être assurées. Il s'agit dans tous les cas de la prise en compte de la sécurité : arrêt d'urgence, protection des personnes (portes d'accès avec détecteurs, basse tension pour les constituants de dialogue, etc.) et du matériel (disjoncteurs, etc.) Des fonctions contraintes supplémentaires peuvent s'imposer :

- les interactions avec un ou plusieurs opérateurs qui agissent manuellement sur le système (car la plupart des machines ne sont pas automatisées à 100%),
- d'autres interactions avec le milieu environnant : fonctions de communication, y compris avec une autre partie commande, fonctions liées aux apports en énergie, etc.
- l'environnement du système : ambiance explosible, températures extrêmes...

3 - 3. Analyse de la structure du système

La décomposition fonctionnelle s'effectue, d'une part, par rapport à la structure du système. C'est le travail de conception mécanique. Le résultat de cette analyse est une liste de solutions techniques qui permettent de définir les types de constituants de la partie opérative : éléments mécaniques, vérins, détecteurs, etc. On s'appuiera ensuite sur cette liste pour procéder aux choix technologiques : élaboration des nomenclatures, des dessins et schémas de réalisation.



figure B-6

La structure est définie par l'ensemble des constituants physiques du système.

3 - 4. Analyse des tâches à accomplir

D'autre part, la décomposition fonctionnelle s'effectue par rapport aux tâches à réaliser. C'est le domaine des automatismes. Les solutions techniques s'exprimeront alors en termes d'actions : autant physiques (rotation de moteurs, avance de tiges de vérins, etc.) qu'informationnelles (traitement de données, dialogues, etc.) La particularité des actions par rapport aux constituants étant qu'il faut les synchroniser dans le temps et non les agencer dans l'espace.

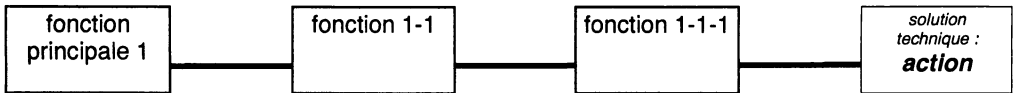


figure B-7

L'analyse permet d'affecter chaque action au niveau fonctionnel le mieux adapté. Les paragraphes ci-dessous généralisent cette notion.

3 - 5. Relation entre fonctions, tâches et actions

- On peut décider que chaque action est associée à une tâche au niveau le plus fin.

Le rôle du grafcet lié à la fonction 1-1 (figure B-8 et figure B-9) est alors la coordination des tâches élémentaires. Il organise la chronologie d'exécution des actions qui leur sont associées.

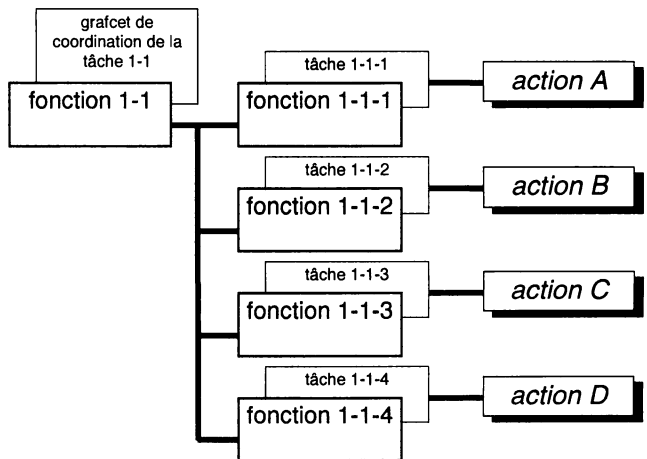


figure B-8

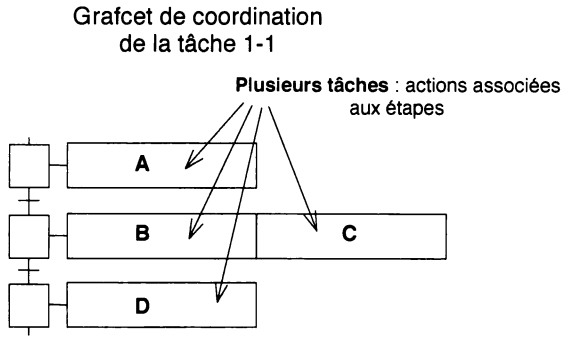


figure B-9

• A l'inverse, toutes les actions peuvent être regroupées en une seule tâche, celle qui réalise la fonction principale (figure B-10 et figure B-11). Le panachage de ces deux procédés est possible si le problème le justifie.

D'ailleurs, dans les automatismes séquentiels, on rencontre très souvent le cas de séquences figées qui sont composées d'une succession indissociable d'actions. Exemple, la saisie d'une pièce par une pince de manipulateur :

- ⇒ avancer la pince,
- ⇒ fermer la pince / desserrer la pièce,
- ⇒ reculer la pince.

Il est alors inutile de pratiquer la décomposition fonctionnelle jusqu'à isoler chacune de ces actions. La plus détaillée des fonctions peut se limiter à l'expression du geste global *saisir une pièce*.

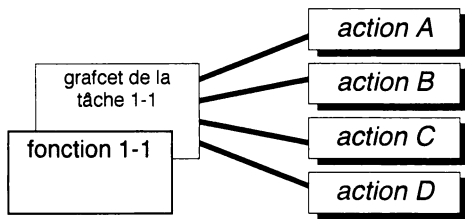


figure B-10

Un grafcet décrira globalement cette fonction comme le montre la figure B-11, qui doit être comparée à la figure B-9.

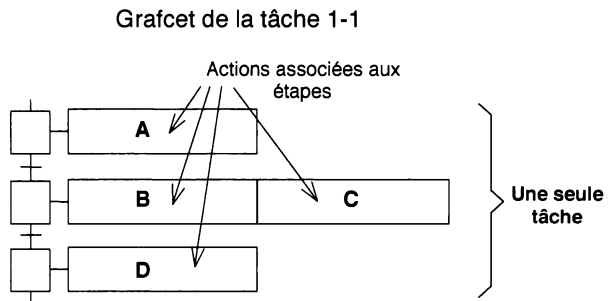


figure B-11

3 - 6. Diagramme général

Selon l'objectif de l'analyse, il est possible de combiner ces différentes approches. Le diagramme de la figure B-12 représente une configuration type qui fait intervenir les différentes possibilités de relations entre les fonctions, les tâches, les grafquets de coordination de tâches ou grafquets de tâche, et les actions.

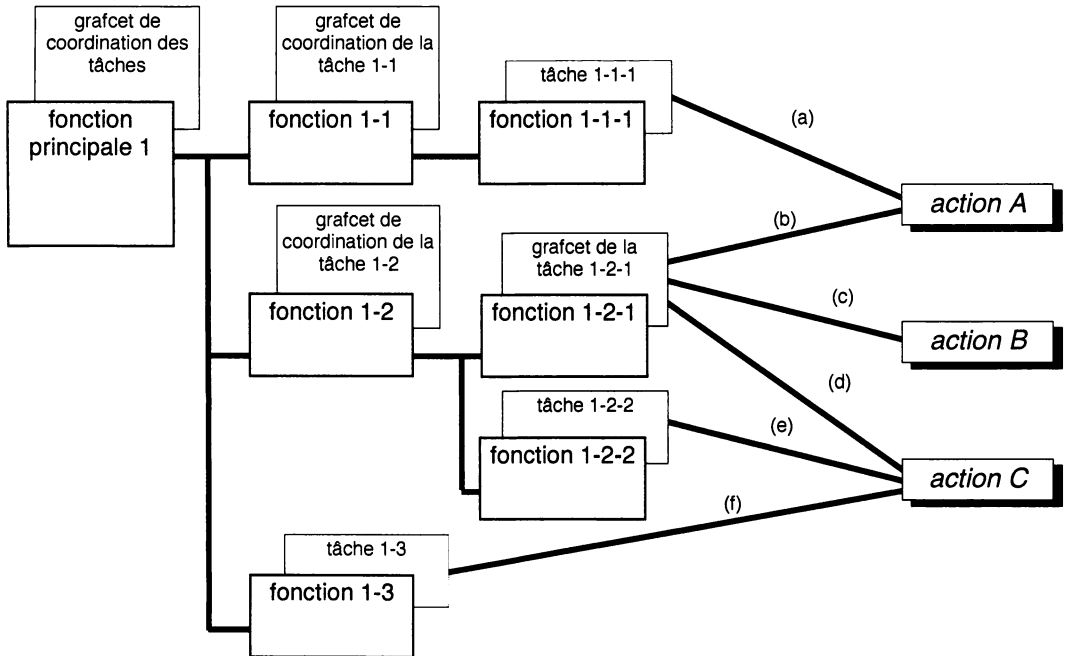


figure B-12

- **Relation (a)**

l'action A est associée à une étape du grafcet de coordination de la tâche 1-1, et est définie de manière précise par une fonction de niveau inférieur (tâche 1-1-1)

- **Relation (b)**

l'action A est directement associée à une étape du grafcet de la tâche 1-2-1

- **Relations (b) et (c) et (d)**

la fonction 1-2-1 nécessite 3 actions différentes, coordonnées par le grafcet

- **Relations (d) et (e) et (f)**

l'action C participe à la réalisation de 3 fonctions

3 - 7. Dépendance entre structure et tâche

Nous avons vu que l'analyse des fonctions est faite à la fois du point de vue du concepteur de la partie opérative auquel cas nous étudions la structure du système, et du point de vue de l'automaticien pour l'étude des tâches et de leur coordination. Il arrive que ces deux points de vue coïncident exactement. C'est le cas par exemple de la fonction *réaliser le mouvement de coupe d'une fraise 2 tailles*.

Les solutions techniques des deux points de vue se réfèrent exclusivement à la même idée : la rotation du moteur de broche est nécessaire et suffisante pour créer le mouvement de coupe.

- Solution technique pour la partie opérative : *moteur de broche*,
- Solution technique pour la partie commande : *faire tourner le moteur de broche*.

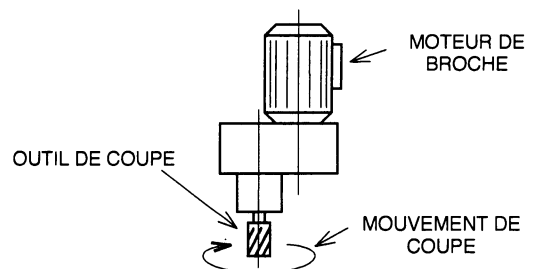


figure B-13 (vue de face)

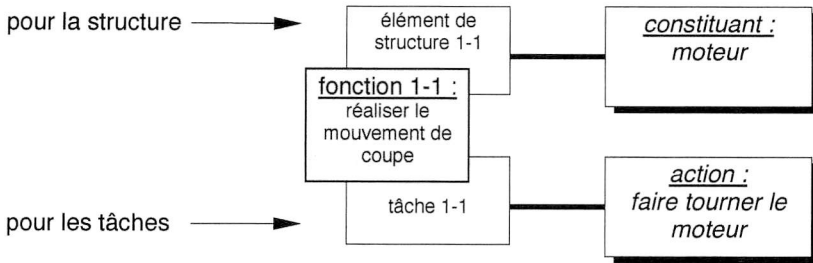


figure B-14

Cette constatation, si elle est interprétée d'une manière superficielle, peut conduire à des conclusions erronées : en particulier, confondre les termes de fonction, structure, constituants, tâche et action. En effet, cet exemple ne représente pas le cas général car très souvent, un même constituant intervient pour réaliser des tâches différentes de même qu'une tâche peut nécessiter plusieurs constituants.

Il faut donc faire une nette distinction entre une analyse de la structure (ou des organes) et une analyse des tâches. L'exemple suivant en donne une illustration.

3 - 8. Indépendance entre structure et tâche

Le poste de perçage de la figure B-15 nécessite, parmi d'autres, les fonctions d'alimentation (avancer simultanément le poussoir et le support) et d'évacuation (retirer le support, reculer *ensuite* le poussoir). Se référer également à l'exercice *Poste de perçage*.

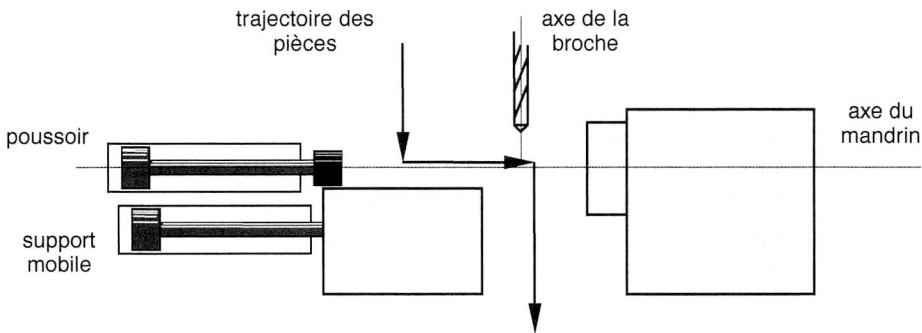


figure B-15 (vue de face)

- Plusieurs constituants pour une tâche

L'évacuation de la pièce se fait par la séquence suivante :

- recul du support mobile (la pièce reste maintenue contre le mandrin),
- recul du poussoir (la pièce tombe dans un bac).

Des actions sur deux constituants différents de la structure sont donc nécessaires pour réaliser une seule et même tâche.

- Un constituant pour plusieurs tâches

A l'inverse, le constituant *poussoir* intervient dans deux tâches :

- tâche alimenter : avance du poussoir (pour positionner et maintenir la pièce contre le mandrin),
- tâche évacuer : recul du poussoir.

Ici, deux actions différentes du même constituant sont sollicitées. Il se peut même qu'une seule action intervienne dans plusieurs tâches comme ci-dessous.

- **Une action pour plusieurs tâches**

L'exemple qui illustre ce point est extrait du *Poste de triage* du chapitre V :

- des objets cylindriques roulent sur une goulotte,
- il faut soit (a) les élever vers la goulotte supérieure (le clapet se soulève une fois qu'un objet est en place contre celui-ci),
- soit (b) leur permettre de poursuivre sur la goulotte inférieure (le clapet se soulève bien avant le passage de l'objet).

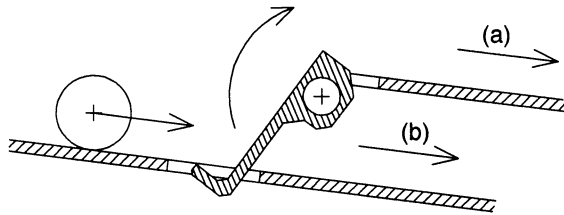


figure B-16 (vue de face)

La levée du clapet intervient bien pour réaliser deux tâches distinctes, la tâche *soulever l'objet* et la tâche *laisser passer l'objet*.

3 - 9. Finesse du grafcet par rapport au cycle de vie

La spécification du système du point de vue de son fonctionnement est réalisée à l'aide du modèle Grafcet. Dans un premier temps, le but du Grafcet est de permettre, grâce à l'outil de communication universel qu'il représente, une bonne compréhension technique entre les différents intervenants. La phase de conception devenant de plus en plus détaillée tout au long de son déroulement, le grafcet de l'application, entièrement détaillé, pourra finalement servir de cahier des charges pour le programmeur. Plusieurs grafcets, de niveaux différents, seront donc élaborés pendant la phase de conception.

Conformément à la figure B-4, ces grafcets permettront de valider les différentes étapes de la phase de réalisation du système. A chaque niveau correspond un grafcet particulier. Généralement, il est possible de distinguer trois stades :

- grafcet du principe général de fonctionnement,
- grafcet de description de la solution technique,
- grafcet en vue de la programmation (d'après les choix technologiques).

Il va de soi qu'on attend une complète cohérence entre (voir la figure B-17) :

- le grafcet qu'on peut tracer avec comme seules indications celles que fournit la lecture du programme et le grafcet qui avait été conçu en vue de la programmation,
- le grafcet qu'on peut tracer en observant le fonctionnement des actionneurs (et les signaux des capteurs) et le grafcet de description de la solution technique,
- le grafcet qu'on peut tracer en observant le système dans son ensemble, c'est-à-dire la réalisation de la tâche principale, et le grafcet du principe général de fonctionnement issu du cahier des charges technique.

Chacun de ces grafquets doit être conçu pour l'usage qu'il remplit. Si nécessaire, ils peuvent donc être de structures très différentes, et surtout, le vocabulaire employé doit être adapté au niveau de précision du grafquet.

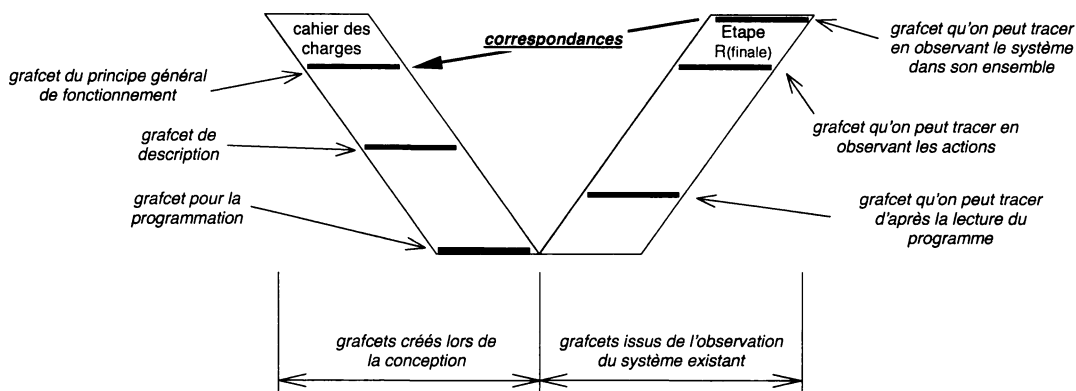


figure B-17

La vie du système se poursuit par sa mise en exploitation, sa maintenance en bon état de marche et ensuite par les modifications qu'il peut nécessiter :

- le grafquet pour la programmation est nécessaire pour la maintenance et les développements,
- par contre, le grafquet de description est à la fois concis et suffisamment détaillé pour l'exploitation du système.

3 - 10. Démarche générale

a) Le niveau de précision recherché

- De manière intuitive

Pour le cahier des charges, le grafquet exprime d'une manière générale la fonction principale du point de vue de son fonctionnement. Il doit être succinct et peut être tracé sans méthode particulière. L'intuition et l'expérience permettent habituellement de tracer rapidement un grafquet correct. On procède de la manière suivante :

- 1) tracé de l'étape initiale (en haut d'une page blanche),
- 2) 3) et 4) tracé des étapes (en leur associant les actions au fur et à mesure) et des transitions (en leur associant les réceptivités au fur et à mesure) selon l'évolution attendue des situations successives (figure B-18-a).

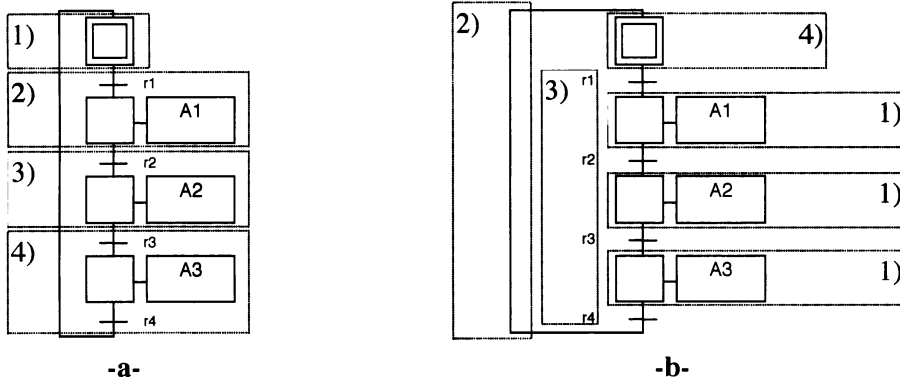


figure B-18

- Avec méthode

Pour les niveaux inférieurs, une analyse plus fine est à envisager. Les grafquets gagnent en précision. Il est alors conseillé de recourir à la suite d'interrogations proposée ci-dessous. Il en ressortira que l'ordre dans lequel on trace les divers éléments du grafquet *ne suit pas l'évolution du cycle* : le grafquet est tracé *couche par couche* (figure B-18-b),

- 1) les étapes et actions d'abord,
- 2) les liaisons et les transitions ensuite,
- 3) puis les réceptivités,
- 4) et pour terminer, l'étape initiale et quelques étapes de passage pour respecter les règles de syntaxe du Grafquet.

Bien que de manière moins évidente *à priori*, cette méthode est plus puissante et plus rigoureuse. Voici les interrogations fondamentales qui permettent de construire les grafquets de cette manière.

b) Questions-réponses

- Quelles sont les fonctions composantes ?

D'abord, il faut en faire l'inventaire et chiffrer les critères correspondants.



L'inventaire des fonctions indique les tâches à effectuer.

- Quelle est la chronologie d'exécution des fonctions ?

Ensuite, et *seulement dans un deuxième temps*, il faut organiser leur chronologie d'appel.



L'organisation de leur appel permet de construire l'ossature du grafquet (son épure).

- Quelles informations permettent le départ des tâches ?

- Et, quelles informations permettent leur arrêt ?

Ces informations sont des composantes du vecteur d'entrées (boutons, capteurs...) ou des variables internes (étapes de grafquets...).



On définit ainsi les réceptivités liées aux transitions.

- Quelles sont les fonctions de niveau inférieur ?

S'il est opportun d'en définir, les grafquets de niveau inférieur sont élaborés comme précédemment :

- faire l'inventaire des sous-fonctions, chiffrer les critères,
- organiser leur chronologie d'appel,
- déterminer les informations qui permettent le départ et l'arrêt des tâches correspondantes.

Réitérer cette suite de questions-réponses pour toutes les fonctions qui le justifient jusqu'à les épuiser. Le passage des tâches aux actions se fait progressivement. Au dernier niveau, il ne reste plus que des actions.

c) A propos de l'ossature des grafquets

La coordination des actions est déterminée par 2 facteurs :

- l'organisation même du grafquet,
- les équations logiques qui forment les réceptivités.

Ces 2 facteurs sont intimement liés car les équations dépendent en partie de l'ossature du grafquet. Le grafquet de la figure B-19-a présente une convergence puis une divergence en OU. L'étape X5 est activée si l'expression {e} est vraie immédiatement après activation de l'étape X3 (X3 est ici une étape de passage, aucune action ne lui est associée). Si le grafquet ne doit pas évoluer de X1 vers X5 même si {e} est vraie, il est possible de conserver l'allure générale de ce grafquet, mais une variable (M sur la figure b) doit gérer cette

contrainte. Il est souhaitable cependant de rechercher une structure mieux adaptée en traçant explicitement les séquences obligées comme sur la figure c :

- le grafcet représente bien mieux les évolutions possibles,
- la réceptivité en amont de l'étape X5 se limite à l'expression réellement utile.

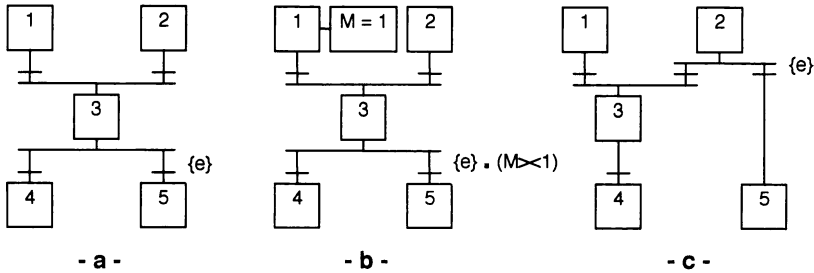


figure B-19

La décomposition fonctionnelle est illustrée par l'étude de cas du chapitre suivant. La notion de séquences obligées sera ensuite développée dans l'étude de cas du chapitre VI. Les exemples sont choisis de façon à bien distinguer ces deux aspects de la construction des grafjets.

La flexibilité de l'outil de production nous conduira dans la troisième partie de ce livre vers un dernier aspect de la méthode : le paramétrage.

V . ANALYSE DES FONCTIONS

ETUDE DE CAS N°1 : POSTE DE TRIAGE

1 - MISE EN SITUATION

Le système à concevoir est un poste de triage installé à la sortie d'une unité de production de tubes en carton de différents diamètres.

- Son but : mettre les tubes à disposition du poste de découpe chargé de la mise à longueur en fonction de la demande.
- Le principe : la pesée d'un tube à l'entrée du poste de triage permet d'identifier son type (un type par diamètre).
- Il existe sept types de tubes.
- Ils se présentent au poste de triage dans un ordre aléatoire.

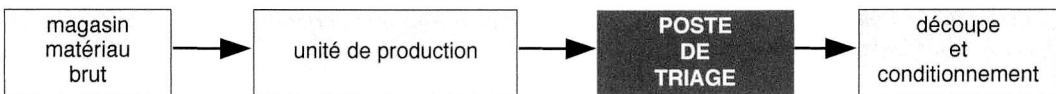


figure B-20

1 - 1. Définition de la fonction principale

Cette activité ne doit jamais être négligée. Elle évite par la suite des explications redondantes. Par ailleurs, l'idée que le client se fait du projet peut être à la fois précise s'il s'agit de son métier (expérience et habitudes) et pourtant confuse car aucune nouvelle machine ne peut réellement s'apparenter *a priori* à une machine existante.

Dans le cas étudié, la fonction principale est de :

transférer les tubes de type (i) vers l'emplacement (i)



figure B-21

Remarquons que l'expression de cette fonction se limite volontairement à quelque chose d'extrêmement général pour l'instant.

1 - 2. Inventaire des critères

	critère	niveau
tube	diamètre extérieur (paroi = 5 mm)	30,40,50,60,70,80,90 mm
	longueur	unique : 1000 mm
transfert	cadence	10 tubes/minute
environnement	robot (saisit les tubes à la sortie du poste de triage)	6 axes
	nombre de tubes de chaque type à stocker	maxi 8 tubes par type

2 - LE CAHIER DES CHARGES TECHNIQUE

2 - 1. Représentation sous forme texte

Avant de procéder à la conception mécanique, il convient, pour les systèmes dynamiques, de décider de son principe de fonctionnement sous l'aspect temporel. En effet, plusieurs options peuvent être retenues. En voici trois variantes.

a) Première variante :

« Si un tube se présente au poste de triage, il est pris en compte et orienté vers sa destination à condition qu'aucun autre tube ne soit en cours de triage » ; on peut dire que *les cycles de triage se font alors successivement*.

b) Deuxième variante :

« Si un tube se présente alors qu'un cycle de triage est déjà en cours, ce nouveau tube est pris en compte immédiatement, sans attente de la fin du cycle en cours » ; *deux ou plusieurs triages* (le nombre reste à définir) *peuvent alors avoir lieu en même temps*, avec la limite suivante : deux tubes de même type ne peuvent pas être triés en même temps.

c) Troisième variante :

Le type de fonctionnement est le même que pour la deuxième variante, mais *le triage simultané de deux tubes de même type est cette fois autorisé*.

Ces solutions sont de complexité variable, très probablement de coût variable. Il paraît clair que le choix hâtif de l'une ou l'autre de ces solutions peut conduire par la suite à des remises en cause du cahier des charges, pour raison de budget ou de contraintes technologiques. Le choix du principe de fonctionnement ne doit donc être fait que conformément à l'expression d'un réel besoin.

Les descriptions des trois variantes ci-dessus peuvent rapidement devenir confuses et sujettes à interprétation dès que le problème n'est plus aussi simple. Il sera donc profitable, pour la bonne compréhension du cahier des charges, et avant de définir la solution retenue, d'accompagner ce texte par un élément graphique exempt de toute ambiguïté. Les trois variantes sont donc présentées maintenant sous la forme de grafquets.

2 - 2. Représentation sous forme Grafquet

a) Première variante : triages successifs

La structure de grafquet qui montre très explicitement qu'un seul cycle de triage peut être exécuté à la fois est celle avec divergence en OU, appelée structure à sélection de séquence. Le grafquet de la figure B-22 montre effectivement qu'un seul des cycles de transfert peut avoir lieu en même temps.

A propos

Sur ce grafquet, le transfert d'un tube (i-1) est en cours. Qu'est-ce qui empêche le démarrage du transfert du tube de type (i+1) alors qu'il est présent à l'entrée du poste de triage ? (la réceptivité correspondante est vraie).

Réponse

Rappelons que pour qu'une transition soit franchie, il faut deux conditions : la réceptivité doit être vraie (ceci est bien le cas) ET la transition doit être validée (et ceci n'est pas le cas). Pour que la transition en question soit validée, il faut que l'étape précédente soit active (Xa), ce qui ne sera le cas que lorsque le transfert en cours sera terminé (réceptivité « transfert terminé » vraie).

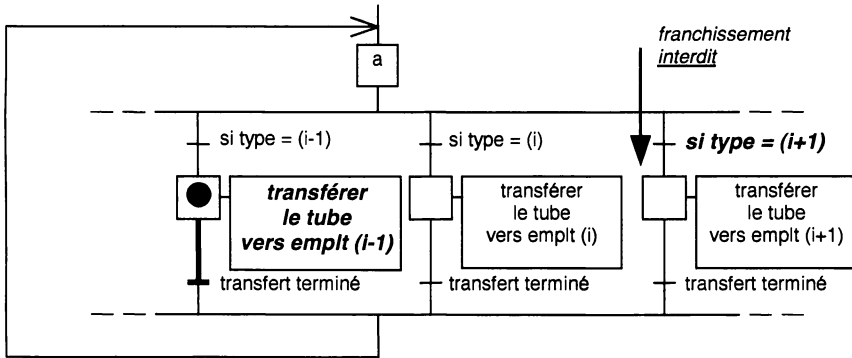


figure B-22

b) Deuxième variante : triages simultanés avec limite imposée

A propos

La première variante nécessite une structure à sélection de séquence car il s'agit effectivement de sélectionner une et une seule séquence. Peut-on directement (hâtivement) en déduire que pour obtenir des triages simultanés, il convient d'utiliser une structure de grafcet à séquences simultanées ? (divergence en ET).

Réponse

Bien que la formulation de la question le laisse supposer, une telle structure n'est absolument pas adaptée. En effet, les séquences simultanées sont systématiquement toutes actives en même temps. Dans notre problème par contre, seules quelques-unes des séquences sont actives à la fois, ou même une seule. De plus, elles ne démarrent pas en même temps.

La simultanéité des séquences est immédiatement perceptible lorsque chaque type de tube est traité par un grafcet spécifique, tel que représenté ci-dessous. Ainsi, toutes les étapes étant actives dès le démarrage, c'est l'indépendance des différents cycles qui est mise en évidence. On notera que deux tubes de même type ne peuvent pas être triés en même temps.

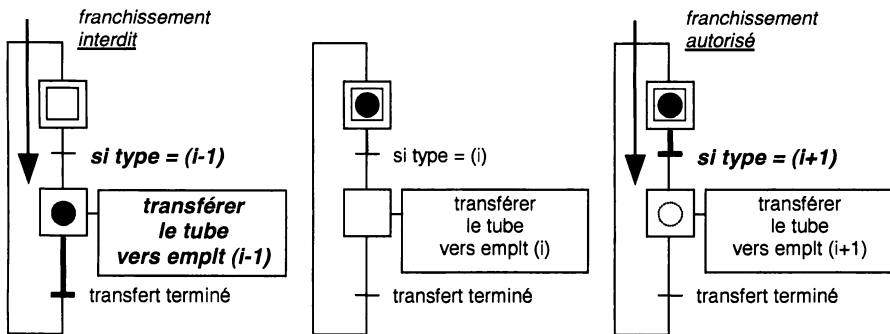


figure B-23

A propos

• Quelles sont les règles qui permettent à deux de ces grafquets (ou davantage) d'évoluer en même temps ?

• Quelle règle empêche le triage d'un tube dont un type identique est en cours de triage ?

Réponses

• Il s'agit de nouveau des règles de franchissement des transitions. Tous les grafquets débutent par une étape initiale, active au démarrage du système. Toutes les transitions qui les suivent immédiatement sont validées, il suffit alors que l'une ou plusieurs des réceptivités deviennent vraies, à n'importe quel moment, pour que les grafquets correspondants évoluent et provoquent les cycles de transfert appropriés.

• Sur le grafcet de gauche, on voit que l'étape initiale n'est plus active, la première transition n'est donc pas validée. Que la réceptivité associée à cette transition soit vraie ou non ne change alors rien à la situation. Mais dès que le transfert en cours est terminé et que ce grafcet est réinitialisé, un nouveau cycle peut immédiatement démarrer.

c) Troisième variante : triages simultanés sans la limite imposée

Afin d'autoriser le triage simultané de deux ou de plusieurs tubes de même type, on écrit les grafkets suivants. Attention, la *structure* globale des grafkets est identique à celle de la variante précédente, seul le *vocabulaire* utilisé est différent. Cette fois, les grafkets sont réinitialisés *dès* que le cycle de triage correspondant a démarré et ils sont donc constamment prêts pour entrer de nouveau en fonctionnement.

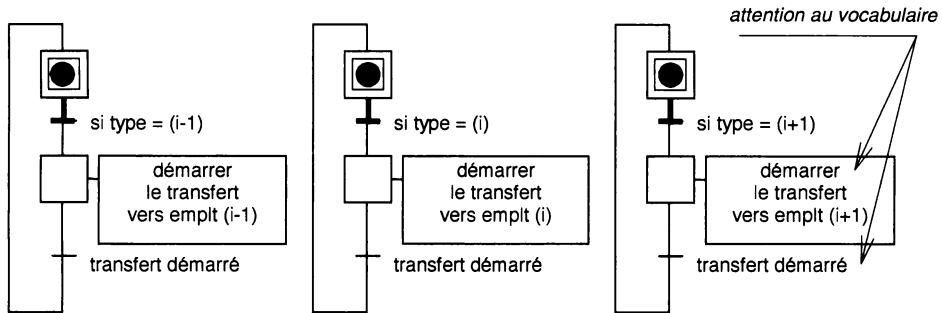


figure B-24

A propos

Sur ces grafkets, combien de temps les étapes qui font démarrer les transferts restent-elles actives ?

Réponse

Cette question ne se pose pas dans ce cas précis, ces grafkets n'étant pas destinés à la programmation. Cependant, si cela était le cas, ce temps d'activation dépendrait de l'algorithme d'interprétation du grafket au moment du codage du grafket dans le langage de l'automate, ou de l'algorithme intégré dans l'automate. Cette durée varie de quelques microsecondes (le temps d'exécution de quelques instructions) à quelques millisecondes (le temps de scrutation du programme) selon la manière de rédiger le programme.

2 - 3. Validation d'une solution

Parmi les trois variantes proposées, il convient de choisir celle dont les coûts d'étude et de réalisation seront les moins élevés en proposant un équipement strictement conforme au besoin réel. Le critère essentiel est dans ce cas *la cadence à respecter*. Sans les démontrer, nous admettrons les idées suivantes :

- le système le plus simple est le moins coûteux,
- la cadence d'un système de production dans son ensemble est celle de son maillon le plus lent (le goulot d'étranglement).

Pour la suite, nous posons comme hypothèse que l'étude du système de production complet vérifie que *la cadence autorisée par la première version, la plus lente mais aussi la plus simple, est compatible avec le fonctionnement global*.

Fonctionnement retenu : 1^{ère} variante (cycles successifs)

3 - PRINCIPES TECHNIQUES

Le cahier des charges commun aux parties opérative et commande étant validé, les conceptions du mécanisme et de l'automatisme peuvent être menées parallèlement.

Décrivons brièvement les choix relatifs au principe mécanique.

3 - 1. Schéma d'ensemble

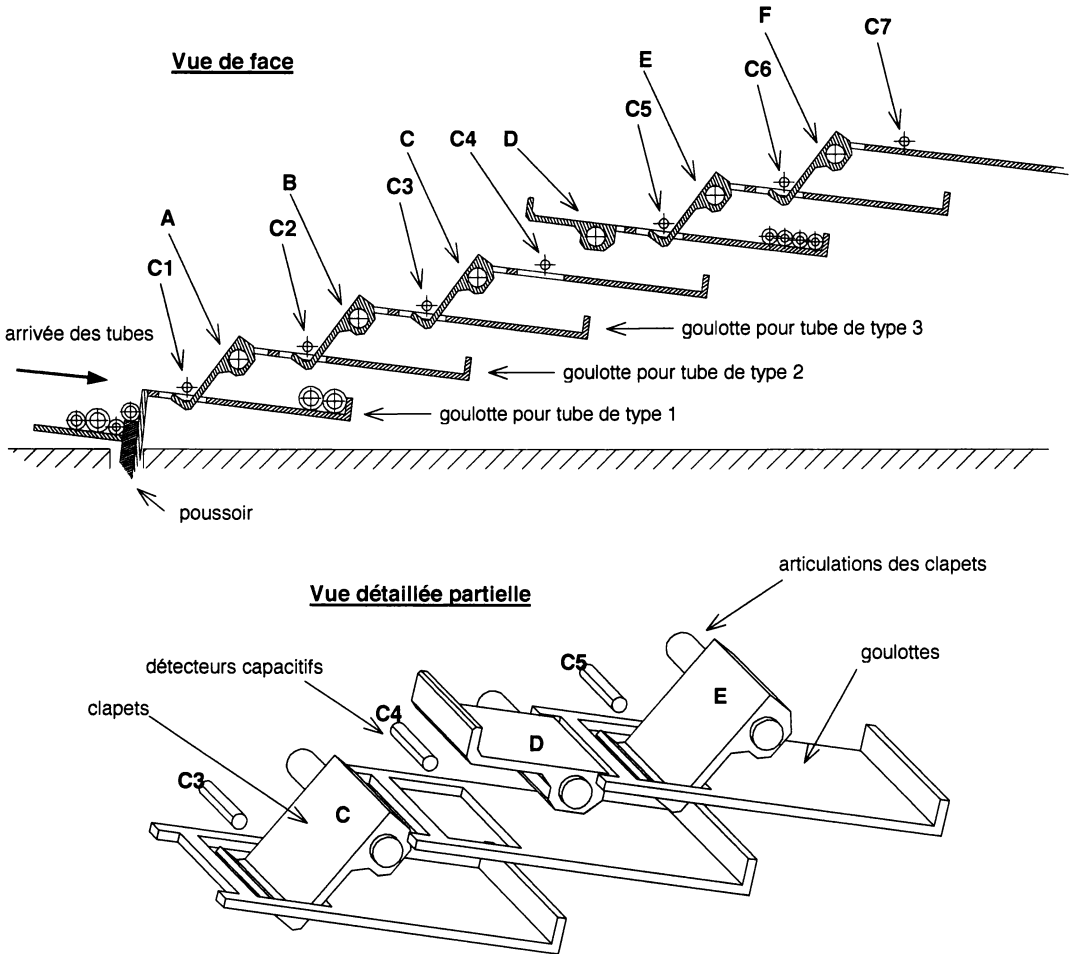


figure B-25

Les figures sont représentées alors qu'un tube de type 4 est en cours de triage : le poussoir élève le tube vers la première goulotte et le clapet n°4 est déjà soulevé.

3 - 2. Principes technologiques

- La géométrie du produit autorise son déplacement dans des goulottes par simple gravité et roulement.
- Chaque type de tube est rangé dans une goulotte spécifique. Les 7 goulottes sont superposées. Les tubes de type 1 doivent être rangés dans celle du bas, ceux de type 7 dans celle du haut.
- Les détecteurs capacitifs numérotés C1 à C7 sont sensibles au passage des tubes mais ne réagissent pas lorsqu'un clapet se soulève (en raison de la portée limitée des détecteurs choisis et du jeu important entre ces détecteurs et les clapets : voir la figure B-26).

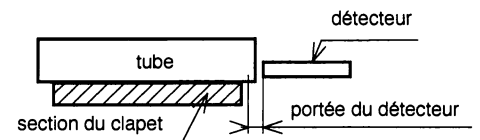


figure B-26

- Les vérins sont tous pilotés par des distributeurs monostables.
- La reconnaissance des tubes se fait par leurs masses respectives.

3 - 3. Fonctionnement du mécanisme

Lorsqu'un tube se présente, le type en est déterminé par pesage. Le clapet de la goulotte adéquate se soulève (elle laissera passer le tube lors de son arrivée) et le poussoir d'alimentation monte. Le tube est hissé de goulotte en goulotte jusqu'à ce qu'il passe en dessous du clapet qui est déjà soulevé. Le tube étant dans la bonne goulotte, toutes les actions en cours sont arrêtées (les clapets s'abaissent et le poussoir descend). Le cycle est terminé. Un nouveau cycle peut démarrer.

4 - DESCRIPTION DU FONCTIONNEMENT

4 - 1. Grafcet de description

Le grafcet de la figure B-27 vient rapidement à l'esprit. D'excellente facture du point de vue de sa fonction descriptive, il a été élaboré intuitivement par reproduction directe du cycle.

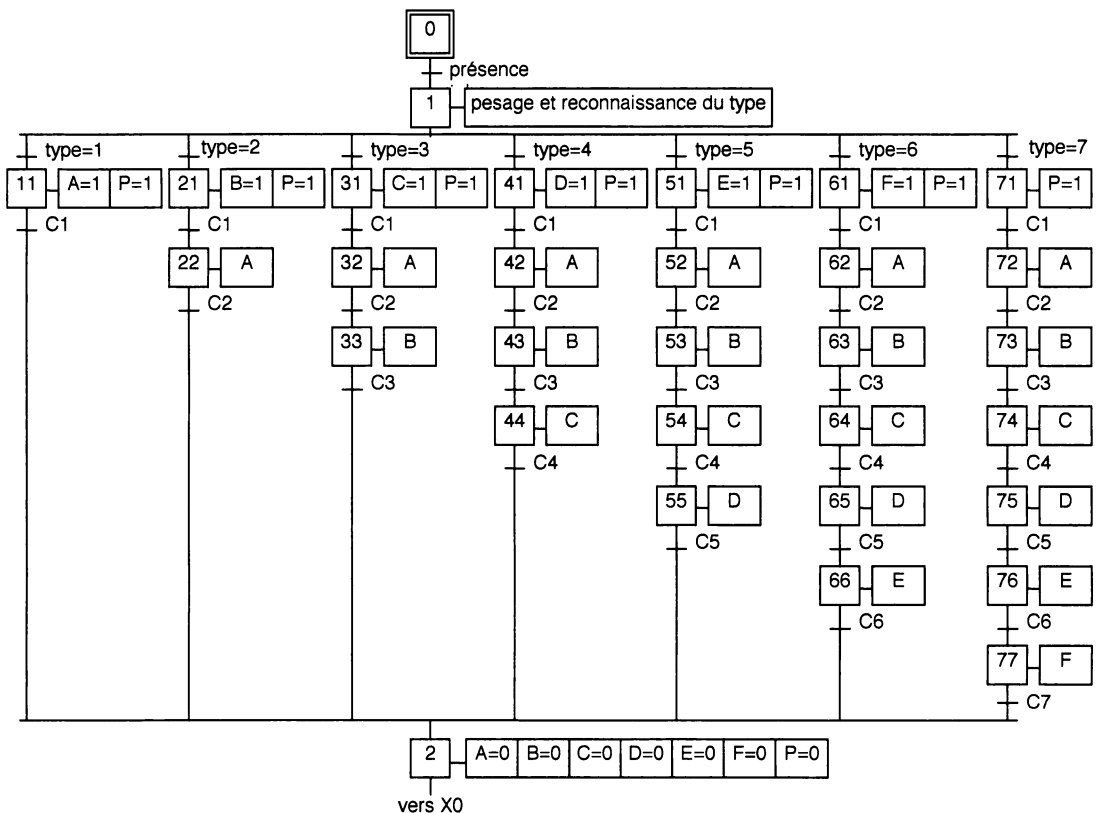


figure B-27

- A l'arrivée d'un tube, détection et reconnaissance de son type.
- Selon son type, ouverture de la goulotte correspondante, d'où la sélection de séquence dans le grafcet, et montée du poussoir.
- Dans chaque séquence, description complète du cycle d'amenée du tube vers sa goulotte de destination.

- Remise en état initial des actionneurs.
- Pour démarrer le cycle suivant, il y a soit attente de l'arrivée d'un nouveau tube, soit prise en compte immédiate du tube déjà présent.

4 - 2. Chronogramme : analyse des états des étapes

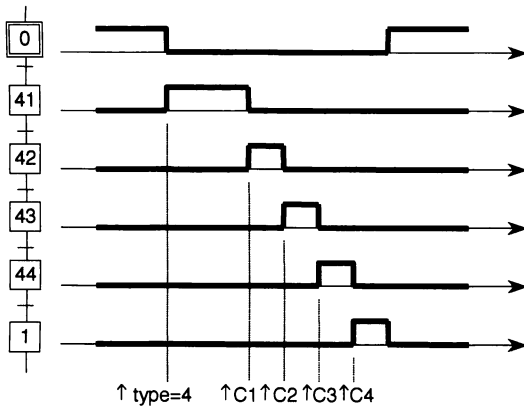


figure B-28

L'analyse de l'évolution des étapes de ce grafcet peut aisément se faire sous forme de chronogramme. Afin d'en simplifier l'expression, la figure B-28 est limitée à la représentation de la séquence qui correspond au type n°4 pour exemple.

L'aspect purement séquentiel de cette partie du grafcet est ainsi parfaitement mis en évidence.

4 - 3. Tentative de simplification en vue de la programmation

On constate immédiatement que ce grafcet entraînera de nombreuses répétitions lors de la programmation. En effet, les séquences se ressemblent. Celle d'un type de tube donné contient intégralement toutes celles des types de numéro inférieur (voir figure B-29). Seul le nom du clapet qui doit s'ouvrir dès le début du cycle varie selon le type de tube en présence. A moins de disposer d'une fonction de type *copier-coller*, il paraît naturel de vouloir faire l'économie de ces répétitions lors de la phase de programmation. Notons que le grafcet de description de la figure B-27 représente en fait le cahier des charges pour le programmeur. Le

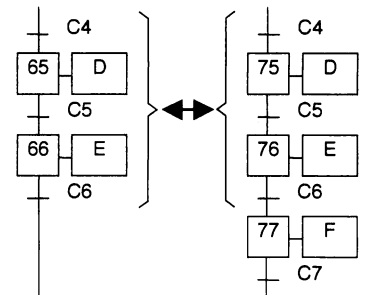


figure B-29

fonctionnement final doit être conforme aux attentes qui y sont exprimées, mais le programmeur décide seul de la manière d'y parvenir. Il tentera probablement de simplifier ce grafcet : l'idée consiste à n'écrire qu'une seule fois la séquence complète et à aménager ensuite des sauts d'étapes qui permettent de la quitter au moment opportun (figure B-30).

4 - 4. Conclusion

- Ce nouveau grafcet nécessite effectivement moins d'étapes. Par contre, les équations ainsi que la structure générale du grafcet sont plus complexes, ce qui représente une source d'erreur potentielle.
- Le fonctionnement du système n'est pas aussi clairement exprimé que précédemment. En fait, le grafcet de description initial permet à celui qui l'étudie de très bien suivre le cheminement d'un tube de type donné. En vertu de sa clarté, nous allons le conserver en tant que cahier des charges de l'analyse détaillée du fonctionnement. Nous reprenons l'étude à partir de ce point et négligeons volontairement le « grafcet simplifié ».

5 - ANALYSE DETAILLEE DU FONCTIONNEMENT

En premier lieu, ne pas oublier qu'il ne faut tracer aucune étape de grafcet avant d'avoir analysé les fonctions à réaliser par le système.

5 - 1. Inventaire des fonctions

La fonction principale a déjà été formulée. Il s'agit maintenant de la décomposer en fonctions de niveau inférieur. C'est la phase la plus délicate à réaliser. Une mauvaise décomposition entraînera un résultat aberrant.

Pour effectuer correctement cet inventaire, deux précautions doivent impérativement être prises.

- Les fonctions doivent être exprimées dans un vocabulaire qui fait abstraction des organes qui les réalisent afin d'éviter la confusion entre analyse des fonctions et analyse des organes.

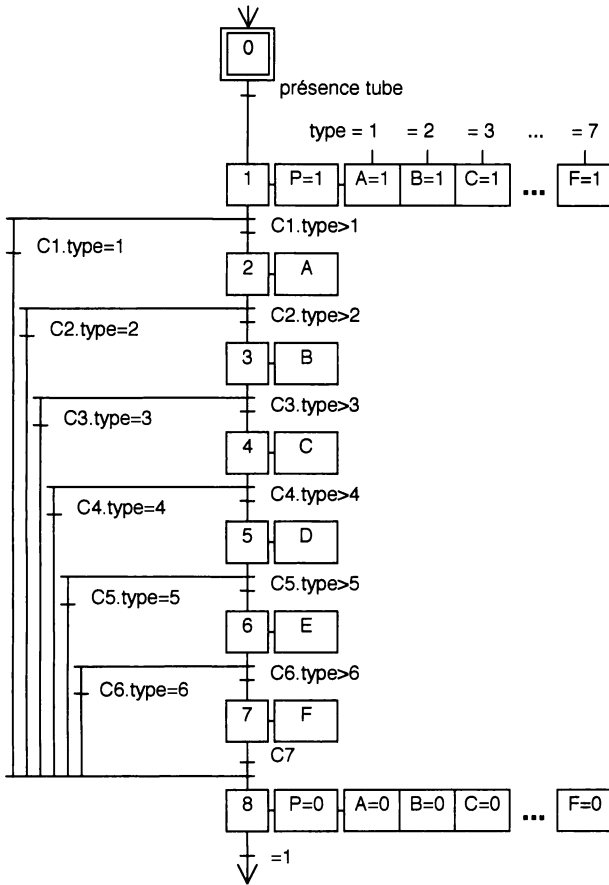


figure B-30

- Il faut éliminer de la réflexion toute référence à la chronologie du cycle. Dans un premier temps, on aura tendance à négliger cette contrainte car elle nécessite un effort intellectuel important.

Afin de bien illustrer ce deuxième point, les fonctions seront présentées pêle-mêle comme sur la figure B-31. Toute autre représentation contient en effet *implicitement* une idée de chronologie, que ce soit un tableau, une liste ou un diagramme.

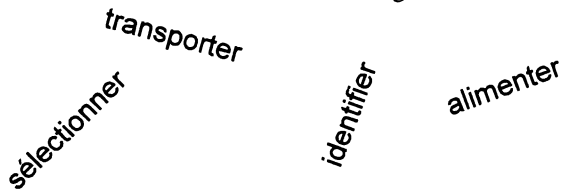


figure B-31

Pour la même raison, les fonctions sont décrites dans un ordre absolument quelconque.

a) Fonction transporter

Cette fonction démarre lorsque le tube commence à rouler sur la première goulotte, tout au début de sa trajectoire. Elle consiste à transporter le tube jusqu'à sa goulotte de destination, *quelle qu'elle soit*, et se termine lorsque le tube est arrivé dans cette position.

b) Fonction *identifier*

Le tube étant en place sur la balance, une pesée est effectuée et le résultat est interprété afin d'attribuer un numéro de type au tube en question.

c) Fonction *alimenter*

Le tube est élevé vers la première goulotte.

d) Fonction *sélectionner*

La goulotte correspondante au type de tube à trier est ouverte afin de permettre le passage du tube lorsqu'il se présente.

Rappel : l'expression de ces fonctions n'a fait intervenir aucun élément technologique. En effet, la description de la fonction *sélectionner* dit :

- [la goulotte] ... [est ouverte] \Rightarrow phrase purement descriptive et non :
- [le clapet est soulevé] \Rightarrow suggère le fonctionnement d'un mécanisme.

5 - 2. Inventaire des critères relatifs aux fonctions

Les critères doivent être définis en collaboration avec l'équipe chargée de la conception mécanique. Il s'agit dans ce cas d'informations qui concernent :

- les vitesses des déplacements,
- les courses et débattements,
- la précision nécessaire de la pesée...

5 - 3. Coordination des fonctions

La question des liens chronologiques entre les fonctions *ne se pose que maintenant*. La réflexion est fractionnée et limitée à ce seul point, elle est donc facilitée.

Lorsqu'un tube se présente, il doit être identifié avant toute autre chose, cela ne fait aucun doute. Par contre, plusieurs variantes peuvent être envisagées pour l'ordre d'appel des trois autres fonctions. Les impératifs sont les suivants :

- *sélectionner* et *transporter* doivent être exécutées en même temps,
- *alimenter* doit être appelée au plus tôt en même temps que *sélectionner* pour permettre le bon fonctionnement pour le triage des tubes de type 1.

Globalement, il est donc possible de provoquer l'appel de ces trois fonctions en même temps une fois l'identification terminée.

De cette analyse découle le grafcet de coordination des tâches (figure B-32). Elle permet de mettre en évidence la différence entre fonction et organe. En effet, la fonction *transporter* est, dans ce cas, enclenchée dès le début de l'alimentation ; or le premier clapet lié à la fonction *transporter* ne se mettra en mouvement que plus tard, lorsque le tube se présente devant ce clapet. La fonction est enclenchée alors que le mouvement correspondant n'a lieu qu'ultérieurement ! Ce point particulier sera encore développé lors de l'analyse du grafcet complet.

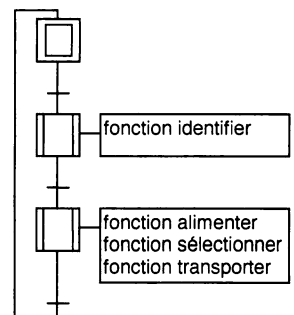


figure B-32

5 - 4. Ebauche des grafquets de tâches

Chacune des fonctions fait l'objet d'une traduction en une ébauche de grafquet de tâche. Au point où en est l'étude, seules les informations disponibles sont mises en place. Si certaines tâches ne sont pas suffisamment définies pour pouvoir être intégralement décrites, elles peuvent faire appel à un grafquet de niveau hiérarchique inférieur, complété par la suite, lorsque cela sera possible. Ceci peut être le cas d'une fonction pour laquelle la conception mécanique n'est pas encore terminée.

a) Grafquet lié à la fonction *identifier*

La pesée s'effectue lorsqu'un tube est présent à l'entrée du poste de triage, c'est-à-dire lorsque la valeur délivrée par la balance est différente de zéro. La temporisation permet la stabilisation de cette valeur avant enregistrement. Il reste à décider si ces actions sont traitées par un grafquet de tâche ou directement écrites dans le grafquet de coordination des tâches.

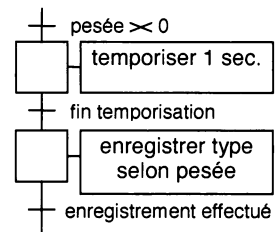


figure B-33

b) Grafquet lié à la fonction *alimenter*

Une seule action est nécessaire pour réaliser cette fonction : élever le poussoir. Le grafquet se limite à une étape unique. Il est donc inutile de prévoir un grafquet de tâche indépendant, cette action pouvant prendre directement sa place dans le grafquet de coordination.

c) Grafquet lié à la fonction *sélectionner*

Revenons au grafquet de la tâche *identifier*. En raison de sa simplicité, il a été élaboré sans méthode particulière. Mais le raisonnement présenté au § IV.3 - 10 peut s'appliquer :

- *inventaire des actions* : enregistrer, temporiser
- *ordre chronologique d'exécution des actions* : temporiser puis enregistrer
- *informations qui provoquent ces actions* : arrivée d'un tube reconnue par une pesée différente de zéro, fin de la temporisation
- *informations qui permettent l'arrêt des actions* : fin de temporisation, type de tube enregistré

La fonction *sélectionner* mérite qu'on applique cette démarche. On recommande de poser les interrogations en termes de fonctions et non immédiatement en termes d'actions.

- Inventaire des sous-fonctions



figure B-34

Les sous-fonctions sont exprimées dans un vocabulaire général sans aucune référence aux organes mis en oeuvre, et sans idée préconçue de chronologie. La figure B-34 adopte la forme déjà présentée.

- **Inventaire des actions correspondantes**

Plus aucune décomposition fonctionnelle n'a lieu, aussi traite-t-on maintenant les actions qui permettent la réalisation des fonctions.

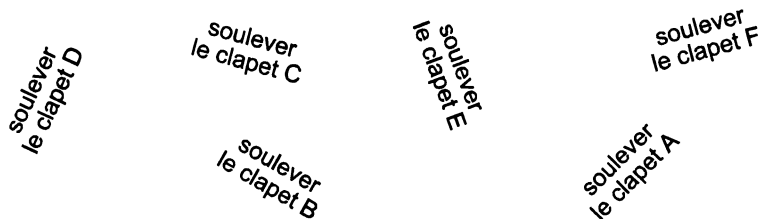


figure B-35

- **Ordre chronologique d'exécution des actions**

Lors d'un cycle de triage donné, un seul clapet se soulève pour satisfaire la fonction *sélectionner*. Selon le type de tube, il peut s'agir de n'importe lequel des clapets. Pour le cas des tubes de type 7, aucun ne se soulève. De ce fait, il n'existe aucune *séquence* d'exécution de toutes ces actions (représenter les étapes associées à ces actions les unes *en-dessous* des autres nuirait à cette idée). C'est pourquoi les étapes sont représentées les unes à côté des autres.

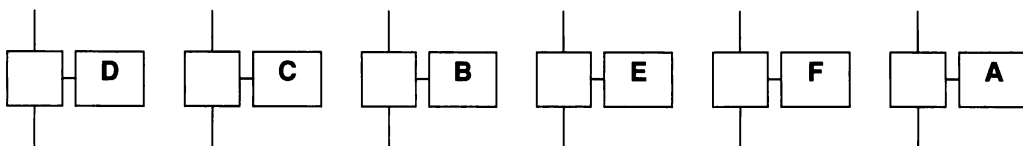


figure B-36

Cette figure représente les actions de la gauche vers la droite dans un ordre A à F quelconque afin de bien mettre en évidence cette idée de non-chronologie.

- **Les informations qui provoquent ces actions**

La sélection du clapet à soulever est directement liée au type de tube en présence, c'est la *condition logique*. Les réceptivités en amont des étapes sont d'écriture immédiate :

- [type=4] pour l'étape associée à l'action D
- [type=3] pour l'étape associée à l'action C, etc.

La *condition temporelle* (à quel moment soulever le clapet) est quant à elle gérée par le grafcet de coordination.

- **Les informations qui permettent l'arrêt des actions**

Un clapet s'abaisse dès que le tube à trier est passé en-dessous de celui-ci, c'est-à-dire sur signal délivré par le détecteur de passage correspondant.

Il n'est peut-être pas inutile de rappeler que la seule fonction étudiée dans ce paragraphe est

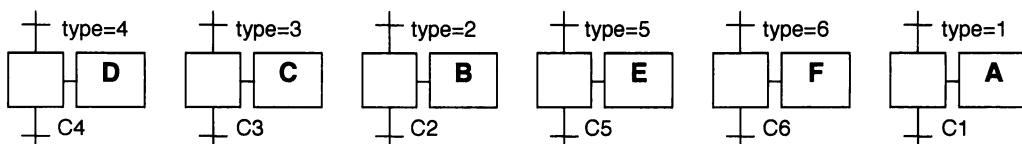


figure B-37

la *sélection* : il ne faut surtout pas tenir compte des mouvements des clapets lorsque ceux-ci interviennent pour réaliser une autre fonction (*transporter* par exemple).

Les réceptivités peuvent maintenant être complétées (figure B-37). On choisira, pour des raisons évoquées plus loin, de tracer un grafcet particulier pour chacune des actions.

- **Remarque quant à la structure du grafcet**

Il eût été possible de prévoir une structure avec divergence en OU (fonction exclusive) et de n'écrire alors qu'un seul grafcet (figure B-38).

Afin de tenir compte du cas particulier du type 7, il convient alors soit de dessiner une branche supplémentaire, qui ne génère aucune action (pour éviter des difficultés de synchronisation avec le grafcet de coordination), soit d'inhiber l'appel de ce grafcet pour le type 7.

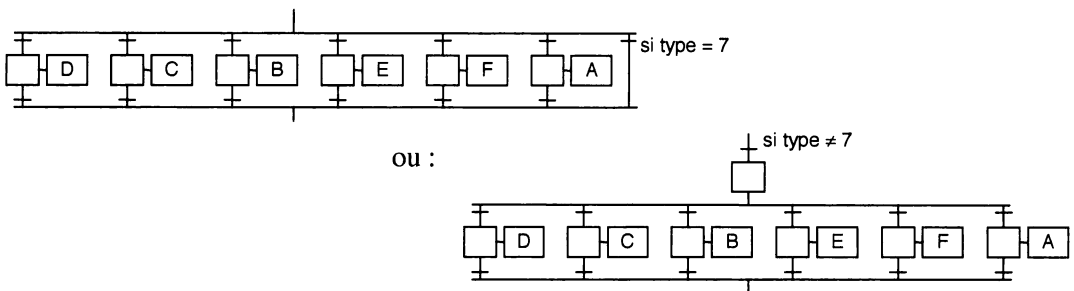


figure B-38

d) Grafcet lié à la fonction transporter

La construction de ce grafcet nécessite également une décomposition fonctionnelle. La même démarche que précédemment nous permet de nouveau de réaliser cette construction par touches successives. Cependant, le problème n'étant pas le même, il faut veiller à ne pas reproduire les réponses d'une manière automatique et irréfléchie.

- **Inventaire des sous-fonctions**

La fonction *transporter* avait été exprimée de la façon suivante : transporter le tube jusqu'à sa goulotte de destination, quelle qu'elle soit. Une première manière de décomposer cette fonction consisterait à considérer individuellement chaque type de tube : transporter un tube de type 1, transporter un tube de type 2, etc.

Une rapide analyse permet de conclure que les 7 séquences qui en découlent reproduiraient les 7 séquences du grafcet de description initial. En fait, cette décomposition n'est pas correcte car elle se réfère directement à la chronologie d'exécution des actions : il ne s'agirait pas de décomposition, mais de description des cycles. Les sous-fonctions ne seraient pas suffisamment isolées de leur contexte.

Un isolement complet consiste à décomposer la fonction *transporter* (qui englobe l'ensemble du processus pour déplacer un tube de son point d'origine vers son point d'arrivée) en *transports élémentaires*. Or, ces fonctions élémentaires consistent à transporter un tube seulement d'une goulotte, quelle qu'elle soit, vers la goulotte supérieure.

Afin d'éviter toute référence à un cycle quelconque, l'inventaire des sous-fonctions se fait de nouveau dans un ordre complètement aléatoire.

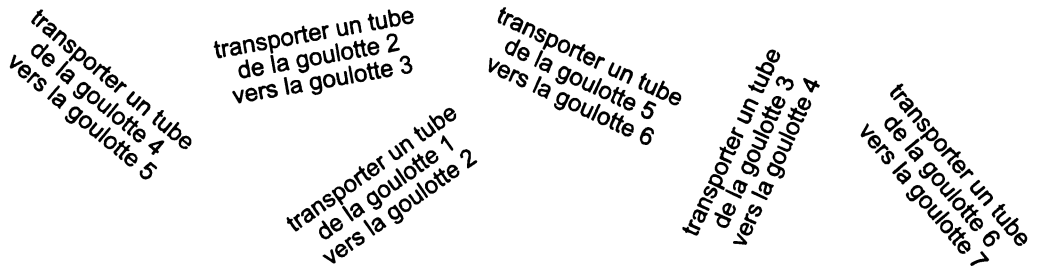


figure B-39

- Inventaire des actions correspondantes

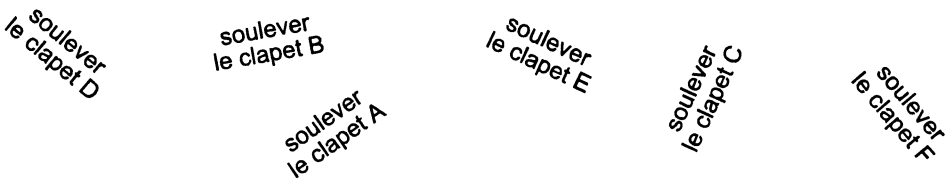


figure B-40

- Ordre chronologique d'exécution des actions

Une première approche, hâtive, peut nous conduire à imposer un séquençage figé dans l'exécution de ces actions, à savoir : d'abord soulever le clapet A, ensuite le clapet B et ainsi de suite. Il faudrait prévoir des sauts d'étapes selon le type de tube à trier. Le grafcet qui en résulterait serait similaire au grafcet de description simplifié, qui avait été rejeté. L'isolement des fonctions serait, là encore, insuffisant.

A condition donc d'isoler correctement les fonctions, on s'aperçoit qu'il est inutile en fait d'imposer une chronologie d'exécution des actions. Les étapes se représentent les unes à côté des autres et également dans le désordre afin de suggérer cette absence d'obligation de séquence.

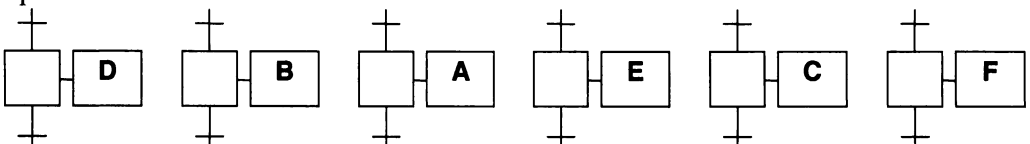


figure B-41

Au contraire du grafcet de la fonction *sélectionner*, il ne faut en aucun cas regrouper ces étapes à l'intérieur d'un même graphe en les associant par une divergence en OU car toutes ou partie de ces étapes sont activées les unes après les autres et restent actives en même temps.

Si un séquençage est pourtant perçu par l'observateur, il n'est dans ce cas qu'apparent. Le système évolue librement, et c'est seulement le déplacement du tube qui ordonne les actions à exécuter par l'intermédiaire des détecteurs de proximité.

- Les informations qui provoquent ces actions

C'est l'arrivée d'un tube sur un clapet abaissé qui provoque le mouvement de ce clapet. Il s'agit d'une *condition temporelle* qui définit l'instant d'exécution d'une action (conditions C1 à C7). A cela s'ajoute une *condition logique* : un clapet donné ne se soulève que pour transporter des tubes de type supérieur (conditions [type > 1] à [type > 6]).

- Les informations qui permettent l'arrêt des actions

Le clapet s'abaisse une fois le tube arrivé sur la goulotte supérieure. Plutôt que par un fin de course du clapet, cette condition peut être vérifiée par le détecteur de passage de la goulotte supérieure. A l'économie des fins de courses, s'ajoute une meilleure adéquation entre le problème et la solution technologique car, au contraire des fins de courses, la détection du tube lui-même répond directement au besoin.

Le grafcet, décomposé en autant de parties que de clapets, peut maintenant être complété par ses réceptivités.

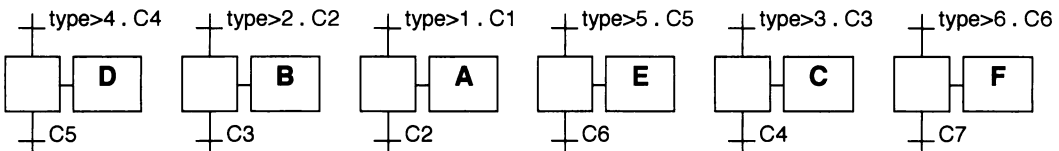


figure B-42

5 - 5. Grafcet du mode de production normale

Si du point de vue didactique il était intéressant de présenter les actions dans le désordre pendant l'étude des fonctions, on préfère les représenter dans l'ordre A à F sur ce document de synthèse (figure B-43 page suivante).

Pour faciliter la lecture des dialogues entre les grafquets, on veille à numéroter les étapes d'une manière judicieuse. En effet, la numérotation adoptée fait à chaque fois référence au type de tube. La rédaction du programme de l'automate ainsi que sa maintenance sera d'autant plus sûre.

6 - A PROPOS DU RESULTAT

a) Comparaison avec le grafcet de description initial

En raison de la simplicité de son interprétation, le grafcet de description initial reste valable tant qu'il s'agit justement de *décrire* le fonctionnement du système. Le nouveau grafcet qui a été obtenu suite à une analyse strictement fonctionnelle est quant à lui très intéressant par deux points :

- lors de la phase de programmation de l'automate,
- pour résoudre des problèmes plus complexes.

- Lors de la phase de programmation

On rappelle que les constructeurs intègrent de plus en plus dans leurs logiciels des fonctions de duplication, que ce soit de listes d'instructions, de portions de grafquets ou même de fichiers complets. Cette tendance doit nous conduire à favoriser la réalisation d'*objets* réutilisables par copie et reconfiguration.

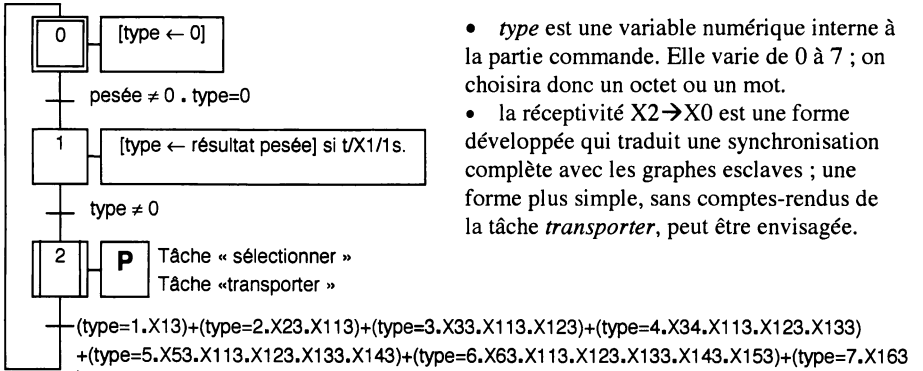
Le grafcet de la figure B-43 illustre très bien cette technique puisqu'elle permet de ne programmer en fait qu'un seul des graphes, tous étant de structure absolument identique et de surcroît tout à fait standards : une étape initiale et deux étapes normales.

- Pour résoudre des problèmes plus complexes

Si une telle méthode peut paraître un peu lourde dans le cas présent qui est assez simple, elle se justifie d'autant mieux que le problème devient plus complexe. Par exemple, si le poste de triage est constitué de 50 goulottes, ou même plus, il est très facile d'étendre le programme sans en augmenter la difficulté.

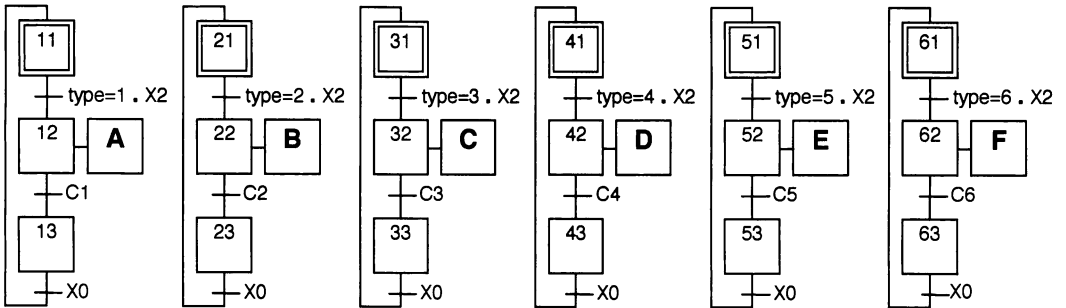
Les troisième et quatrième parties de ce livre traiteront de problèmes pour lesquels le comportement du système doit être flexible, capable de s'adapter à des situations multiples et à des cycles qui ne sont pas figés. C'est surtout dans ce cadre qu'une analyse de ce type montre tout son intérêt.

Coordination des tâches



- *type* est une variable numérique interne à la partie commande. Elle varie de 0 à 7 ; on choisira donc un octet ou un mot.
- la réceptivité $X2 \rightarrow X0$ est une forme développée qui traduit une synchronisation complète avec les graphes esclaves ; une forme plus simple, sans comptes-rendus de la tâche *transporter*, peut être envisagée.

Tâche sélectionner



Tâche transporter

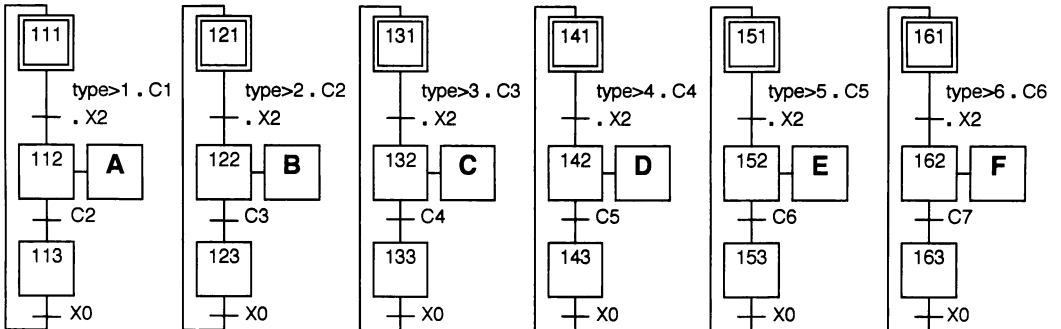
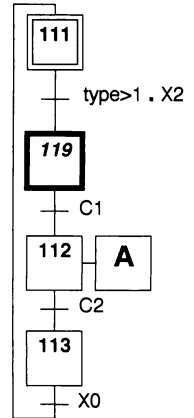


figure B-43

6 - 2. Remarques sur la tâche transporter

L'étape supplémentaire X119 qu'on peut ajouter au grafcet X111 n'améliore certes pas le fonctionnement, mais elle permet de mieux faire la distinction entre *fonction* et *action*.

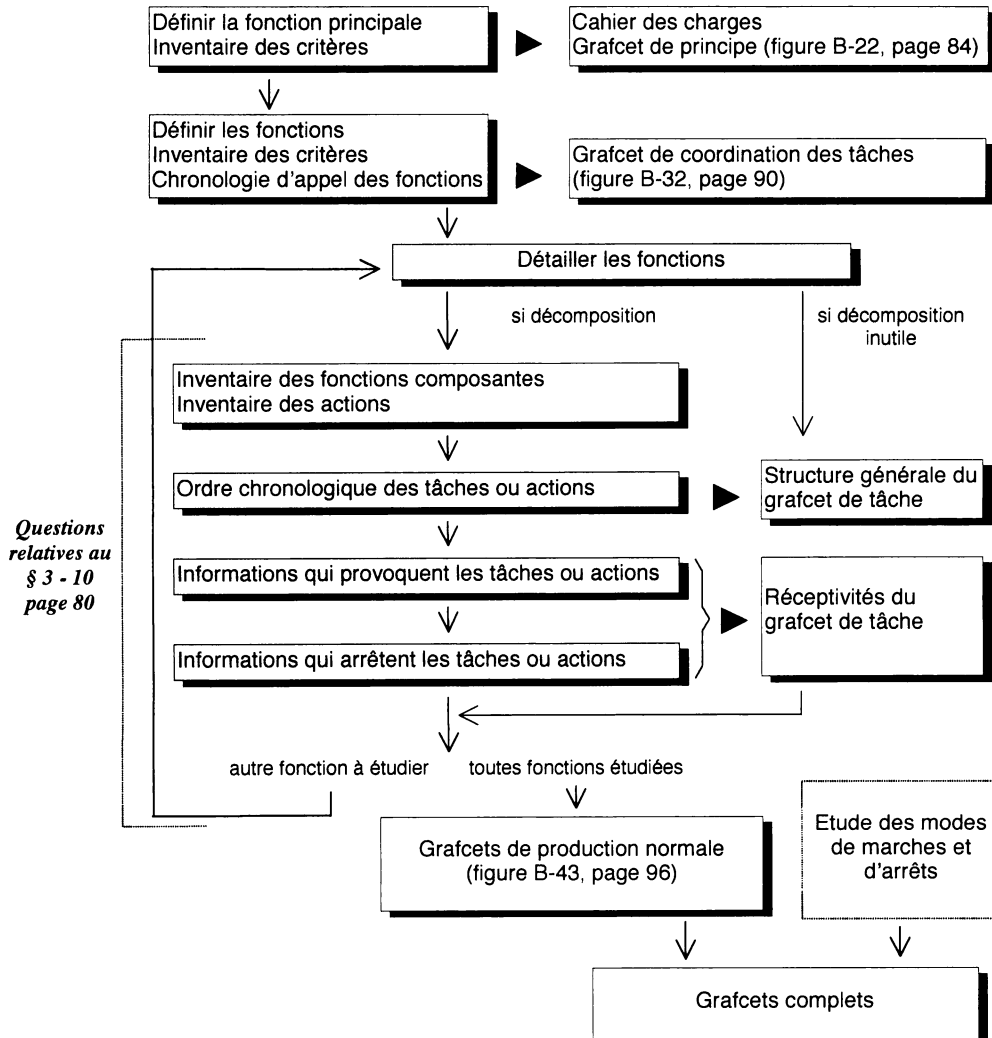
L'étape X119 est activée dès l'apparition de l'ordre de départ de la fonction transporter (étape X2). Mais aucune action n'a lieu. Pour cela, il faut attendre que le tube arrive en C1, l'étape X112 étant activée seulement à ce moment-là. L'action A est alors réalisée.



6 - 3. Résumé de la démarche

Le diagramme ci-dessous rappelle les différentes étapes qui ont été suivies pour développer le grafcet de production normale et il indique la place des modes de marches et d'arrêts.

figure B-44



Questions relatives au § 3 - 10 page 80

figure B-45

7 - RECAPITULATION

En référence à la représentation en VÉ, la figure B-46 montre la place des différents grafquets dans la vie du système *poste de triage*. On notera que le passage d'un *niveau* à un autre ne se limite pas simplement à aménager la notation. La structure même des grafquets peut être mise en cause. Chaque graphisme a son intérêt propre.

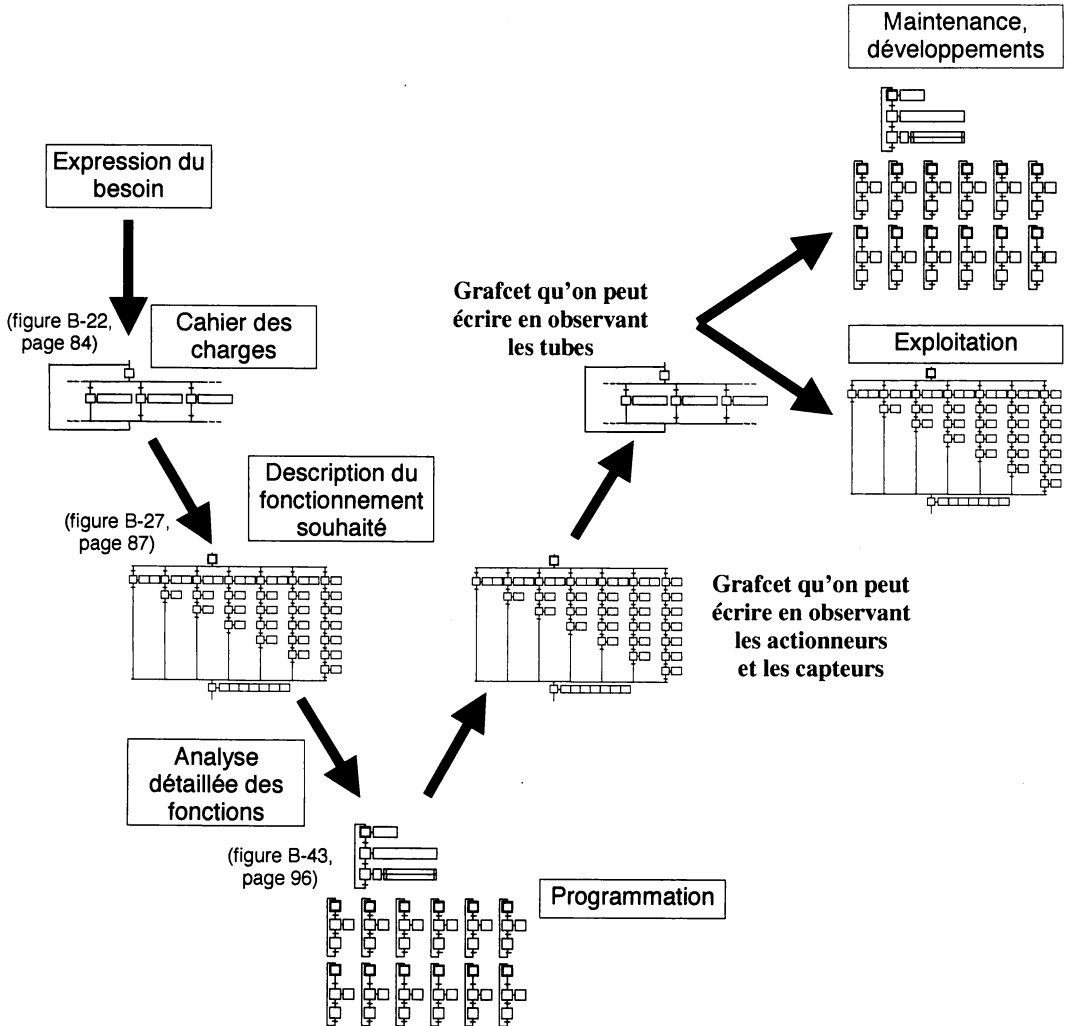


figure B-46

VI . SEQUENCES OBLIGÉES

ETUDE DE CAS N°2 : CELLULE DOUBLE

1 - MISE EN SITUATION

1 - 1. Contexte

Dans l'atelier d'assemblage d'une unité de production de petit matériel électrique (interrupteurs, porte-fusibles, etc.) on envisage l'étude d'une cellule composée de 4 postes manuels d'assemblage et d'un système automatique de convoyage (figure B-47). Les boîtes qui contiennent les composants à assembler proviennent d'un atelier de préparation, non étudié. La première phase de l'assemblage est réalisée soit sur le poste A, soit sur le poste B. Les opérateurs des postes C et D sont ensuite chargés de la seconde phase d'assemblage pour laquelle des pièces complémentaires sont requises. Les postes A et B sont absolument identiques et les opérateurs ont les mêmes compétences : l'assemblage peut donc être réalisé aussi bien par A que par B. Il en est de même pour les postes C et D.

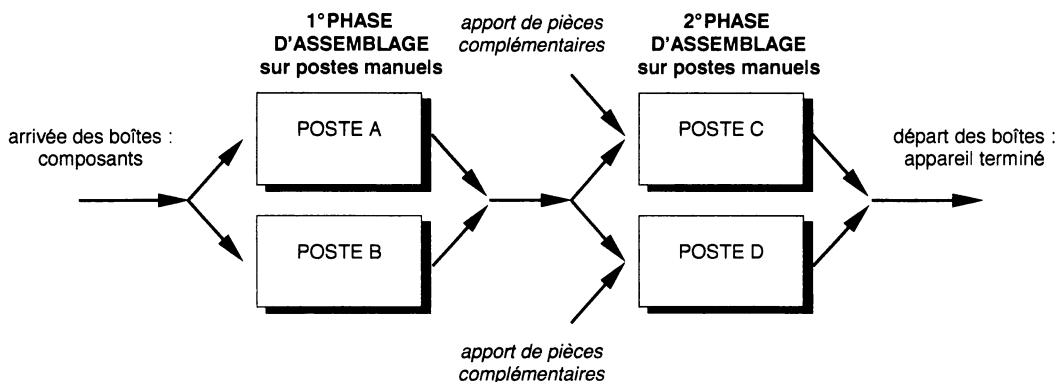


figure B-47

1 - 2. Aménagement des postes de travail

Les 4 postes sont aménagés de façon identique. Les postes C et D sont cependant équipés de casiers pour les articles complémentaires. Des tapis à rouleaux inclinés permettent aux

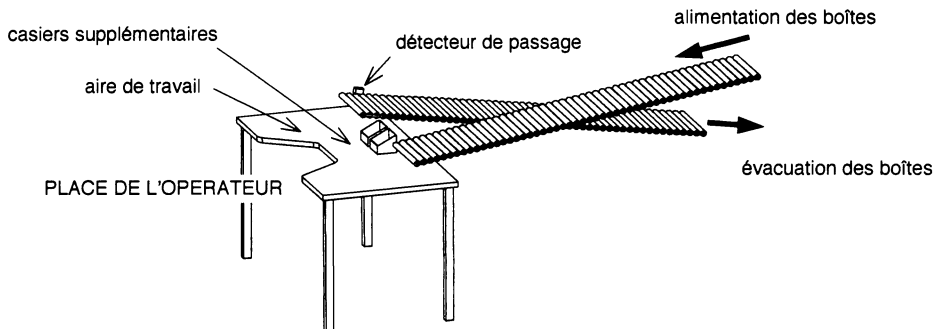


figure B-48

boîtes de circuler par gravité. Elles sont alimentées à *main droite* de l'opérateur et évacuées à *main gauche* sur tous les postes. Le détecteur de passage signale que l'opérateur vient de libérer une boîte de son poste. Il constitue la frontière entre les stocks d'alimentation et d'évacuation (on considère ainsi que la boîte en cours de traitement fait partie du stock d'alimentation).

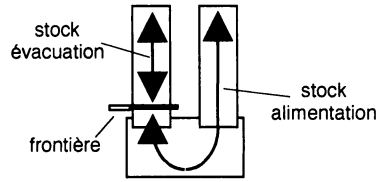


figure B-49

1 - 3. Fonction principale

Il faut distinguer :

- les 4 postes dont la fonction principale est de réaliser la valeur ajoutée sur le produit,
- le système de convoyage qui est chargé d'alimenter et d'évacuer les postes.

La première de ces fonctions étant réalisée manuellement, seule la gestion du flux des boîtes est concernée par l'étude d'automatisation. Cependant, à cause de l'interaction entre l'opérateur et la machine, il ne faut pas négliger l'analyse du comportement de l'opérateur.

1 - 4. Critères

Les critères chiffrés concernent la cadence globale du flux des boîtes, leur taille, leur masse, leur morphologie, etc... ainsi que la taille maximale des stocks et l'encombrement au sol de la cellule : pour satisfaire à ces deux derniers critères, compte tenu de la place disponible dans l'atelier, il est nécessaire de prévoir 3 aires de stockage intermédiaire (appelées transit) entre les deux groupes [A, B] et [C, D].

critère		niveau
dimensions des boîtes		200 x 200, hauteur 70 mm
taille maximale des stocks	à l'entrée de la cellule	6 boîtes
	à l'entrée de chaque poste (amont)	6 boîtes
	à la sortie de chaque poste (aval)	6 boîtes
	à la zone de transit	18 boîtes au total
	à la sortie de la cellule	pas de limite imposée
longueur maximum de la zone de transit		2 mètres

2 - LE CAHIER DES CHARGES TECHNIQUE

2 - 1. Remarque préliminaire au sujet du cahier des charges

Cette étude fait intervenir une grande part de gestes humains dont certains interfèrent avec la machine, à savoir : la prise et la dépose de boîtes au niveau des 4 postes. Ce paragraphe précise comment inclure ces gestes dans le cahier des charges.

Une interprétation erronée du sens du mot *action* peut conduire à une mauvaise utilisation du Grafcet si l'on ne distingue pas correctement la frontière entre l'opérateur et la machine. En effet, les *actions*, dans le sens du Grafcet, sont toujours pilotées par la machine. Une *action réalisée par un opérateur* par contre ne peut pas être associée à une étape de grafcet. Bien au contraire, lorsque l'opérateur réalise l'action *appuyer sur le bouton DCY*, il s'agit d'une information et non d'une action du point de vue de la machine.

• Autrement dit, d'après le Grafcet, les actions de l'opérateur doivent être représentées exclusivement dans les réceptivités (et non dans les actions, figure B-50)

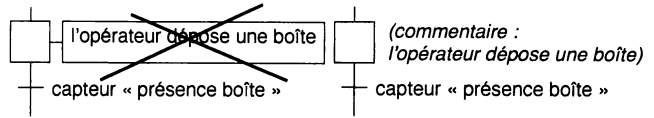


figure B-50

• Une autre solution pourrait consister à considérer l'opérateur comme étant *le système*, la machine représentant alors le *milieu environnant* (figure B-51) ; le Grafcet en tant que modèle théorique peut jouer ce rôle, mais n'est pas approprié dans le contexte de la conception d'une machine automatique à cause des confusions possibles.

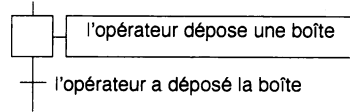


figure B-51

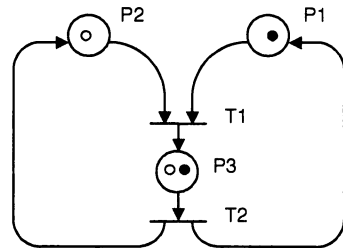
2 - 2. Enoncé du cahier des charges

a) Les 4 postes d'assemblage

Pour la partie de l'étude qui porte plus particulièrement sur les temps d'exécution, on réalise une grille MTM (Movement and Time Measurement) dont les premiers mouvements sont codifiés dans le tableau suivant :

MOUVEMENT MAIN GAUCHE	DIST.	CAS	TEMPS	MOUVEMENT MAIN DROITE	DIST.	CAS	TEMPS
atteindre boîte terminée	16	RA	7.1	atteindre nouvelle boîte	50	RB	13
saisir boîte terminée		G1A	2	saisir nouvelle boîte		G1A	2
mouvoir boîte terminée	50	MB	18	mouvoir nouvelle boîte	50	MB	18
lâcher boîte terminée		RL1	2	lâcher nouvelle boîte		RL1	2
atteindre composant 1	10	RC	8.4	atteindre composant 11	10	RC	8.4
saisir composant 1		G1A	2	saisir composant 11		G1A	2
mouvoir composant 1	6	MC	5.8	mouvoir composant 11	6	MC	5.8
positionner composant 1		P3NS	53.4			P3NS	53.4
lâcher composant 1		RL1	2	positionner composant 11		RL1	2
etc...				lâcher composant 11			
				etc...			

L'aspect dynamique peut également être décrit par un réseau de Petri (figure B-52). Le travail à réaliser pour chaque boîte est décidé par l'opérateur au vu de son contenu. Il prélève les boîtes une à une dans le stock en amont de son poste, réalise le travail adéquat et les dépose dans le stock en aval de son poste. Ce fonctionnement n'est évidemment pas à gérer par le système automatique.



P1 : l'opérateur prélève une nouvelle boîte (main droite ●)
 P2 : l'opérateur dépose la boîte terminée (main gauche ○)
 T1 : l'opérateur a échangé les boîtes
 P3 : l'opérateur réalise la valeur ajoutée
 T2 : le travail sur la boîte en cours est terminé

figure B-52

La suite de l'étude portera exclusivement sur le système de convoyage des boîtes.

b) Le système de convoyage des boîtes

- Contraintes principales

Les ateliers d'assemblage sont traditionnellement conçus de telle sorte que les opérateurs sont astreints à une présence continue à leur poste de travail, faute de quoi, la chaîne complète est bloquée. Pour cette nouvelle étude, le service ergonomie de l'entreprise souhaite assouplir cette contrainte :

- les opérateurs choisiront librement leurs moments de pause (la durée étant cependant limitée),
- les opérateurs travailleront à une cadence personnalisée, qui doit pouvoir être variable (soutenue en début de poste ou pour un travailleur expérimenté et plus lâche en fin de poste ou pour un débutant...)

Le flux des boîtes doit automatiquement s'adapter à ces deux contraintes.

A propos

Réponse

Une telle décision n'est-elle pas dommageable sur le plan de la productivité du fait d'un taux d'occupation moindre ?

Bien au contraire. L'expérience montre que les ateliers équipés de machines ergonomiques affichent des taux d'absentéisme bien moins élevés car le personnel est davantage motivé. Dans l'ensemble, le surcoût dû à cette souplesse ainsi qu'à une esthétique soignée, la présence d'accessoires qui facilitent la vie quotidienne (casier personnel directement sur le poste de travail, supports pour bouteilles d'eau et gobelet...) sont très rapidement rentabilisés, sans compter l'amélioration des conditions de travail.

2 - 3. Les propositions qui découlent du cahier des charges

Deux premières propositions sont directement dûes à la gamme d'assemblage :

Quant au chemin à suivre à l'intérieur de la cellule :

- ◆ chaque boîte, quel que soit son contenu, doit d'abord passer par le poste A, ou bien par le poste B,
- ◆ chaque boîte, quel que soit son contenu, doit ensuite passer par le poste C, ou bien par le poste D.

Les propositions suivantes permettent l'auto-régulation du flux des boîtes :

Quant à la gestion des priorités :

- ◆ les boîtes doivent être déposées en priorité au poste qui en dispose le moins,
- ◆ elles doivent être retirées en priorité au poste qui en dispose le plus.

Ces dernières propositions permettent en effet de réguler la cellule.

- Des regroupements de boîtes se produisent en plusieurs endroits de la cellule. La meilleure fluidité du flux de production est obtenue lorsqu'aucun goulot d'étranglement n'apparaît car la cadence d'ensemble de la cellule est ralentie voire même stoppée quand le nombre de boîtes à l'un des stocks devient excessif. Les niveaux des différents stocks doivent donc être équilibrés.
- L'assouplissement souhaité au niveau de la présence des opérateurs est satisfait. Si l'un des opérateurs s'absente momentanément, le stock en amont de son poste n'est tout naturellement pas renouvelé puisque qu'il n'y a plus de prélèvement sur ce tapis.
- En cas d'interruption momentanée des deux postes A et B simultanément, les boîtes en attente aux transits permettent aux postes C et D de poursuivre le travail. A l'inverse, ce stockage intermédiaire évite la formation immédiate d'un goulot d'étranglement aux sorties de A et B en cas d'interruption simultanée des postes C et D.

3 - CONFIGURATION DE LA CELLULE

3 - 1. Principe technique du système de transport des boîtes

Un détecteur de passage indique l'arrivée d'une nouvelle boîte dans la cellule (figure B-53). Un premier manipulateur est chargé de prélever les boîtes à l'aire d'alimentation de la cellule, d'alimenter et d'évacuer les postes A et B, et de déposer les boîtes sur les tapis à

rouleaux du transit. De même que les tapis des postes, l'alimentation et l'évacuation de la cellule ainsi que le transit sont inclinés pour permettre aux boîtes de se déplacer par gravité. Le second manipulateur prélève les boîtes aux 3 positions basses du transit, gère les postes C et D, et dépose les boîtes sur le tapis d'évacuation de la cellule.

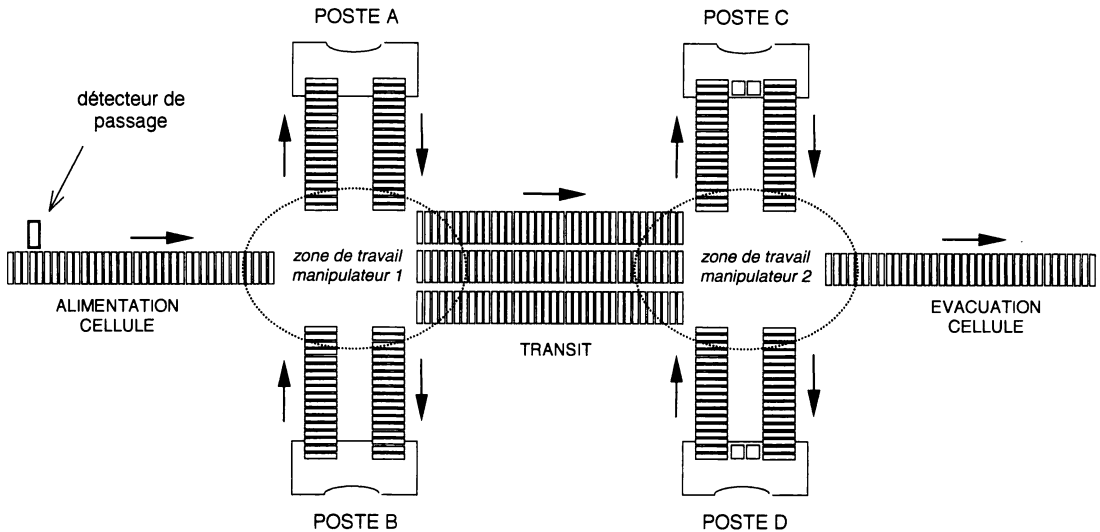


figure B-53 (vue de dessus)

3 - 2. Désignation des détecteurs et des stocks

Pour une écriture plus aisée, on affecte aux différents détecteurs et aussi au nombre de boîtes présentes dans chaque stock les symboles définis dans le tableau de la figure B-54. On associe un compteur à chaque stock.

	symbole associé au détecteur	symbole associé à la valeur du stock (compteur)
alimentation cellule	D_a (détecteur à l'alimentation de la cellule)	Ca (nombre de boîtes à l'alimentation de la cellule)
alimentation poste i	/	Ca_i (nombre ... à l'aire d'alimentation du poste i)
évacuation poste i	D_i (détecteur au poste i)	$Cé_i$ (nombre ... à l'aire d'évacuation du poste i)
transit j	/	CT_j (nombre ... sur le transit j)
évacuation cellule	/	/

i varie de A à D

j varie de 1 à 3

figure B-54

Aucun détecteur supplémentaire ne sera requis.

3 - 3. Décomposition en 2 sous-ensembles

L'automatisation du système de transport ne concerne que les 2 manipulateurs. Puisque chaque manipulateur entre en service en fonction des niveaux de stocks, leur synchronisation se fait exclusivement par l'intermédiaire du comptage-décomptage des boîtes en attente sur le transit. Aussi sont-ils totalement indépendants : l'étude porte dans un premier temps sur le manipulateur n°1. Un résumé pour le second manipulateur sera présenté dans un deuxième temps.

3 - 4. Architecture du système

Une partie de l'intelligence du système est délocalisée dans les commandes des manipulateurs : les tâches élémentaires (ou sous-programmes). Cette décomposition est en fait le résultat d'une première analyse fonctionnelle qui concerne tant les tâches que la structure matérielle du système. Les sous-programmes des manipulateurs ne peuvent pas être détaillés tant que leurs solutions technologiques ne sont pas adoptées. Mais l'énoncé global de leurs fonctions suffit amplement à la poursuite de l'étude d'automatisation.

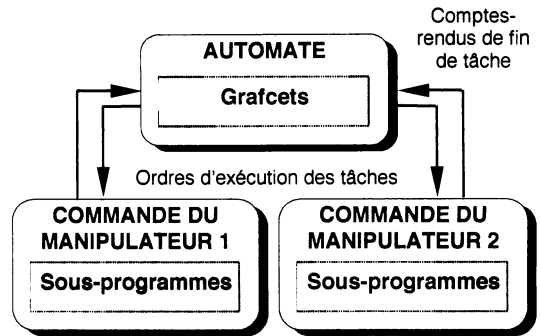


figure B-55

MANIPULATEUR 1		MANIPULATEUR 2	
STA	Saisir au Tapis d'Alimentation	DTE	Déposer au Tapis d'Evacuation
DA	Déposer au poste A	DC	Déposer au poste C
SA	Saisir au poste A	SC	Saisir au poste C
DB	Déposer au poste B	DD	Déposer au poste D
SB	Saisir au poste B	SD	Saisir au poste D
DT1	Déposer au Transit 1	ST1	Saisir au Transit 1
DT2	Déposer au Transit 2	ST2	Saisir au Transit 2
DT3	Déposer au Transit 3	ST3	Saisir au Transit 3

Le sous-programme *DA* par exemple est chargé :

- de positionner l'organe de préhension du manipulateur au-dessus du point *haut* du tapis d'alimentation du poste A,
- d'ouvrir ensuite la pince,
- puis d'attendre un nouvel ordre.

Remarque : ce positionnement se fait à partir de n'importe quel point où pourrait se trouver la pince, il ne s'agit donc pas réellement d'une trajectoire.

4 - DESCRIPTION DU MANIPULATEUR N°1

4 - 1. Approche traditionnelle

Une approche globale du cahier des charges conduit à l'ébauche de la figure B-56-b. Ce grafcet a été développé en suivant la démarche traditionnelle qui consiste à représenter les étapes dans l'ordre d'exécution du cycle. Probablement que 2 temps sont nécessaires :

- Tracer le grafcet en suivant par la pensée le trajet d'une boîte (figure B-56-a)
 - ⇒ *On s'aperçoit qu'après avoir déposé une boîte à l'un des postes, le manipulateur n'a d'autre possibilité que d'attendre de pouvoir saisir une boîte à ce même poste. Ses services peuvent cependant être requis soit au tapis d'alimentation soit au poste B.*
- L'inventaire exhaustif des possibilités d'évolution supplémentaires permet ensuite de compléter le grafcet par adjonction de transitions et de liaisons.
 - ⇒ *Le résultat donne une bonne image de la complexité du fonctionnement mais la structure du grafcet n'est pas très parlante, ce qui devrait pourtant être la caractéristique essentielle d'une représentation graphique.*

Le grafcet de la figure b peut également être tracé directement en analysant à chaque étape les différentes possibilités d'évolution.

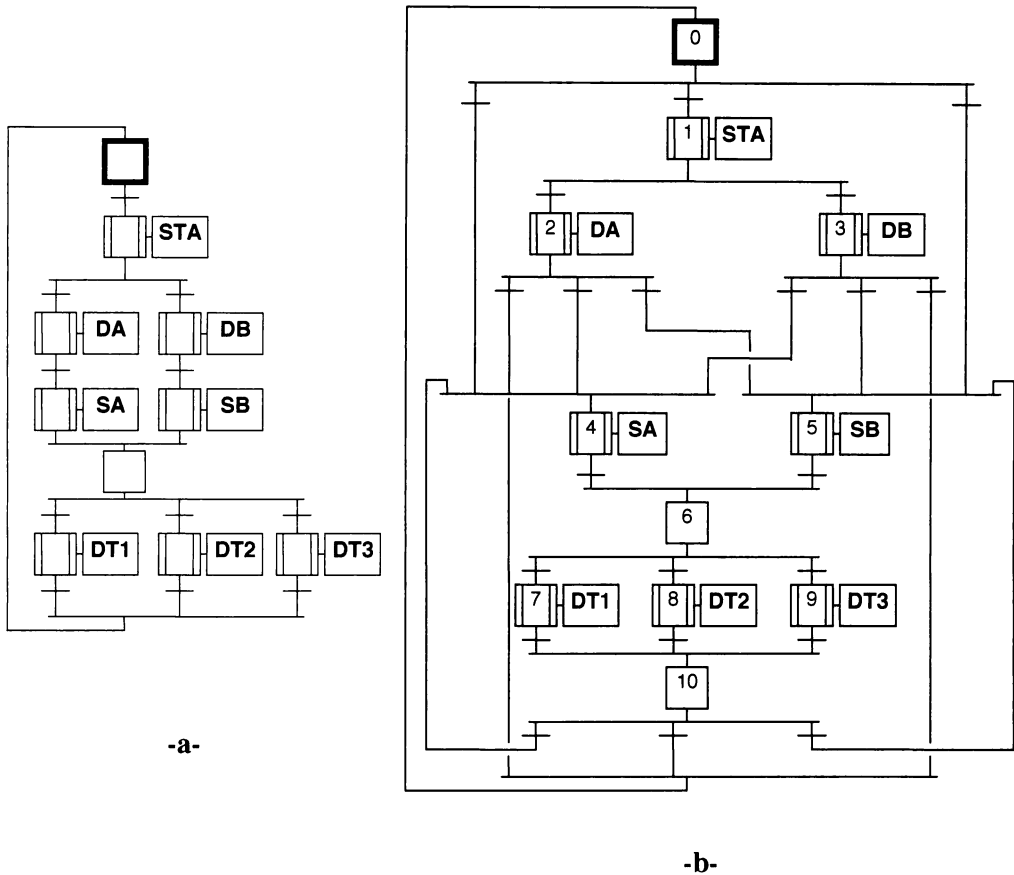


figure B-56

4 - 2. Réceptivités

Les réceptivités mettent en relation les états des différents niveaux de stocks et comportent en outre des conditions de priorité. Certaines parties du grafcet sont redondantes : par exemple les liaisons en aval de X10 et de X2 rejoignent toutes les deux les étapes 4, 5 et 0. Une question se pose : les réceptivités respectives sont-elles les mêmes ? Cette analyse supplémentaire est source d'erreur potentielle.

4 - 3. Analyse des fonctions mal maîtrisée

Si l'on tire des conclusions hâtives de l'étude du poste de triage et qu'on essaie d'appliquer directement son résultat, le morcellement des grafcets comme représenté figure B-57 sera exagéré car en réalité, et contrairement au poste de triage, l'analyse qui suit montre que certaines séquences ont forcément lieu : il s'agit en fait de choisir le bon niveau

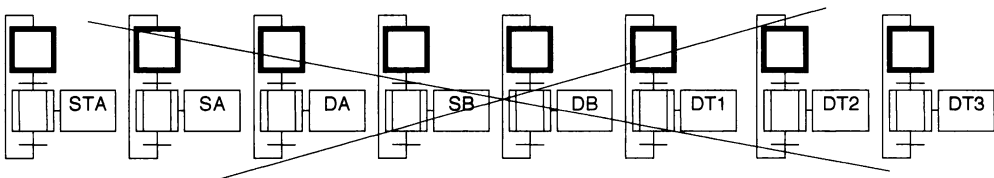


figure B-57

d'isolement des fonctions selon le problème traité.

Le tracé développé de ces séquences permettra en outre de limiter les réceptivités au strict nécessaire (voir également § 3 - 10, figure B-19, page 81).

4 - 4. Approche selon la méthode proposée

Seul le manipulateur 1 est concerné dans un premier temps. L'étude porte sur la commande générale du système, à savoir l'automate, maître de la cellule (figure B-55). Les fonctions qu'il réalise constituent les ordres d'exécution des sous-programmes du manipulateur et permettent ainsi de gérer les stocks.

a) Fonctions relatives aux sous-programmes du manipulateur

La fonction principale peut être exprimée ainsi :

alimenter ou évacuer les postes

La décomposition en fonctions composantes, que ce soit pour *alimenter* ou pour *évacuer*, se fait en deux temps :

- saisir
- déposer

L'analyse de ces fonctions montre qu'une chronologie apparaît nécessairement dans leur exécution : une saisie est obligatoirement suivie d'une dépose ; une dépose est obligatoirement suivie d'une saisie (cette proposition est tellement évidente qu'on pourrait négliger de l'exprimer et même d'en tenir compte pendant le développement du grafcet). Il en découle une première épure (figure B-58).

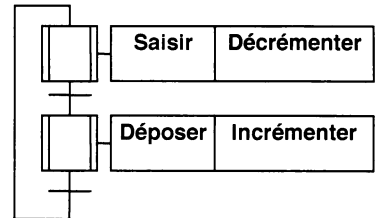


figure B-58

b) Fonctions liées à la gestion des stocks

• Ces fonctions concernent les évolutions des différents compteurs, images des niveaux de stocks. A chaque saisie correspond une décrémentation, à chaque dépose correspond une incrémentation (figure B-58).

• Les stocks évoluent par ailleurs lorsqu'un opérateur engage une boîte sur le tapis d'évacuation de son poste. A l'instant où un signal apparaît au détecteur D_i , le compteur Ca_i doit être décrémentation d'une unité, et le compteur $Cé_i$ doit être incrémentation d'une unité. Cette opération est totalement indépendante du grafcet.

4 - 5. Structure du grafcet

a) Les deux phases

Dans le respect du grafcet de la figure B-58, on représente les fonctions composantes suivant deux phases : une phase *saisir*, et en-dessous une phase *déposer*. Cette mise en place traduit l'aspect systématique de cette séquence (figure B-59).

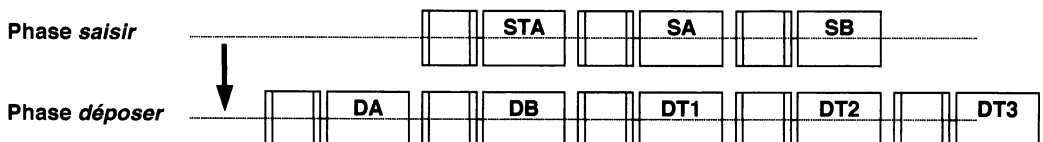


figure B-59

b) Les séquences obligées

Les possibilités d'évolution de la première vers la seconde phase sont limitées. En effet, après avoir saisi une boîte au tapis d'alimentation, cela n'a aucun sens de la déposer immédiatement sur l'un des tapis du transit. Il n'y a donc pas lieu de représenter la liaison entre les étapes X_a et par exemple X_b . L'inventaire de ces limitations conduit à développer une structure de grafcet parfaitement lisible. La figure B-60 illustre de manière immédiate le

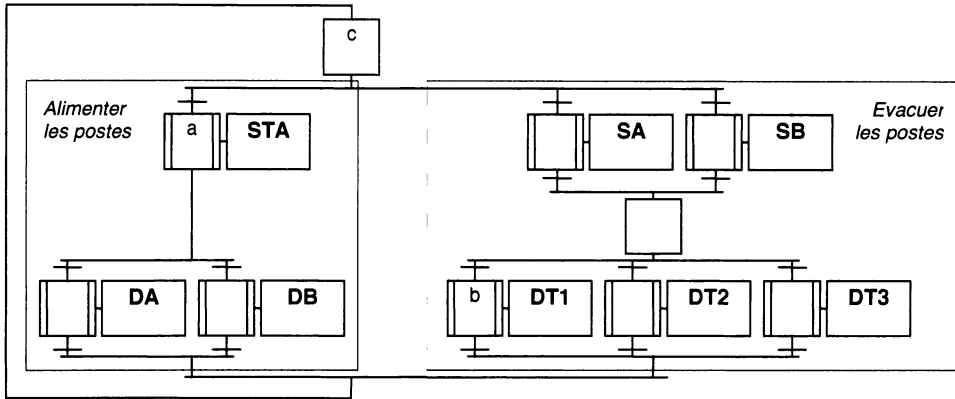


figure B-60

fonctionnement du manipulateur. En particulier, l'étape X_c (qui pourra être l'étape initiale) est une *étape ressource* à partir de laquelle le manipulateur est rendu *disponible* pour accomplir toute nouvelle tâche en fonction du besoin. Par ailleurs, on retrouve aisément la fonction principale : *alimenter* (partie gauche du grafcet) ou *évacuer les postes* (partie droite du grafcet). Dans un souci de clarté, la gestion des compteurs n'est pas représentée sur cette figure.

4 - 6. Détail des évolutions des compteurs

a) Comptage-décomptage

Les évolutions des compteurs sont de deux natures :

- celles liées au fonctionnement séquentiel, pour lesquelles on associe les évolutions des compteurs aux étapes de grafcet comme exprimé par la figure B-58 :

$$\begin{aligned} X_{STA} &: [Ca \leftarrow Ca - 1] \\ X_{SA} &: [Cé_A \leftarrow Cé_A - 1] \\ X_{SB} &: [Cé_B \leftarrow Cé_B - 1] \\ X_{DA} &: [Ca_A \leftarrow Ca_A + 1] \\ X_{DB} &: [Ca_B \leftarrow Ca_B + 1] \\ X_{DT1} &: [CT_1 \leftarrow CT_1 + 1] \\ X_{DT2} &: [CT_2 \leftarrow CT_2 + 1] \\ X_{DT3} &: [CT_3 \leftarrow CT_3 + 1] \end{aligned}$$

La décrémentation des compteurs CT_i se fait lorsque le manipulateur n°2 prélève des boîtes sur le transit. Ces fonctions apparaîtront donc sur le grafcet de gestion de ce manipulateur.

- celles liées au fait que l'opérateur engage une boîte devant le détecteur D_A (ou D_B). Ces équations purement combinatoires complètent le traitement séquentiel :

$$\begin{aligned} [Ca_A \leftarrow Ca_A - 1] \text{ si } \uparrow D_A & & [Ca_B \leftarrow Ca_B - 1] \text{ si } \uparrow D_B \\ [Cé_A \leftarrow Cé_A + 1] \text{ si } \uparrow D_A & & [Cé_B \leftarrow Cé_B + 1] \text{ si } \uparrow D_B \end{aligned}$$

b) Initialisation des compteurs

Dans cet exemple, l'initialisation ne se fait pas sur une étape de grafcet. Les valeurs d'initialisation ne sont pas obligatoirement 0. En réalité la valeur initiale de chaque compteur doit être conforme au niveau de stock qu'il représente au moment du démarrage de la machine. D'ailleurs, des interventions manuelles pourront probablement perturber l'évolution normale des stocks : on choisira une interface utilisateur qui permet de forcer les compteurs aux valeurs adéquates lorsque cela s'avère nécessaire.

4 - 7. Les réceptivités

a) Les conditions qui permettent les *saisies*

- Conditions élémentaires

L'écriture de ces conditions doit prendre en compte l'ensemble des contraintes. A ce niveau de détail, il est cependant probable que toutes n'aient pas été expressément formulées dans le cahier des charges ce qui nécessite une analyse attentive.

En partant de la contrainte principale, le modèle sera enrichi au fur et à mesure.

- Pour que le manipulateur saisisse une boîte sur le tapis d'alimentation (sous-programme STA), il faut que le nombre de boîtes en attente sur ce tapis soit supérieur au nombre de boîtes en attente sur chacun des tapis d'évacuation des postes de travail. La condition d'évolution élémentaire vers l'étape qui appelle la tâche STA est donc :

$$\text{Condition vers STA : } (Ca > Cé_A) \cdot (Ca > Cé_B)$$

- La même contrainte, cette fois appliquée aux postes de travail, conduit à l'écriture des conditions d'évolution élémentaires vers les étapes qui appellent les tâches SA et SB :

$$\text{Condition vers SA : } (Cé_A > Ca) \cdot (Cé_A > Cé_B)$$

$$\text{Condition vers SB : } (Cé_B > Ca) \cdot (Cé_B > Cé_A)$$

- Enrichissement des conditions

Pour les cas où au moins 2 de ces stocks comportent la même quantité de boîtes, le système peut se bloquer. La levée de ce blocage se fait en définissant un ordre de priorité. La modification de la première des conditions (comme indiqué ci-dessous) permet de favoriser la tâche STA dans le but de maximiser le taux d'occupation de la cellule.

$$\text{Condition vers STA : } (Ca \geq Cé_A) \cdot (Ca \geq Cé_B)$$

$$\text{Condition vers SA : } (Cé_A > Ca) \cdot (Cé_A \geq Cé_B)$$

$$\text{Condition vers SB : } (Cé_B > Ca) \cdot (Cé_B > Cé_A)$$

La modification de la seconde condition (ou bien de la troisième) favorise l'un des postes A ou B (A dans ce cas). Ce choix n'a pas d'importance *a priori*, mais on peut envisager une sélection par commutateur pour favoriser momentanément l'un ou l'autre poste. Les conditions, non représentées sur le grafcet complet, seraient alors du type :

$$\text{Condition vers SA : } (Cé_A > Ca) \cdot \{(Cé_A \geq Cé_B) \cdot \text{Commut} + (Cé_A > Cé_B) \cdot /\text{Commut}\}$$

$$\text{Condition vers SB : } (Cé_A > Ca) \cdot \{(Cé_B > Cé_A) \cdot \text{Commut} + (Cé_B \geq Cé_A) \cdot /\text{Commut}\}$$

- La condition d'évolution vers STA doit être inhibée lorsqu'aucune boîte n'est présente sur le tapis d'alimentation (en particulier si les stocks d'évacuation des postes sont également vides) :

$$\text{Condition vers STA : } (Ca \geq Cé_A) \cdot (Ca \geq Cé_B) \cdot (Ca \neq 0)$$

La nature de la condition d'évolution vers SA nécessite la même précaution :

$$\text{Condition vers SA : } (Cé_A > Ca) \cdot (Cé_A \geq Cé_B) \cdot (Cé_A \neq 0)$$

- La longueur des tapis d'alimentation des postes de même que les tapis du transit ne permettent de stocker qu'un nombre limité de boîtes, défini par les critères du cahier des charges : la saisie au tapis d'alimentation de la cellule ne peut avoir lieu que si au moins l'un des postes A et B n'est pas saturé ; de même, les saisies aux postes de travail ne doivent être autorisées que si au moins l'un des transits n'est pas saturé.

$$\text{Condition vers STA : } (Ca \geq Cé_A) \cdot (Ca \geq Cé_B) \cdot (Ca \neq 0) \cdot \{(Ca_A < 6) + (Ca_B < 6)\}$$

$$\text{Condition vers SA : } (Cé_A > Ca) \cdot (Cé_A \geq Cé_B) \cdot (Cé_A \neq 0) \cdot \{(CT_1 < 6) + (CT_2 < 6) + (CT_3 < 6)\}$$

$$\text{Condition vers SB : } (Cé_B > Ca) \cdot (Cé_B > Cé_A) \cdot \{(CT_1 < 6) + (CT_2 < 6) + (CT_3 < 6)\}$$

b) Les conditions qui permettent les *déposes*

L'analyse est plus simple du fait que la seule vérification nécessaire, une fois une boîte saisie, concerne les niveaux des stocks. Il reste à choisir les priorités en accord avec les options retenues précédemment. Les conditions s'écrivent donc immédiatement :

$$\text{Condition vers DA : } (Ca_A \leq Ca_B)$$

$$\text{Condition vers DB : } (Ca_B < Ca_A)$$

$$\text{Condition vers DT1 : } (CT_1 \leq CT_2) \cdot (CT_1 \leq CT_3)$$

$$\text{Condition vers DT2 : } (CT_2 < CT_1) \cdot (CT_2 \leq CT_3)$$

$$\text{Condition vers DT3 : } (CT_3 < CT_1) \cdot (CT_3 < CT_2)$$

Remarque : toutes ces réceptivités peuvent encore évoluer tout au long de l'exploitation de la cellule si l'on souhaite modifier des ordres de priorité, permettre des évolutions temporaires qui obéissent à des règles particulières, réguler différemment les taux d'occupation des postes, etc.

c) Grafcet complété

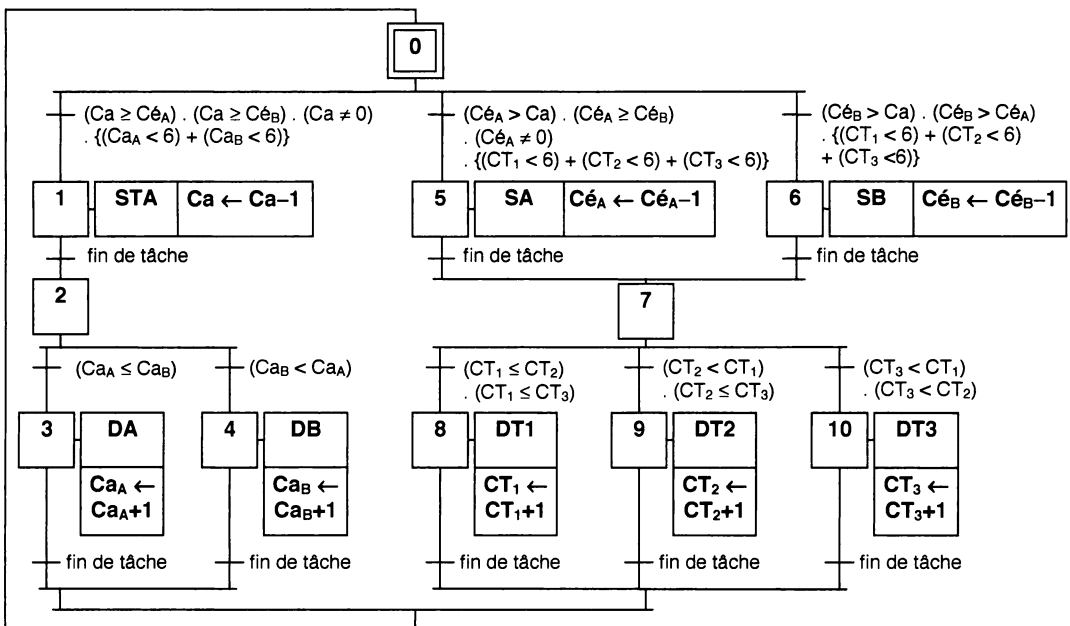


figure B-61

4 - 8. Solutions technologiques de la liaison automate-manipulateur

La liaison physique entre l'automate et la commande du manipulateur peut être réalisée de diverses manières et conditionne la nature des actions *STA*, *SA*, etc. du grafcet au niveau *automate*.

a) Liaison par entrées-sorties TOR

- Dans le sens maître-esclave

On peut relier un certain nombre de sorties de l'automate à des entrées de la commande du manipulateur et affecter à chaque point de contact un message différent. Dans cet exemple, les 8 tâches nécessiteraient 8 E/S. On peut limiter le nombre d'entrées-sorties en codant les messages. Le nombre maximum de messages est alors de $2^{\text{nombre de conducteurs}} - 1$ (l'un des codes devant n'appeler aucune tâche). Cette solution est économique si le nombre total de messages différents à transmettre est limité. Bien que rudimentaire, elle est souvent mise en oeuvre sur des systèmes industriels complexes et est bien adaptée au cas présent.

- Dans le sens esclave-maître

Un seul conducteur suffit à transmettre l'information de fin de tâche, quelle qu'elle soit, la discrimination de la tâche n'étant pas utile. S'agissant d'une seule et même information pour toutes les réceptivités concernées, il convient d'en évaluer le front au risque de rendre instables les étapes d'appels de tâches. Les tâches étant celles du manipulateur, constituant distinct de l'automate, il ne s'agit pas de *tâches dans le sens du Grafcet*. C'est pourquoi les étapes de la figure B-61 (ainsi que de la figure B-62 au § 5 -) sont toutes représentées par des carrés simples. Sur le grafcet au niveau automate, les actions notées *STA...* seront décomposées pour former les codes à émettre vers la commande du manipulateur ; de même, l'information *fin de tâche* sera directement formée par l'entrée TOR correspondante.

b) Liaison série - liaison réseau

Ces solutions sont à envisager lorsque le flux d'informations est plus important comme dans l'étude de cas de la partie C.

5 - RESUME POUR LE MANIPULATEUR 2

5 - 1. Structure du grafcet et actions associées

Le grafcet de commande du manipulateur 2 (figure B-62) est similaire à celui du manipulateur 1. Sa structure générale fait également apparaître :

- les deux niveaux *saisir* et *déposer*,
- une étape ressource,
- l'organisation des fonctions

Du fait qu'elle intègre les séquences obligées, cette structure facilite la recherche des réceptivités : en aval des étapes, ce sont les comptes-rendus de fin de tâche ; en amont des étapes, elles sont exclusivement liées aux niveaux des stocks.

Les actions découlent directement de l'analyse des fonctions. La mise en place des compteurs ne pose pas de problème particulier.

5 - 2. Les réceptivités

On conseille de composer les réceptivités tout aussi progressivement que pour la première partie de l'étude :

- Une première discrimination dans la sélection des séquences consiste à comparer les niveaux de stocks grâce à des inégalités strictes, compteur par compteur.
- En cas d'égalité, 2 niveaux de priorité se dégagent :
 - sélectionner la saisie aux transits plutôt qu'aux postes pour favoriser le taux d'occupation de la cellule,
 - donner la priorité à l'un des postes, donner la priorité à l'un des transits.
- Pour certaines conditions, vérifier si les stocks ne sont pas vides.
- Avant d'engager une saisie, vérifier qu'au moins l'une des destinations possibles n'est pas saturée.

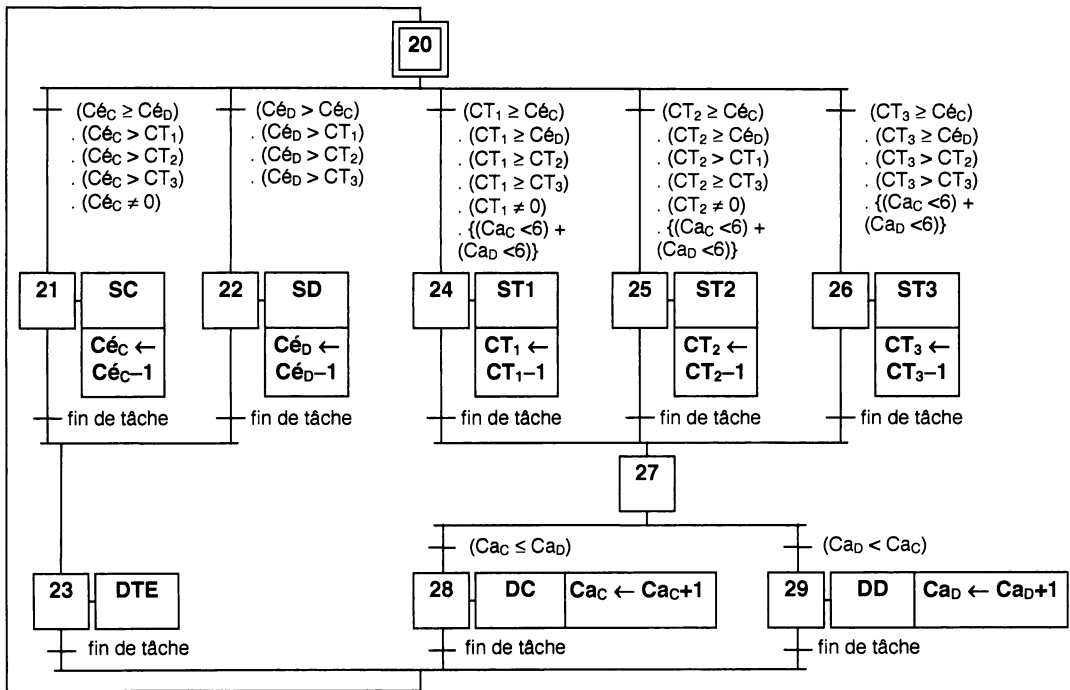


figure B-62

La synchronisation entre les deux manipulateurs se fait uniquement par l'intermédiaire des compteurs relatifs aux transits.

6 - VERIFICATION DES RECEPTIVITES

La justesse des réceptivités peut être vérifiée par simulation de plusieurs cas de figure :

- affecter à chaque stock une valeur quelconque (en particulier les valeurs limites),
- imaginer le comportement logiquement adopté par le système,
- résoudre toutes les réceptivités pour les valeurs choisies,
- imaginer l'évolution du grafcet,
- vérifier la cohérence et réitérer l'essai.

Les valeurs choisies et particulièrement leur combinaison doivent évidemment répondre aux hypothèses de départ.

7 - RECAPITULATION DES EQUATIONS DES COMPTEURS

Le recensement de toutes les conditions d'évolution des compteurs permet de dresser le tableau récapitulatif de la figure B-63.

Certaines conditions sont séquentielles (X_i), d'autres sont combinatoires (D_i).

	Nom compteur	Incrémentation	Décrémentation
Alimentation cellule	C_a	$\uparrow D_a$	$\uparrow X_1$
Poste A	C_{aA}	$\uparrow X_3$	$\uparrow D_A$
	$C_{éA}$	$\uparrow D_A$	$\uparrow X_5$
Poste B	C_{aB}	$\uparrow X_4$	$\uparrow D_B$
	$C_{éB}$	$\uparrow D_B$	$\uparrow X_6$
Transit	CT_1	$\uparrow X_8$ (manipulateur 1)	$\uparrow X_{24}$ (manipulateur 2)
	CT_2	$\uparrow X_9$ (manipulateur 1)	$\uparrow X_{25}$ (manipulateur 2)
	CT_3	$\uparrow X_{10}$ (manipulateur 1)	$\uparrow X_{26}$ (manipulateur 2)
Poste C	C_{aC}	$\uparrow X_{28}$	$\uparrow D_A$
	$C_{éC}$	$\uparrow D_A$	$\uparrow X_{21}$
Poste D	C_{aD}	$\uparrow X_{29}$	$\uparrow D_B$
	$C_{éD}$	$\uparrow D_B$	$\uparrow X_{22}$

figure B-63

8 - REMARQUES AU SUJET DU RESULTAT

Le grafcet proposé figure B-61 ne comporte pas de liaisons redondantes comme le tracé empirique de la figure B-56-b. L'analyse des réceptivités est donc plus immédiate.

Par ailleurs, ce grafcet peut très facilement être complété par des modules supplémentaires si un plus grand nombre de postes doit être géré : la structure générale reste de même qualité de lisibilité et les réceptivités sont complétées sans nouvelle difficulté. (La structure en arborescence du premier tracé par contre conduirait à un grafcet de plus en plus illisible et complexe.)

Le grafcet retenu ne décrit plus la totalité du cycle tel qu'il se déroule pendant le fonctionnement. Le grafcet propose seulement des possibilités d'évolutions, autrement dit, des séquences élémentaires, exécutées selon l'évolution du besoin en réponse à des conditions établies dans les réceptivités.

Le travail finalement réalisé par le système n'est en fait pas déterminé à l'avance lors de la conception du grafcet : il résulte de la combinaison des règles générales de fonctionnement et de l'occurrence des événements. Il ne se répète pas de façon cyclique.

VII . IDEES DIRECTRICES

1 - LE CONTEXTE INDUSTRIEL

1 - 1. L'organisation de l'entreprise

A des degrés divers et en fonction du domaine d'activité de l'entreprise, la *flexibilité des moyens de production* est couramment recherchée pour la raison essentielle d'assurer une rotation rapide des capitaux, grâce en particulier à la diminution des stocks. Cette préoccupation centrale a une incidence capitale sur l'organisation de l'entreprise et les méthodes de travail, et par conséquent sur les technologies employées.

a) L'intégration des moyens

L'intégration du système de production est obtenue en interconnectant l'ensemble des composantes de l'entreprise. Elle peut concerner tout le cheminement des informations : de la commande du client jusqu'à la livraison. Elle assure la possibilité d'une gestion serrée et permet ainsi la réduction des coûts matière, outillage et fournitures tout en garantissant la constance dans l'obtention de la qualité. Les domaines de compétence du personnel qui assure le pilotage des systèmes de production automatisée s'orientent de ce fait vers une plus grande diversité et une technicité accrue.

b) Les méthodes

La codification des moyens et des produits, un système unique et interchangeable de palettes, des systèmes de corrections automatiques et des méthodes de changement rapide de campagne sont adoptés. Cet effort est fait du point de vue des mécanismes (remplacement des outillages...) ainsi que du point de vue des logiciels de conduite. On recherche également une certaine autonomie de l'outil de production et la possibilité de son adaptation à l'évolution du marché dans des délais très brefs. Par ailleurs, la simplification du processus de fabrication est prise en compte dès le stade de la conception des produits.

1 - 2. Flux physiques de production

L'organisation de la production peut être envisagée selon deux points de vue distincts :

- le point de vue du *poste de travail*,
- le point de vue de la *pièce ouvragée*.

Dans un système de production traditionnel,

- un *poste de travail* réalise la même tâche tout au long de la journée (des changements de série interviennent naturellement, mais à un rythme faible),
- toutes les *pièces* d'un type donné suivent le même chemin dans l'atelier.

Dans un système flexible,

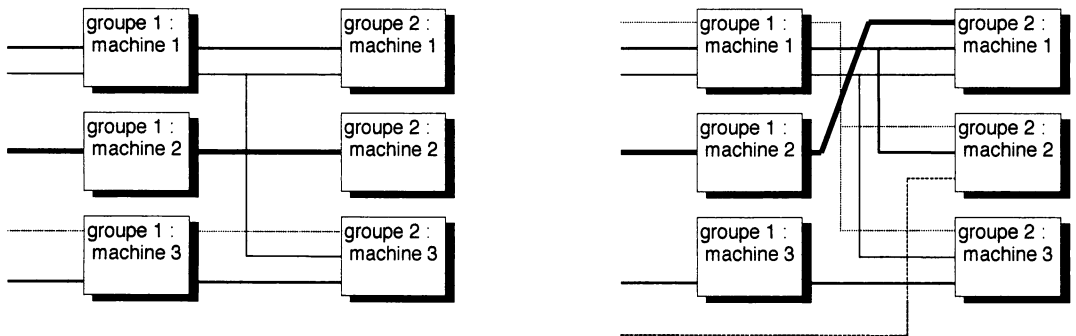
- un poste de travail donné peut effectuer une tâche différente à chaque nouvelle pièce - ou petite série de pièces - en respectant un temps de changement extrêmement bref : la reconnaissance automatique juste en amont du poste permet alors une adaptation automatique ainsi qu'une intégration informatique de la gestion de production,

- les pièces d'un même type ne suivent pas toutes le même chemin dans l'atelier : à chaque nouveau stade de son évolution, un certain nombre de conditions orientent, en temps réel, la pièce vers le poste suivant le plus favorable.

Exemple (Une solution à ce problème est proposée page 127)

Une cellule de production est composée de deux groupes de trois machines : toutes les pièces subissent une opération sur l'une des machines du premier groupe, et ensuite une opération sur une machine du deuxième groupe, éventuellement sélectionnée parmi 2 ou 3 d'entre elles. A cela peuvent s'ajouter des pièces qui ne subissent une opération que sur une machine du deuxième groupe, également avec possibilité de choix. Les transferts des pièces du premier vers le deuxième groupe se font par un chariot automatisé.

2 exemples de configurations :



Les différents types de traits symbolisent différents types de pièces

figure C-1

Une *configuration* détermine les possibilités de chemins que peuvent prendre les différents types de pièces. Selon les besoins, l'agent de production sélectionne l'une des nombreuses configurations répertoriées. Et à cause de l'évolution du marché, on envisage que de nouvelles configurations, encore inconnues, peuvent apparaître à l'avenir.

Le changement de configuration nécessite que la partie commande soit instruite des nouvelles règles de fonctionnement de la cellule d'une manière précise et exhaustive.

a) Les technologies

Une brève comparaison entre machines conventionnelles et machines spéciales montre l'intérêt des systèmes flexibles.

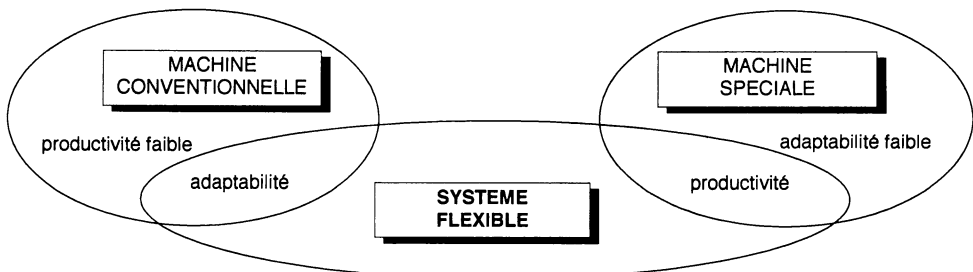


figure C-2

Les machines conventionnelles sont adaptables à une grande variété de travaux mais elles sont peu productives de par leur nature. Les machines spéciales de production sont à l'inverse très productives pour une série donnée mais peu adaptables en raison de la durée excessive du changement de série.

Par exemple : la tête d'une fraiseuse universelle comporte toute une série de roues dentées et de pièces intermédiaires qui permettent de l'orienter à souhait et de l'équiper d'accessoires divers. Cette machine est adaptable mais bien moins robuste que son équivalent à tête horizontale dont la chaîne cinématique est beaucoup plus directe. La puissance transmissible limitera par conséquent les vitesses de coupe et d'avance, les épaisseurs de copeaux ainsi que les géométries et diamètres d'outils.

L'architecture mécanique des machines modernes est basée sur des modèles de robustesse plutôt que de versatilité. Le relatif manque de souplesse qui les caractérise est cependant compensé par l'apport des technologies numériques : elles comportent des calculateurs dédiés à la commande des machines. Ces technologies offrent la possibilité de temps d'adaptation extrêmement brefs, par exemple par téléchargement de programmes très variés.

Les systèmes flexibles bénéficient des avantages des deux solutions traditionnelles : adaptabilité et productivité.

- Adaptabilité

A tout instant, les actions réalisées par le système de production sont adaptées à la tâche à effectuer. Les tâches ne sont pas exécutées de façon répétitive car des pièces de types différents sont susceptibles de se présenter à un poste donné.

- Productivité

La nature des matériels employés (rigidité mécanique, puissance disponible, performance des outils de coupe, etc.) est l'un des atouts qui assure une bonne productivité. Un autre atout essentiel réside dans une bonne structuration de la partie commande. Une grande sûreté de fonctionnement de même que la continuité du flux de production sont ainsi garanties lors des régulières modifications liées à l'évolution des contraintes de fabrication.

- Degré d'Intégration

Les systèmes flexibles sont constitués d'un ensemble de moyens : machines-outils standards, machines spéciales, robots, convoyeurs, etc. On peut les classer selon leur degré d'intégration :

- machine-outil à commande numérique (MOCN) : machine relativement spécialisée de type fraisage, pointage, tournage, poinçonnage, pliage, etc.
- centre d'usinage (CU) : MOCN complétée de périphériques tels que magasin d'outils ou réserve de palettes et aux possibilités étendues grâce à l'asservissement numérique d'un plus grand nombre d'axes,
- cellule flexible : ensemble de plusieurs machines avec circulation automatique des pièces entre elles, autonomie accrue, contrôle intégré de la qualité,
- atelier flexible : composé de plusieurs cellules avec interaction en elles,
- usine intégrée (CIM : Computer Intergrated Manufacturing) : moins une réalité courante qu'un concept phare qui représente un modèle.

1 - 3. Un concept universel

La flexibilité des machines est une préoccupation majeure pour les techniciens : fabrication automatique de costumes sur mesure, fabrication automatique de mobilier sur mesure, etc. Bien des exemples dépassent le domaine industriel : sélection automatique des voies empruntées par les trains en fonction du trafic réel, prélèvement et rangement de

documents écrits ou audio-visuels dans une médiathèque... C'est une des composantes de l'intelligence dont les machines seront de plus en plus dotées.

2 - LA COMMUNICATION INDUSTRIELLE

La notion de système telle qu'elle a été définie dans les deux premières parties de ce livre est maintenant étendue à la réunion de plusieurs machines en une cellule, et à la réunion de plusieurs cellules en un atelier. A un niveau supérieur, on considère l'entreprise dans sa totalité, composée de ses divers ateliers, mais également de services comme la gestion du personnel ou la gestion commerciale.

2 - 1. Les besoins de communication dans une entreprise

Les outils informatiques utilisés dans ces différents milieux ne sont pas issus des mêmes normes ni fournis par les mêmes constructeurs : on dit qu'il sont hétérogènes. Le système complet doit pouvoir gérer cette *hétérogénéité* de même que son *évolutivité*, garante de la pérennité des investissements. Il doit également permettre le *partage des ressources* nécessaire à la cohérence au sein d'une même activité.

Deux types d'acteurs interviennent dans le fonctionnement d'une entreprise : les hommes et les machines. L'interactivité étant la condition première de la vitalité de tout système, l'entreprise est face à trois types de relations :

- la communication entre les hommes,
- celle entre les hommes et les machines,
- et celle entre les machines.

La mise en oeuvre d'un système de production complet implique donc que l'ensemble des éléments qui la composent soit organisé d'une manière interactive et hiérarchisée.

Par ailleurs, les services de communication que doivent fournir les différents équipements nécessitent des *temps de réponses* variés et peuvent accepter des *taux d'erreurs* plus ou moins sévères selon leur fonction. Les *volumes d'informations* peuvent être très variables : un message de synchronisation n'a évidemment pas la même taille qu'un fichier ou un programme. Dans le contexte industriel, l'*environnement électromagnétique* des ateliers peut être parasité par les organes de puissance. Les *distances* à couvrir sont également très variables : quelques mètres entre deux ordinateurs de bureau ou plusieurs centaines de mètres pour relier des contrôleurs d'ateliers d'une usine importante.

Les performances des moyens de communication à mettre en oeuvre dépendent de toutes ces contraintes.

2 - 2. Les modèles de communication

a) La connexion entre deux machines

Le principe du cheminement d'un message d'une machine à l'autre est décalqué sur le comportement humain : d'une façon très schématique, il est souvent comparé à l'échange d'un courrier postal entre deux personnes. L'expéditeur (appelé *émetteur*) prend l'initiative de l'envoi car c'est lui qui en éprouve le besoin : il s'agit de la transmission d'une information ou au contraire de la sollicitation d'une réponse de la part de son correspondant. Le courrier est élaboré suite à une série d'actions bien identifiées. Elles sont décrites figure C-3. Une même série d'actions est nécessaire pour que le destinataire (le *récepteur*) prenne connaissance du message.

Pour les équipements, cette procédure générale de communication a été formalisée et normalisée par l'ISO (International Standard Organization) : c'est le modèle OSI (Open

Systems Interconnection). « Ouvert » signifie que des équipements hétérogènes peuvent communiquer entre-eux.

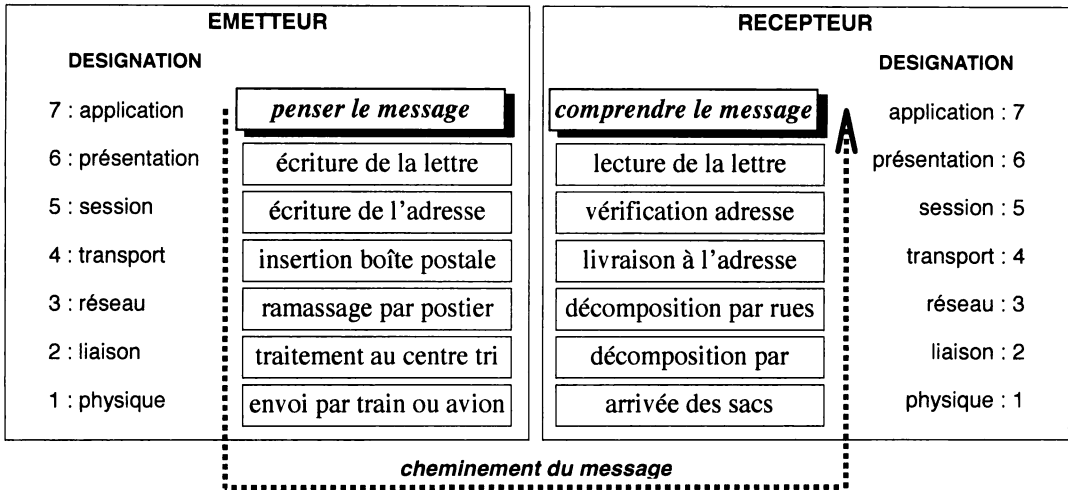


figure C-3

Les couches élevées relèvent exclusivement de logiciels (de la pensée des individus dans la métaphore du courrier) alors que les couches basses relèvent également du matériel (du service postal) :

- les logiciels sont chargés de la gestion interne des données et de la synchronisation des échanges avec le correspondant,
- les couches intermédiaires définissent le mode de codage et la préparation des données,
- les aspects matériels concernent les tensions et intensités des signaux, le brochage des connecteurs et la nature des conducteurs.

Deux points sont à noter :

- la correspondance des fonctions, couche par couche, du côté de l'émetteur et du côté du récepteur : à chaque couche correspond un *protocole* qui définit les règles de dialogue,
- chaque couche rend un *service* à la couche immédiatement supérieure au sein de chaque unité.

Lorsque plus de deux correspondants sont en relation, on parle de *réseau*, et plus particulièrement de *réseau local industriel (RLI)* quand tous les abonnés sont géographiquement regroupés au sein d'une même entreprise et que les contraintes liées à l'industrie, énoncées au § 2 - 1, sont prises en compte. La priorité d'accès au réseau, la taille maximale des messages, la sûreté des transmissions, la complexité des protocoles sont autant de paramètres gérés différemment selon le type de réseau mis en oeuvre. Le modèle OSI sert de base à la normalisation des moyens de communication. Ce domaine est régulièrement en mutation et fait l'objet d'importants enjeux commerciaux.

b) Comparaison entre le modèle OSI et les réseaux locaux industriels

L'informatique de type bureautique ou grand public prend en charge la totalité des couches OSI : l'utilisateur dispose immédiatement d'outils prêts à l'emploi. A cause de la plus grande diversité des services attendus, les fournisseurs de *réseaux locaux industriels* n'implémentent souvent que les couches inférieures : c'est l'utilisateur qui doit alors mettre

en forme les couches élevées, bien que la couche *application* fasse l'objet de nombreux développements.

Illustration dans le cas d'un système bureautique

Soit à tracer un dessin réalisé en DAO sur table traçante. L'utilisateur sélectionne la fonction à réaliser par simple clic de souris sur l'icône appropriée : « tracé du fichier courant ». C'est le système informatique qui réalise le passage des informations, couche par couche :

Présentation : codage des données de couleurs, d'épaisseurs de traits, d'échelle...

Session : contrôle de la synchronisation des échanges avec la table traçante...

Transport : vérification du temps d'établissement de la liaison et des erreurs...

Réseau : décomposition du fichier en paquets et vérification de leur acheminement...

Liaison : constitution des trames au format requis (adresse du destinataire + portion de message + contrôle des trames)

Physique : transmission des bits (respect des connecteurs, des tensions, du codage...)

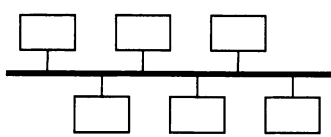
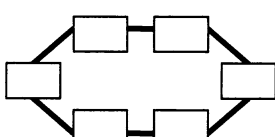
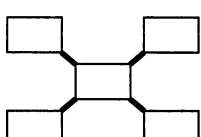
Illustration dans le cas d'un système industriel

Soit à transférer un signal de « départ cycle » d'un automate vers un autre automate. La quasi totalité de l'application est cette fois écrite par l'utilisateur, en particulier sous forme de grafcet pour la partie séquentielle et également sous forme d'instructions logiques pour stocker le message à transmettre dans un tableau de données. La mise en forme du message est ainsi laissée à l'initiative de l'utilisateur, de sorte que la programmation des couches 7 à 3 lui incombe. La taille maximale des messages est souvent limitée, ce qui nécessite quelquefois l'écriture de procédures de dialogues supplémentaires. Seul le transfert des données à destination du récepteur est pris en charge par le réseau : ce sont les couches 2 et 1.

La couche 7 peut de plus en plus souvent être intégrée sous forme d'une interface conviviale : une fenêtre de configuration prête à l'emploi permet alors de définir les variables à échanger.

2 - 3. Topologies des réseaux locaux industriels

Chaque équipement, quelle que soit sa nature, est appelé *noeud* du réseau, ou *station*, ou encore *abonné*. L'organisation et la répartition physique des noeuds peuvent se faire de diverses façons appelées *topologies* dont les principales sont les suivantes :

BUS	ANNEAU	ETOILE
		
Les stations sont raccordées à un médium unique.	Les stations sont reliées deux à deux afin de former un anneau	Chaque station périphérique est reliée à la station centrale par un médium spécifique.
Des procédures plus ou moins complexes doivent gérer les conflits d'accès au bus.	Un jeton informatique circule continuellement sur l'anneau : il recueille et distribue les informations de station en station.	La station centrale concentre les échanges d'informations.
L'ajout, le retrait ou l'indisponibilité d'une station se fait sans remise en cause du restant de l'application.	La longueur de l'anneau peut être très importante du fait de la régénération des signaux à chaque station.	La modification du réseau est aisée.

2 - 4. Architecture d'un système de production

La diversité des besoins des entreprises et de l'offre en matière de composants d'automatismes est telle qu'il est difficile de tracer une architecture type. La figure C-4 propose néanmoins une solution à titre indicatif.

a) Les équipements

Par ordre hiérarchique ascendant, les équipements sont les suivants :

- A : capteurs et actionneurs (également désignés par Entrées-Sorties)
- B : interfaces intelligentes : variateurs de vitesse, appareils de mesure, etc.
- C : nano-automates : assurent les fonctions d'automatismes de base
- D : micro-automates : assurent toutes les fonctions pour la commande de machines spéciales, des convoyeurs, fours, cuves, magasins d'outils, appareils de contrôle, etc.

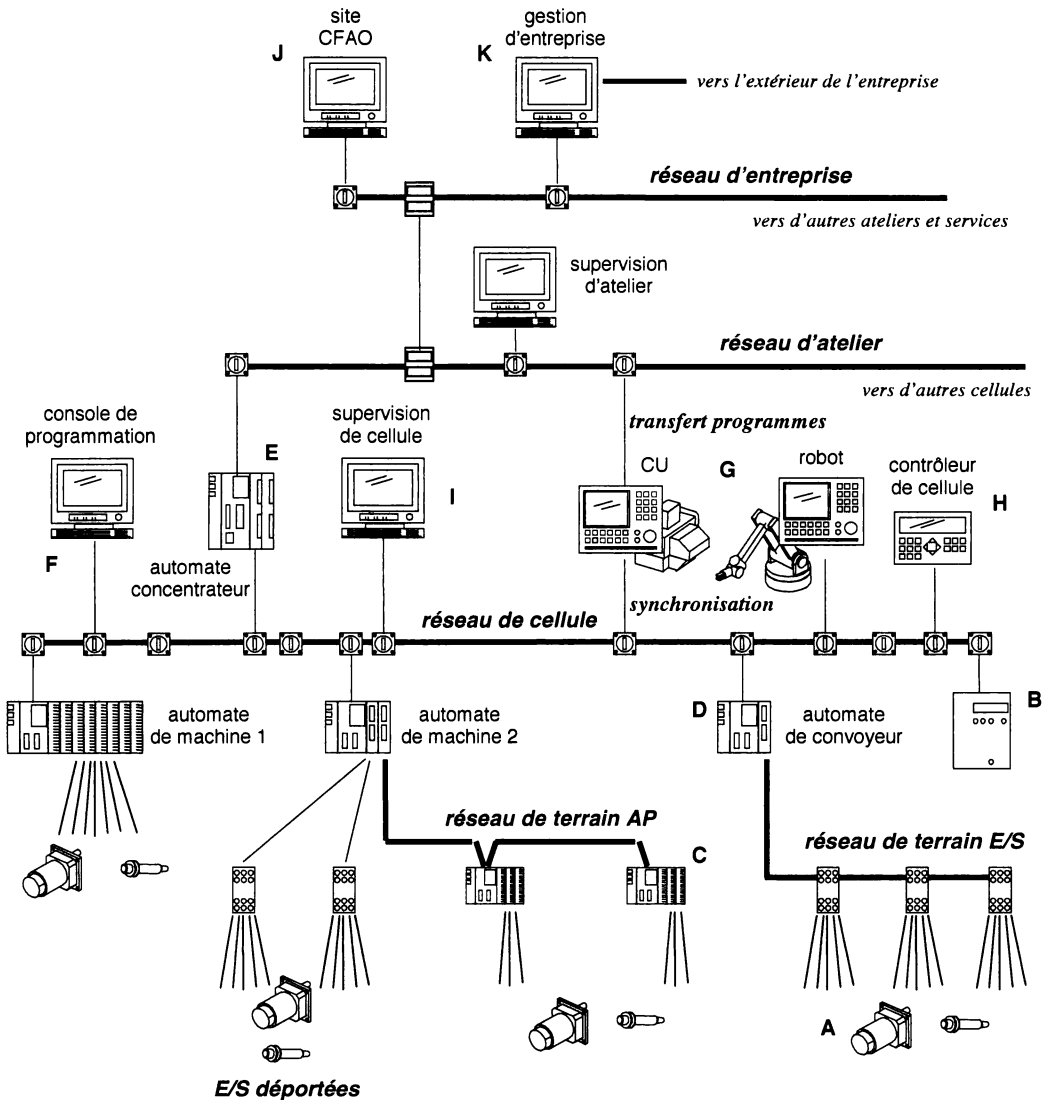


figure C-4

- E : automates : peuvent être équipés de cartes spécialisées très variées de type unité centrale au format PC, disque dur... peuvent recueillir les données process afin de les mettre en forme, les stocker, les transmettre à un superviseur, éditer un rapport d'activité... des capteurs et actionneurs peuvent évidemment leur être raccordés
- F : consoles de programmation : création et archivage des programmes d'automates, modifications, mise au point...
- G : divers équipements de production : CU, robot...
- H : contrôleurs de cellule : du plus simple terminal programmable au contrôleur avec affichage graphique
- I : superviseurs : pilotage central de la cellule et/ou de l'atelier
- J : site CFAO : plusieurs ordinateurs et périphériques reliés en réseau pour la conception des produits, les calculs mécaniques, les contextes de fabrication, la génération et l'archivage des programmes d'usinage
- K : gestion d'entreprise : concerne la production (GPAO), la trésorerie, le personnel, les clients et les commandes...

b) Les composants de liaison

Dans une solution *homogène*, le réseau et le matériel connecté sont fournis par un même constructeur. Les liaisons entre des réseaux de natures différentes peuvent cependant être envisagées grâce à l'emploi de passerelles, appareils capables de gérer simultanément plusieurs protocoles (par exemple un superviseur doté de cartes de communication à protocoles multiples).

Le modèle OSI a généré des normes relatives aux RLI pour garantir l'interconnexion de matériels hétérogènes. Celles-ci restent dans la pratique fédérées par les grands constructeurs. Les produits proposés par les fournisseurs de matériels périphériques et de logiciels sont désormais conformes à l'une ou à plusieurs de ces normes, ce qui permet tout de même une grande souplesse dans le choix des équipements.

Parmi ces *normes*, citons WORLDIFIP (extension de FIP, initié par le Ministère de l'Industrie français : Schneider-Electric en est le principal fournisseur), PROFIBUS (autour de Siemens), INTERBUS-S, DEVICE-NET etc. Le développement du réseau AS-I (pour Actuator Sensor Interface) permet le raccordement de capteurs et d'actionneurs fournis par un grand nombre de constructeurs.

Pour les réseaux d'entreprise, ETHERNET est un *standard* au niveau *physique* qui, issu de l'informatique bureautique, a largement été adopté par l'industrie en association avec les *protocoles* CSMA/CD et TCP/IP (ce dernier étant le couple de protocoles aux niveaux transport et réseau d'Internet).

- Au niveau du terrain

La connexion des capteurs et actionneurs aux automates peut se faire de diverses manières:

- l'exemple de la machine 1 (figure C-4) représente la solution traditionnelle où chaque constituant est relié à une entrée ou à une sortie de l'automate,
- pour la machine 2, les interfaces d'E/S sont d'une part déportées au plus près des capteurs et des actionneurs, et d'autre part un *réseau de terrain d'automates* distribue une certaine intelligence vers un niveau hiérarchique inférieur : le câblage est simplifié et les fonctions sont réparties,
- l'automate du convoyeur est équipé d'un *réseau de terrain d'entrées-sorties* : le câblage est encore simplifié et le coût final du point de connexion est minimum.

- Au niveau de la cellule

Les réseaux de cellule permettent l'échange d'informations entre automates, terminaux, ordinateurs de pilotage, systèmes d'aide à la maintenance, commandes de robots, directeurs de machines-outils à commande numérique, etc.

Des prises supplémentaires peuvent être mises à disposition des agents de maintenance : depuis n'importe lequel de ces endroits, répartis judicieusement, des interventions au pied de la machine sont possibles grâce au raccordement temporaire d'une console de poche (réglages, visualisation, modification de programme, etc.)

- Au niveau de l'atelier

Les réseaux d'atelier assurent la liaison entre les réseaux de cellules et les ordinateurs de gestion : GPAO, téléchargement de programmes entre machines-outils et logiciels de CFAO, suivi de la qualité, etc.

Remarque : la connexion du CU sur le réseau de cellule permet d'échanger des signaux de synchronisation avec les autres machines de la cellule (quelques bits ou octets) alors que sa connexion sur le réseau d'atelier est destinée au téléchargement de programmes complets. Ces réseaux doivent donc répondre à des besoins différents.

- Au niveau de l'entreprise

Ce sont des réseaux de type bureautique qui relient essentiellement des ordinateurs et des périphériques standards : imprimantes, disques durs, tables traçantes, etc.

- L'offre des constructeurs

Les produits normalisés font référence aux niveaux de l'entreprise et sont disponibles pour certains sous des dénominations commerciales propres au fournisseur :

- au niveau du terrain, Schneider-Electric propose FIPIO (pour In-Out), Siemens propose PROFIBUS-DP (et PROFIBUS-PA pour les applications en milieu explosible), la plupart des marques proposant également AS-I,
- au niveau cellule, on trouve FIPWAY et toujours PROFIBUS-DP,
- au niveau atelier : MAPWAY ou ETHWAY et PROFIBUS-FMS,
- au niveau entreprise, le réseau IBM est de type Token-Ring, Digital Equipment propose DECNET, Siemens propose INDUSTRIAL ETHERNET, le réseau de Schneider est ETHWAY etc.

Notons quelques solutions propriétaires :

Sinec-L1 ou MPI pour Siemens, Uni-Telway pour Schneider-Electric, Jbus pour April.

Remarque :

Ce chapitre a traité deux concepts totalement différents : les niveaux hiérarchiques (du terrain à l'entreprise) et les couches du modèle OSI (de la couche physique à la couche application). A chaque niveau de communication, toutes les couches du modèle OSI sont traversées par les informations lors d'une transmission.

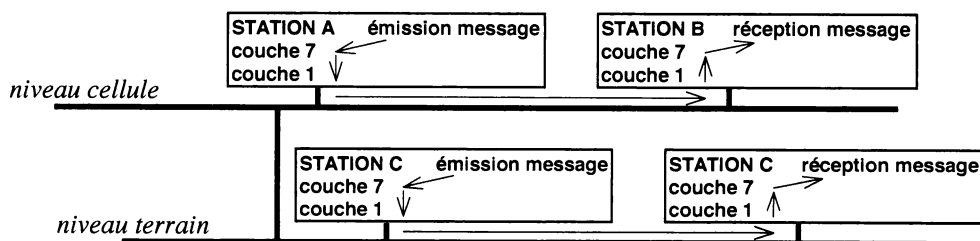


figure C-5

c) Exemples de production intégrée

- Production de voitures

Compte tenu de l'ensemble des options, des couleurs ou encore des conditions spéciales pour l'export... le nombre de variantes d'un même modèle de voiture automobile est impressionnant. La majorité des véhicules sont commandés à l'usine par les concessionnaires en fonction de leur connaissance du marché. Certains clients ont des souhaits particuliers qui font alors l'objet de commandes supplémentaires. En tout cas, les commandes sont personnalisées et transmises à l'usine par voie informatique.

Les pièces standards sont bien entendu produites en grande série : la tôlerie, les moteurs et les transmissions etc. Mais la peinture et le montage final des voitures se font au fur et à mesure de la demande. Bien qu'une chaîne de montage soit dédiée à un seul modèle, deux voitures qui s'y suivent ne se ressemblent jamais. La production est ordonnancée de manière à combiner tous les paramètres, véhicule après véhicule. Un tel résultat est la conséquence de l'évolutivité du marché et du besoin de personnalisation des consommateurs. Notons à ce sujet, à cause de la diversité des variantes, que les séries dites spéciales sont justement les seules qui sont produites dans l'esprit de la grande série. On est loin de l'époque où les constructeurs engageaient les clients à choisir la couleur de leur voiture, pourvu qu'elle soit noire...

- Production de mobilier de cuisine

Là encore, l'intégration peut être totale. Le vendeur prend les mesures de la cuisine vide et saisit toutes les caractéristiques du mobilier commandé : toutes les dimensions, y compris les emplacements des étagères, la qualité des finitions, la couleur, etc. Les données sont transmises à l'usine par le réseau téléphonique depuis l'ordinateur du vendeur. Elles sont directement exploitées par les machines lorsque sont découpés les panneaux de bois : chaque planchette porte dès cette première étape le numéro de la commande sous forme de code barre. De même, le directeur de commande numérique de la perceuse est informé des coordonnées et diamètres des trous individuellement pour chaque planchette. Les pièces détachées d'une même cuisine sont regroupées en fin de fabrication et l'assemblage est automatiquement géré grâce aux données fournies par le vendeur.

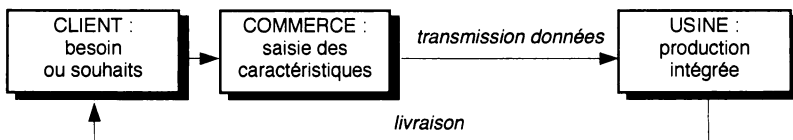


figure C-6

2 - 5. Interfaces opérateurs

a) Les boîtes à boutons et voyants

Au même titre que les capteurs et les préactionneurs, les boutons et les voyants sont reliés soit directement aux coupleurs d'E/S des automates, soit par l'intermédiaire d'interfaces déportées. Des dispositifs complémentaires simples comme les roues codeuses, les potentiomètres ou les afficheurs à segments et les galvanomètres sont souvent requis. On rappelle que les boutons et voyants dédiés à la sécurité sont câblés dans les circuits de sécurité.

b) Le terminal (ou pupitre) opérateur programmable

Les boutons et voyants sont souvent remplacés fort économiquement par un terminal opérateur qui offre de vastes possibilités supplémentaires : visualisation des messages sous forme de texte, navigation dans des menus, accès réservé par mot de passe, etc.

- la liaison avec l'automate se fait la plupart du temps par réseau local et permet donc l'économie d'un grand nombre d'E/S,

- la programmation se fait par configuration de tables de données échangées en permanence avec l'automate.

c) La supervision

La fonction *supervision* consiste à contrôler le processus (visualiser les états des différents équipements) et à le piloter (agir en forçant certains paramètres) à partir d'un poste central. Une vaste gamme de services peuvent en réalité être attendus de cet outil. La supervision existe sous deux formes :

- des contrôleurs commercialisés par les fournisseurs de matériels d'automatismes (d'aspect similaire aux terminaux opérateurs, ils offrent un large écran et un clavier étendu),
- des progiciels fournis par des développeurs informatiques.

Les fonctionnalités sont tout à fait comparables. Quel qu'en soit le type, la supervision offre la souplesse habituelle des outils informatiques : écran graphique, éventuellement tactile, clavier standard, programmation de l'application par configuration, langages standards (C, Visual) ou encore liens avec d'autres logiciels (import-export de données). Le superviseur est généralement connecté à un ou plusieurs réseaux pour échanger des informations avec toute une série d'automates et d'équipements divers. Tous les domaines de l'automatisme sont concernés : les industries de procédés discontinus et continus, la gestion technique des bâtiments (immotique et domotique), immense potentiel d'avenir, ainsi que la gestion des voies de communication (autoroutes, rail, canaux navigables...) ou encore les stations de pompage et d'irrigation.

d) Méthode de développement d'une application de supervision

Deux étapes bien distinctes sont nécessaires :

- la configuration des données
- et ensuite l'édition des écrans et l'écriture de programmes complémentaires.

- Configuration des données

Les tables. Grâce aux outils fournis par le progiciel de supervision, le développeur de l'application définit un certain nombre de tables de données en réception et en émission. Il les déclare aux formats de sa convenance (bits, mots, texte...) et aux tailles et adresses de sa convenance. Chaque table est affectée à l'un des réseaux (dans le cas où plusieurs réseaux sont connectés au même superviseur) et ensuite à l'une des stations ou globalement à toutes les stations. En mode d'exploitation, elles seront échangées avec la ou les stations correspondantes d'une manière cyclique ou sur demande. Elles sont mémorisées ou non pendant l'arrêt du superviseur. Des protocoles différents, simultanément activés, autorisent les échanges de données entre matériels hétérogènes.

Les variables externes. Dans les tables, il faut ensuite définir les variables, en général sous forme symbolique : moteur_broche, vanne_3, pompe_acide, etc. Les adresses de ces variables doivent être cohérentes avec celles déclarées dans les équipements connectés.

Les variables internes. Il est souvent nécessaire de déclarer des variables pour le seul usage du superviseur : registres intermédiaires pour calculs, mémoires de stockage, etc.

- Edition des écrans

Une fois les données configurées, on structure l'application en définissant les fonctions générales de chaque écran et leur arborescence. On détermine ensuite le fond de chaque écran : gris uni, synoptique schématisé ou photographie numérisée de l'installation... Des objets de natures diverses sont ensuite ajoutés : du texte, fixe ou variable, des champs de saisie et/ou d'affichage numérique et alphanumérique, des symboles standards (véris, vannes, etc.) ou personnalisés (fichiers créés par l'utilisateur), des indicateurs de type bar-

graphe ou à aiguille, des curseurs, des voyants, des touches, des courbes, des scénari vidéo. On associe à chaque objet une ou plusieurs variables. Lors du fonctionnement en mode exploitation, l'évolution des variables peut générer l'animation des objets en temps réel. A l'inverse, l'action sur certains objets peut provoquer l'évolution des variables.

- Programmes complémentaires

La simple configuration des objets ne permet cependant pas de réaliser toutes les fonctions habituellement attendues d'un superviseur : calculs logiques (combinaison de conditions pour déclencher un événement par exemple) et calculs numériques (gestion des temps et des coûts de fabrication, boucles de régulation ou algorithmes d'asservissement par exemple). Des outils spécifiques ou des langages de programmation standards sont mis à disposition de l'utilisateur pour réaliser ces fonctions.

- Fonctions intégrées

Le développement rapide des applications est possible grâce à une panoplie de fonctions usuelles directement disponibles :

- gestion des alarmes : cette fonction surveille automatiquement les changements d'états des variables et génère des messages de défaut dont les dates d'apparition, d'acquiescement et de disparition sont consignées. Les défauts sont en outre affectés à des groupes et définis en niveaux de criticité, ce qui permet ensuite de filtrer leur affichage ou leur impression. Pour les variables non binaires (valeurs analogiques), les défauts peuvent être générés sur franchissement de seuils préréglés,
- statistiques : des registres de comptage spécifiques surveillent les temps de fonctionnement des équipements ainsi que le nombre de manoeuvres effectuées. Ils peuvent déclencher des tâches (mémorisation, calcul, affichage de messages, impression...) lorsque des seuils sont atteints,
- tenue du journal : tous les événements souhaités peuvent être inscrits dans le journal de l'application. Il est ainsi possible d'imprimer et de stocker sur disque le rapport d'activité d'une équipe de travail et de garantir la traçabilité de la production sous une forme compréhensible par tous,
- recette : une recette est un tableau de données dont les valeurs attribuées aux cellules sont interchangeable par simple appel d'un numéro de fiche différent. Chaque recette contient autant de fiches que souhaité, toutes les fiches de la même recette étant bien sûr structurées de manière parfaitement identique.

Exemple de recette (figure C-7)

Une brasserie industrielle fabrique une célèbre boisson en quatre phases principales :

1. obtention de la maische par brassage de farine de malt dans de l'eau tiède
2. transformation en moût par filtration, trempé et ajout de houblon
3. refroidissement, fermentation et mûrissement
4. conditionnement

Le type de produit est essentiellement déterminé pendant la deuxième phase. Il dépend :

- du nombre de trempes (1, 2 ou 3),
- des températures successives de la chaudière à trempé,
- des temps d'ébullition,
- de la quantité de houblon.

En réglant ces paramètres, le brasseur peut créer des produits différents (la Kronheibrau, la 1227, la Spéciale, la Tradition, etc.) Les données de fabrication (autrement dit les *recettes* de fabrication) sont mémorisées dans autant de fiches qu'il existe de produits. Pour changer de production, il suffit d'appeler une nouvelle fiche. Les variables sont alors mises à jour immédiatement.

etc ...		température	temps
		42	84

1227		température	temps
		50	0

Kronhelbrau		température	temps
filtration	73	45	0
trempe 1	92	52	18
trempe 2	96	37	
trempe 3	0	0	
houblon	260		

figure C-7

- Fonctionnement en mode exploitation

Chacune des fonctions lit ou écrit des informations dans *une base de données temps réel* qui est commune à toute l'application : la mise à jour de tous les objets du superviseur se fait donc simultanément. Les variables qui sont liées au processus évoluent en même temps que les variables associées des constituants de commande (automate, variateur...) Un léger délai de mise à jour, de l'ordre de plusieurs dixièmes de secondes, est cependant incontournable du fait des différents temps de scrutation qui s'additionnent (superviseur + réseau + automate par exemple). Ces différents interlocuteurs n'étant pas synchronisés, le délai varie continuellement.

e) Exemple simple d'application de supervision

Un chariot se déplace suivant un axe linéaire. Sa position est contrôlée par un codeur incrémental et une carte de comptage rapide. La valeur reçue de l'automate est celle du compteur et correspond donc à un nombre de points du codeur.

On souhaite :

- visualiser les mouvements du chariot sur un écran graphique,
- afficher sa position exacte en nombre de points du codeur et aussi en mètres,
- provoquer le « départ cycle » du chariot à partir du superviseur.

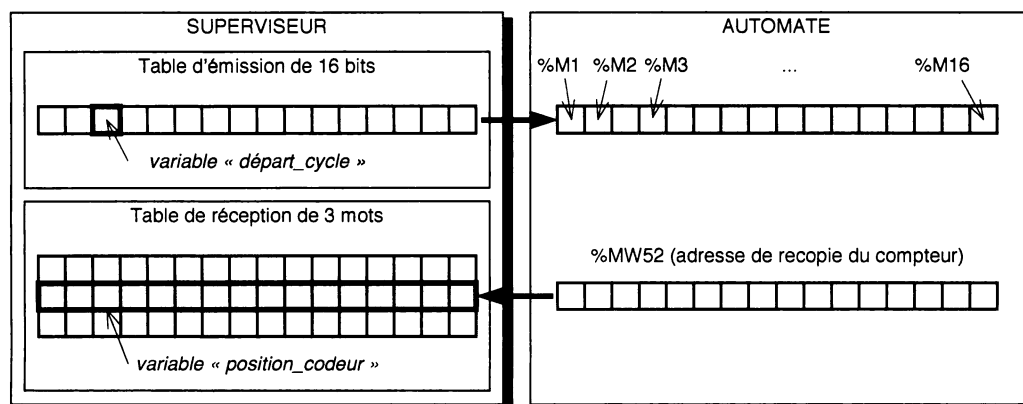


figure C-8

On configure une table en émission de 16 bits à partir de l'adresse 1 (figure C-8) : elle sera transmise vers les 16 bits consécutifs à %M1 de l'automate.

On configure également une table en réception de 3 mots à partir de l'adresse 51 : celle-ci correspondra à %MW51, %MW52 et %MW53 de l'automate.

La variable « départ_cycle », déclarée en indice 3 sur le superviseur, correspond ainsi à %M3. La variable « position_codeur », déclarée en indice 2, correspond à %MW52. De plus, on crée une table interne sur le superviseur qui comporte la variable interne « position_m ». Après conversion, cette variable contiendra la valeur en mètres de la position réelle du chariot.

L'écran de la figure C-9 propose les objets suivants :

- A : les rails appartiennent au fond d'écran, aucune animation n'étant requise,
- B : les touches Marche et Arrêt positionnent la variable « départ_cycle » respectivement à un et à zéro,

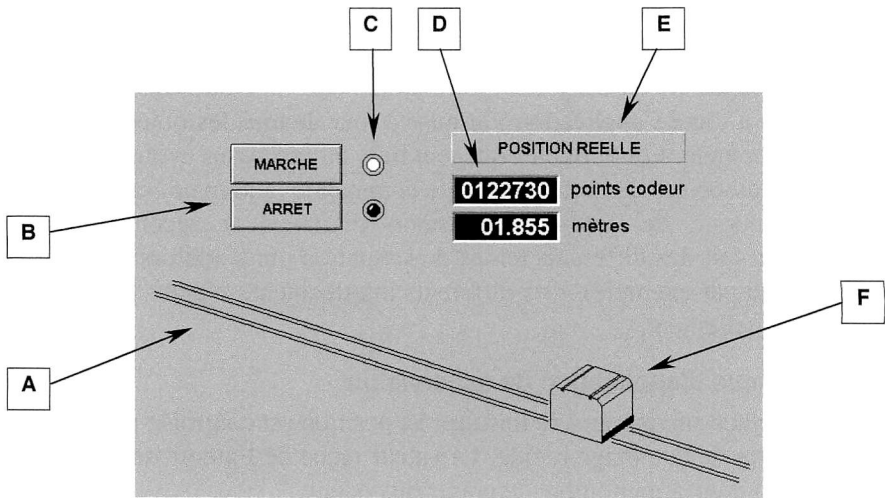


figure C-9

- C : si les objets voyants changent de couleur avec l'état de la variable « départ_cycle », ils visualisent l'état interne du superviseur : il serait préférable de leur associer une variable en lecture issue du processus pour visualiser l'état réel du chariot,
- D : les fenêtres d'affichage visualisent les valeurs des variables « position_codeur » et « position_m » en continu, au temps de rafraîchissement près : dans le cas général, des formats différents peuvent leur être attribués (binaire, entier, flottant, alphanumérique...),
- E : les textes fixes sont des objets paramétrables du superviseur ou peuvent faire partie du fond d'écran,
- F : le dessin du chariot appartient à un fichier graphique spécifique :
 - cet objet se déplace sur l'écran en même temps qu'évolue la variable « position_codeur » : on paramètre cette animation en indiquant les coordonnées écran en X et Y pour les 2 valeurs limites de la variable, la mise à l'échelle est calculée automatiquement par le superviseur,
 - la couleur peut changer selon l'état de la variable « départ_cycle »,
 - un fichier différent comme celui de la figure C-10 peut être appelé momentanément pour montrer par exemple que le chariot transporte une palette.

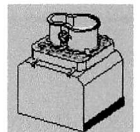


figure C-10

3 - GRAFCET ET FLEXIBILITE

3 - 1. Programmes spécifiques

Une fonction supplémentaire que peuvent offrir les superviseurs et contrôleurs industriels est le téléchargement automatique de programmes vers les automates. Ainsi, il est possible d'organiser un atelier flexible pour lequel chaque type de production nécessite des programmes spécifiques. Un changement de programme nécessite cependant un démarrage à froid de l'automate, ce qui implique une interruption momentanée de la production : arrêt de l'automate, transfert du nouveau programme, redémarrage et initialisation de la partie opérative avant remise en production.

Cette solution est donc parfaitement adaptée à tout système où le temps de changement n'est pas critique : c'est le cas par exemple des machines d'essais qui doivent constamment s'adapter à de nouvelles variantes d'un produit (moteurs, missiles, etc.)

Elle est par ailleurs couramment employée au niveau des machines-outils à commande numérique dont les directeurs de commande sont reliés à un serveur de programmes d'usinage, le système de Fabrication Assistée par Ordinateur (FAO).

3 - 2. Téléchargement de sous-programmes

L'interruption de l'automate peut être évitée en téléchargeant seulement une petite partie de son programme. Ainsi le remplacement de l'expression logique relative à une réceptivité permet de modifier immédiatement le comportement du système. Les automates et les réseaux n'offrent cependant pas tous cette possibilité.

3 - 3. Le grafcet en liaison avec les structures de données

Une autre solution consiste à concevoir pour chaque automate un programme universel. Il est bien entendu que cette « universalité » est limitée à un cahier des charges très précis. En effet, la production d'un atelier, même *très* flexible, est toujours ciblée : un fabricant de moteurs ne produira pas de pièces d'horlogerie, de même, les sous-traitants sont toujours relativement spécialisés (décolletage, mécanique de précision, tôlerie...)

La souplesse d'un programme unique, capable de faire face à toute évolution de la production, est obtenue grâce à sa décomposition en une partie « traitement » et une partie « données » :

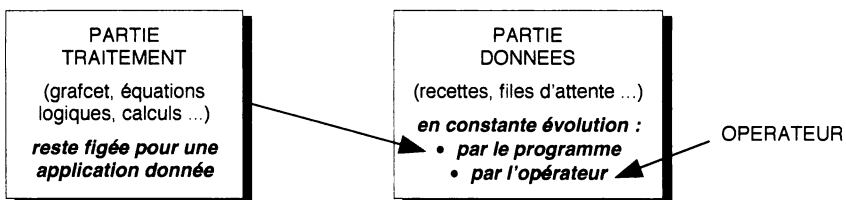


figure C-11

Le traitement doit être le plus court possible pour garantir la rapidité de réaction tout en prenant en compte la lisibilité du programme, en particulier de la partie Grafcet. Pendant l'exploitation, l'adaptation continuelle aux contraintes se fait par manipulation des données à la fois par le programme et par l'opérateur.

- Illustration : solution de l'exemple de la figure C-1 page 114

En premier lieu, l'inventaire des fonctions à réaliser par le chariot conduit à la définition des sous-programmes suivants :

- C/1 pour charger au niveau de la machine 1 du premier groupe, etc...
- C/A pour charger les pièces qui ne passent par aucune machine du premier groupe,
- D/1 pour décharger au niveau de la machine 1 du deuxième groupe, etc...

Ce niveau de précision est suffisant pour l'étude de la gestion du chariot. On recherche ensuite les séquences : une seule pièce est transférée à chaque cycle, ce qui implique que les fonctions globalement notées *charger* et *décharger* (quelle que soit la machine) doivent être exécutées à tour de rôle. Par ailleurs, toutes les trajectoires sont virtuellement possibles : aucune ne doit être écartée. On en déduit la structure générale du grafcet (figure C-12).

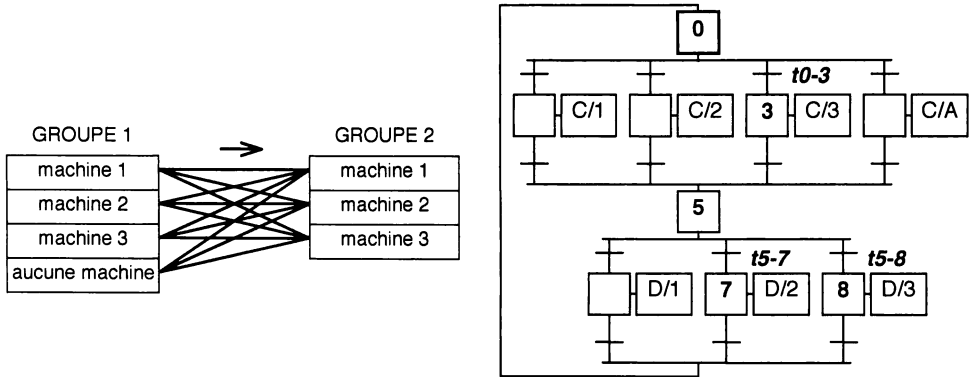


figure C-12

En ce qui concerne la gestion des priorités du côté du chargement, on peut admettre que les niveaux de stocks en aval du groupe 1 déterminent l'ordre de traitement des pièces. Cet exemple est tout à fait comparable à la cellule traitée au chapitre VI (cellule double) : le lecteur se reportera à cette étude pour écrire sans difficulté les réceptivités correspondantes, celles en sortance de l'étape X0.

Il en est de même des réceptivités qui permettent la sélection de la machine de destination (2^o groupe), celles en sortance de l'étape X5. Celles-ci doivent toutefois être complétées par les conditions supplémentaires liées aux différentes configurations des chemins acceptés. (On rappelle qu'une configuration regroupe l'ensemble des possibilités d'évolution de chaque type de pièce). Après reconnaissance du type de pièce, les machines de destination sont déterminées selon la configuration en cours. Ces configurations sont mémorisées dans autant de fiches-recette. Le changement de configuration se fait par appel d'une nouvelle fiche.

		etc ...		groupe 1	groupe 2	
Fiche n°2		groupe 1	groupe 2			3
Fiche n°1		groupe 1	groupe 2			0
	type 1	1	3	0	2	3
	type 2	A	0	0	1	2
	etc ...					

figure C-13

La recette constitue dans le cas présent une *ressource commune* au système d'amenée de pièces en amont du premier groupe ainsi qu'au chariot qui dessert les machines du deuxième groupe.

légende :

1 : machine 1, etc...

A : aucune machine du premier groupe

0 : pas de possibilité

Dans la configuration de la fiche recette n°1 (figure C-13), les pièces de type 1 peuvent passer par les machines 1 ou 3 du premier groupe, puis par les machines 2 ou 3 du deuxième groupe. Les pièces de type 2 passent directement sur l'une des machines du deuxième groupe.

Soit le cas suivant :

La configuration n°1 est en cours d'exploitation. Les conditions liées à la gestion des stocks permettent le franchissement de la transition t0-3. Le chariot charge donc une pièce à la machine 3 du groupe 1 (activation de X3). Il s'agit par exemple d'une pièce de type 1. D'après la recette, cette pièce doit être orientée soit vers la machine 2 soit vers la machine 3 du deuxième groupe. La sélection définitive se fait selon les niveaux de stocks de ces deux machines.

A titre d'exemple, cette condition est exprimée sous forme logique de la manière suivante :

t5-7 = la machine2 est autorisée par la recette ET le stock2 est plus faible que le stock3

Dans la situation décrite, seules les données relatives

- à la configuration n°1
- et au type de pièce n°1

sont significatives à cet instant. Lors de l'évaluation des réceptivités, seules ces informations sont utiles. Afin de minimiser l'expression des réceptivités, il convient donc de restreindre le nombre de paramètres. Les informations adéquates devant être disponibles au moment opportun, le grafcet doit être complété par les fonctions de gestion des données par ajout éventuel d'étapes supplémentaires. Dans cet exemple, la pièce qui vient d'être chargée sur le chariot sera identifiée à l'étape X5 et la ligne de données relative au type de pièce reconnu est dupliquée dans une mémoire temporaire, de taille limitée, seule à être évaluée par les réceptivités.

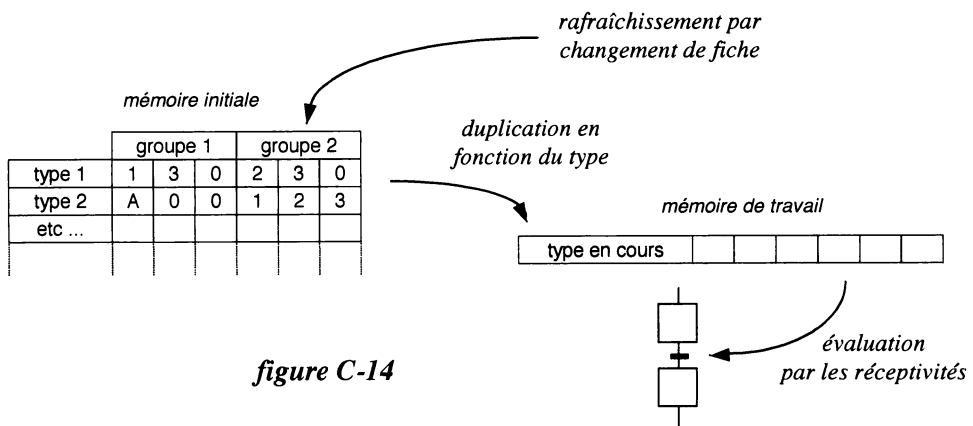


figure C-14

3 - 4. Méthodologie

Les phases successives du développement d'une application sont résumées figure C-15.

La méthode décrite nécessite :

- *un tracé judicieux du grafcet*, obtenu grâce à des analyses séparées des fonctions et des séquences (partie B de ce livre),

- l'établissement de *structures de données* : en cours d'exploitation, elles permettent la manipulation d'un grand nombre de situations dans des temps très brefs,
- la *gestion des données* qui garantit une plus grande simplicité des expressions des conditions d'évolution et par conséquent une excellente sûreté de conception,
- et bien entendu la combinaison du Grafcet et du paramétrage.

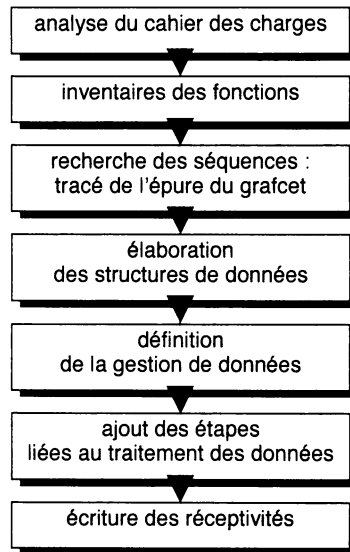


figure C-15

Les avantages sont les suivants :

- l'opération de changement de série est immédiate,
- les changements se font en cours d'exploitation, le flux de production reste fluide,
- les possibilités d'adaptation au marché sont évolutives en ne nécessitant qu'un minimum d'intervention sur le programme d'automate,
- le programme est unique,
- le grafcet comporte très peu d'étapes au vu de la variété des cycles qu'il permet,
- les modifications peuvent s'opérer pour certaines à la demande expresse d'un opérateur et pour d'autres d'une manière automatique en fonction de l'évolution de la production.

VIII . MISE EN OEUVRE

ETUDE DE CAS N° 3 : TRANSFERT LINEAIRE DE CELLULE

1 - MISE EN SITUATION

1 - 1. Contexte

Une usine produit des pompes à la demande des clients : les séries sont faibles voire à l'unité, les morphologies et les tailles des pièces sont très variées. Les opérations d'usinage sont très diverses. Néanmoins, l'entreprise a mis en place *une gamme limitée de pièces standards regroupées en 10 familles* : elles sont personnalisées lors de leur usinage.

1 - 2. Description des palettes

En vue de leur usinage, les pièces (corps, couvercles, etc.) sont maintenues sur des palettes standards adaptées à toutes les machines de la cellule ainsi qu'au système de transitique.

Chaque palette est équipée d'une étiquette magnétique. Des informations peuvent y être inscrites lorsqu'elles passent devant une tête d'enregistrement : numéro de pièce transportée par la palette, horodatage des événements, nom du client et numéro de la commande, etc. Ces informations peuvent être lues lors du passage de l'étiquette devant une tête de lecture, particulièrement pour déterminer le numéro du programme d'usinage à lancer au moment où une palette entre dans une machine, ou bien pour sélectionner la machine de destination lors du transport d'une palette comme on le verra plus loin.

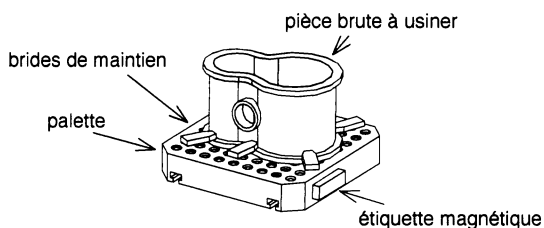


figure C-16 : exemple de palette

1 - 3. Composition de la cellule

La figure C-17 montre schématiquement la composition de la cellule : 6 centres d'usinage autonomes équipés chacun d'un magasin d'outils préréglés et d'un système d'échange de

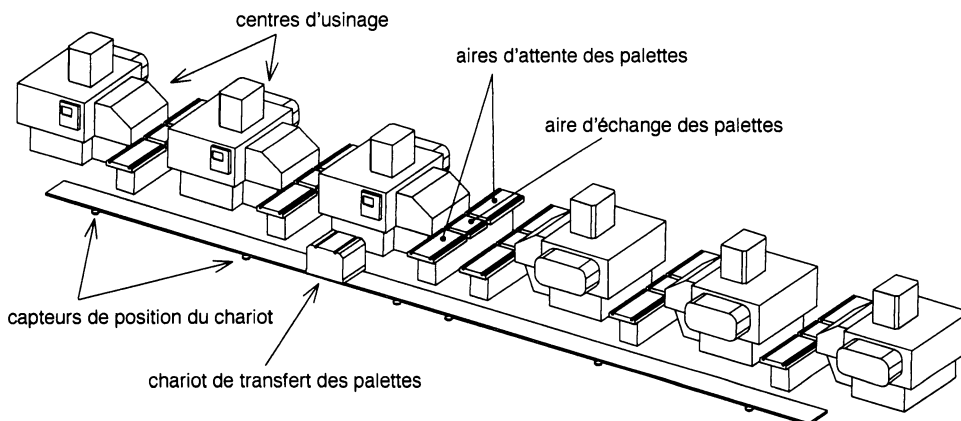


figure C-17

palettes, un chariot linéaire pour le transport des palettes et des aires d'attente pour les palettes en amont et en aval de chaque machine.

2 - FONCTIONNEMENT DE LA CELLULE

2 - 1. Flux physique

La cellule est scindée en deux groupes de 3 machines :

- les palettes sont préparées en amont du premier groupe de machines,
- une première phase d'usinage est réalisée sur l'une des machines du premier groupe (opérations de type fraisage, perçage... ne nécessitant que des outils standards),
- les palettes sont ensuite traitées par l'une des machines du deuxième groupe (opérations spéciales de type fraisage profond, alésage... nécessitant des outils spéciaux),
- elles sont ensuite dirigées vers l'atelier suivant.

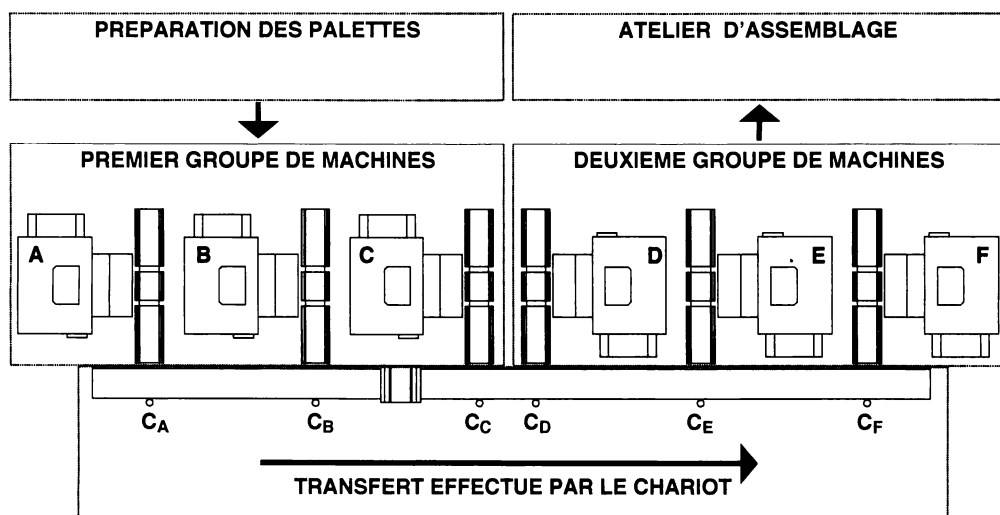


figure C-18 : cellule vue de dessus

2 - 2. Contraintes liées aux méthodes de fabrication

- En amont du premier groupe de machines

L'ordre dans lequel les opérateurs préparent les palettes est déterminé par la gestion de production (commandes des clients, disponibilité des bruts, gestion des stocks, etc.) Les palettes sont orientées indifféremment vers l'une des machines A, B ou C selon leur taux d'occupation. Plusieurs palettes peuvent être en attente devant chaque machine.

- Le premier groupe de machines

Ces machines sont toutes capables de réaliser la première phase de tous les types de pièces. A chaque nouveau cycle d'usinage, le numéro du programme est automatiquement sélectionné par la machine grâce aux informations contenues dans les étiquettes magnétiques.

- En aval du premier groupe de machines

Une palette qui quitte une machine après usinage est temporairement stockée sur l'aire d'attente. Jusqu'à 3 palettes par machine peuvent ainsi être mises en attente avant leur transfert vers le deuxième groupe de machines.

- Le chariot

La fonction du chariot est de transporter les palettes des machines A-B-C vers les machines D-E-F. Il ne peut transporter qu'une seule palette à la fois. Il comporte également un dispositif de lecture-écriture des étiquettes magnétiques.

L'ordre dans lequel les palettes doivent être prises en compte par le chariot doit permettre d'équilibrer le taux de charge des machines A, B et C et éviter en particulier la saturation des stocks en aval de ces machines.

- En amont du deuxième groupe de machines

Un stock maximum de 3 palettes par machine est prévu. Les machines D-E-F sont équipées pour réaliser la deuxième phase d'usinage, plus délicate : elles sont davantage spécialisées et les outils sont spécifiques à des opérations bien déterminées. Ainsi, *la sélection de la machine est faite individuellement pour chaque type de pièce par le Bureau des Méthodes afin de garantir l'adéquation entre les moyens utilisés et les opérations à réaliser.*

- Le deuxième groupe de machines

Comme pour les machines du premier groupe, les étiquettes magnétiques permettent le lancement automatique du programme adéquat à chaque nouveau cycle d'usinage.

- En aval du deuxième groupe de machines

Les palettes sont orientées vers l'atelier suivant dès que l'usinage est terminé.

3 - PRINCIPES TECHNIQUES

3 - 1. Architecture informatique

a) Vue d'ensemble de la cellule

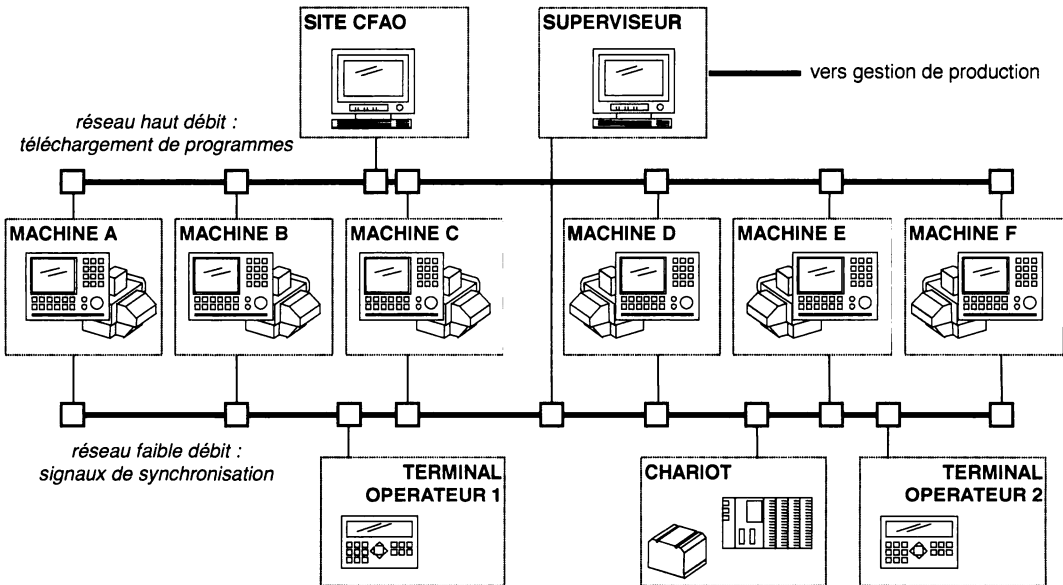


figure C-19

- Les échanges des palettes sont gérés par chaque commande de machine (directeur de commande numérique et automate intégré à la machine),
- les chargements et déchargements des palettes sur le chariot ainsi que les déplacements de celui-ci sont commandés par un automate programmable,

- deux terminaux répartis à des points stratégiques de la cellule permettent certaines fonctions de pilotage,
- un superviseur, également relié à la gestion de production, permet de piloter et de contrôler l'ensemble de la cellule,
- les machines dialoguent en temps réel avec le site de création des programmes d'usinage (FAO) afin de rafraîchir leur mémoire à chaque palette.

b) Interface entre les machines et le chariot

Grâce à l'interconnexion de toutes les commandes par le réseau, les machines A, B et C peuvent émettre chacune un code spécifique dès la fin des cycles. Ces informations permettront au chariot de gérer les stocks en aval de ces machines. La taille de ces informations est limitée du fait qu'il s'agit de signaux de synchronisation.

4 - ETUDE DU FONCTIONNEMENT SEQUENTIEL

Le cas étudié dans ce chapitre a spécialement été sélectionné en raison de la simplicité de l'analyse des fonctions et de la recherche des séquences qui constituent les deux premiers volets de la méthode, étudiés aux chapitres V et VI. Ces points seront donc traités sans difficulté pour aborder rapidement les éléments de base qui permettent à un système d'évoluer d'une manière flexible.

L'objet de ce livre limite l'étude de cette cellule à la seule gestion des déplacements du chariot en vue de la programmation de son automate. Il sera cependant nécessaire de prendre en compte certains éléments de son environnement.

4 - 1. Situation initiale

La fonction principale du chariot étant de faire transiter les palettes du premier vers le deuxième groupe de machines, on peut considérer qu'un cycle est terminé lorsque le transit est effectué. La situation initiale est donc la suivante :

- chariot en position devant l'une des machines D, E ou F,
- chariot vide.

4 - 2. Analyse des fonctions

En référence au chapitre V, l'inventaire des fonctions à réaliser par le chariot est le suivant:

transporter
identifier
charger
décharger
préparer

figure C-20

- **Charger et décharger**

Ces fonctions sont assurées par un mécanisme unique, solidaire du chariot. Le chargement s'effectue à la sortie des machines A-B-C et le déchargement aux stocks d'entrée des machines D-E-F. Ce mécanisme n'étant pas défini dans le cadre de cette étude, les séquences correspondantes seront traitées par des graficets de tâche notés *charger* et *décharger*.

- **Préparer**

Il s'agit du déplacement du chariot à vide vers la gauche en vue de se présenter devant l'une des machines du premier groupe.

Il est important de noter que la destination du chariot (la machine devant laquelle il doit se présenter) ne fait pas partie de la fonction, mais que l'obtention de cette condition constitue l'événement qui produira la fin du déplacement.

Le comportement réel (accélération, décélération, positionnement précis) n'est pas concerné par cette étude, aussi cette action sera-t-elle directement notée *GAUCHE*.

- Transporter

C'est le déplacement vers la droite du chariot : celui-ci transporte une palette. De même que pour *préparer*, l'endroit de destination n'est pas à inclure dans la fonction. Et également par soucis de l'essentiel, cette action sera notée *DROITE*.

- Identifier

Lors du chargement d'une palette sur le chariot, l'étiquette magnétique est lue. Ceci permet en particulier de reconnaître le type de pièce en présence. On rappelle que les pièces ont été définies en un nombre limité de familles et qu'une personnalisation plus poussée est réalisée au moment de l'usinage (§ 1 - 1). En conséquence, *seule l'appartenance à telle ou telle famille est déterminante pour sélectionner la machine de destination du chariot*. Considérons pour l'instant cette fonction comme une tâche subalterne.

4 - 3. Séquence

Le chariot effectue simplement des allers-retours entre les deux groupes de machines. Partant de la situation initiale, l'enchaînement des fonctions est purement linéaire dans ce cas. L'épure du grafcet de commande est immédiate (figure C-21). Les réceptivités t2-3 et t5-0 sont respectivement les comptes-rendus de la fin des tâches *charger* et *décharger*. Il reste à élaborer les réceptivités t0-1, t1-2, t3-4 et t4-5.

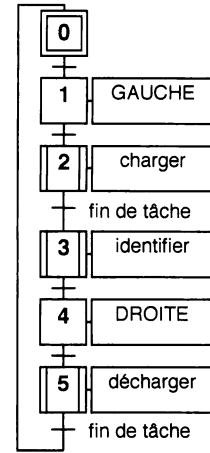


figure C-21

5 - RECEPTIVITES

5 - 1. Chargement au premier groupe de machines

La productivité de la cellule est garantie si les stocks en aval des machines n'arrivent pas à saturation. Un problème similaire a été traité par une gestion de compteurs au chapitre VI (cellule double). On propose ici un cahier des charges plus serré en précisant *que l'ordre dans lequel les palettes sont évacuées des machines soit également celui dans lequel elles sont prises en charge par le chariot*.

a) Structure de données

Une structure de données de type *file d'attente* permet de satisfaire à cette contrainte : il s'agit d'empiler les messages de fin d'usinage émis par les machines au fur et à mesure de leur réception et de les dépiler dans le même ordre à chaque nouveau cycle de transit. La lecture du code courant indique la machine à décharger. Dans l'exemple de la

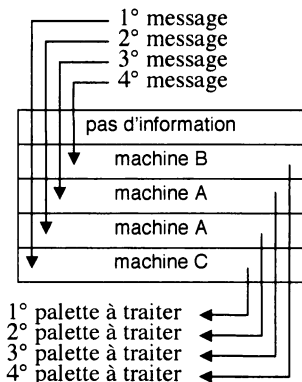


figure C-22

figure C-22, les machines ont évacué des palettes dans l'ordre C → A → A → B.

b) Evolution de la file d'attente

Si la file d'attente est dans l'état de la figure C-22 à la mise en service du chariot, un cycle de transit pour la palette issue de la machine C démarre. Au cycle suivant, la première palette en attente devant la machine A devra être traitée : comme il est inutile de conserver l'information *machine C* et pour éviter la saturation de la file d'attente, celle-ci se décale entièrement d'un pas. Ainsi, à chaque cycle, c'est la même adresse qui détermine l'endroit où charger une palette (le pointeur est fixe). La figure C-23 montre l'état de la file d'attente dans ces 2 situations successives.

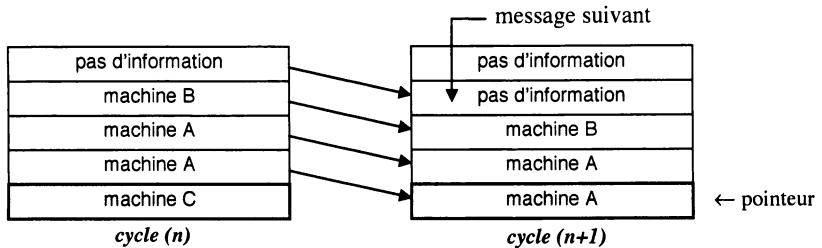


figure C-23

Le décalage de la file d'attente constitue une nouvelle fonction qui peut être réalisée :

- soit en intercalant une étape supplémentaire dans le grafcet,
- soit en la combinant avec une autre action.

Il suffit de veiller à ce que l'information adéquate soit disponible lorsque le chariot exécute la fonction *préparer* et qu'elle soit ensuite régénérée pour le cycle suivant.

Par ailleurs, l'une des machines peut être amenée à évacuer une nouvelle palette à n'importe quel instant : le message qui en résulte est alors empilé.

c) Démarrage et positionnement du chariot

- En début de campagne, la file d'attente ne contient aucune information de sorte que le chariot conserve son état de repos.
- A condition que le chariot soit en service, un quelconque message présent au niveau du pointeur permet d'engager un cycle (évolution $X_0 \rightarrow X_1$) :

la réceptivité t_{0-1} est donc : « pointeur » différent de « pas d'information »

- Pendant le déplacement vers la gauche, le premier capteur rencontré est C_C (voir la figure page 132). Si et seulement si le pointeur indique « machine C », alors le chariot doit s'arrêter suite à l'évolution du grafcet $X_1 \rightarrow X_2$. Sinon le déplacement vers la gauche doit se poursuivre (X_1 reste actif). Une situation similaire se produit en C_B puis en C_A . La condition d'arrêt du chariot (conséquence de l'évolution du grafcet) s'écrit sous forme logique de la manière suivante :

réceptivité t_{1-2} : [(« pointeur » = « machine A ») ET « capteur C_A »]
 OU [(« pointeur » = « machine B ») ET « capteur C_B »]
 OU [(« pointeur » = « machine C ») ET « capteur C_C »]

5 - 2. Déchargement au deuxième groupe de machines

On rappelle que chaque famille de pièces est orientée individuellement vers l'une des machines D, E ou F en fonction des gammes opératoires élaborées par le Bureau des Méthodes. Il est impératif de mettre à disposition des agents de production une interface qui permet la saisie des gammes sans risque d'erreur.

a) Dialogue entre l'opérateur et la cellule

- Terminaux programmables (voir aussi la figure C-19)

Une première façon d'instruire la cellule des gammes opératoires est offerte par les terminaux dont les touches de fonction et la fenêtre de visualisation sont programmables.

D'autres fonctions de pilotage, non abordées dans cette étude, leur sont également assignées : réglage des paramètres d'asservissement du chariot, validation ou inhibition de chaque machine, etc.

Un système de navigation par arborescence permet d'accéder aux paramètres comme illustré dans l'exemple de la figure C-24 : la manipulation décrite aura comme effet d'orienter les pièces de type 03 vers la machine E lorsque celles-ci se présentent.

Par ailleurs, un système de mots de passe à plusieurs niveaux permet l'accès sélectif aux différents menus : un agent de maintenance pourra par exemple régler les paramètres d'asservissement à l'exclusion de toute autre intervention, le paramétrage des gammes étant réservé aux agents de production.

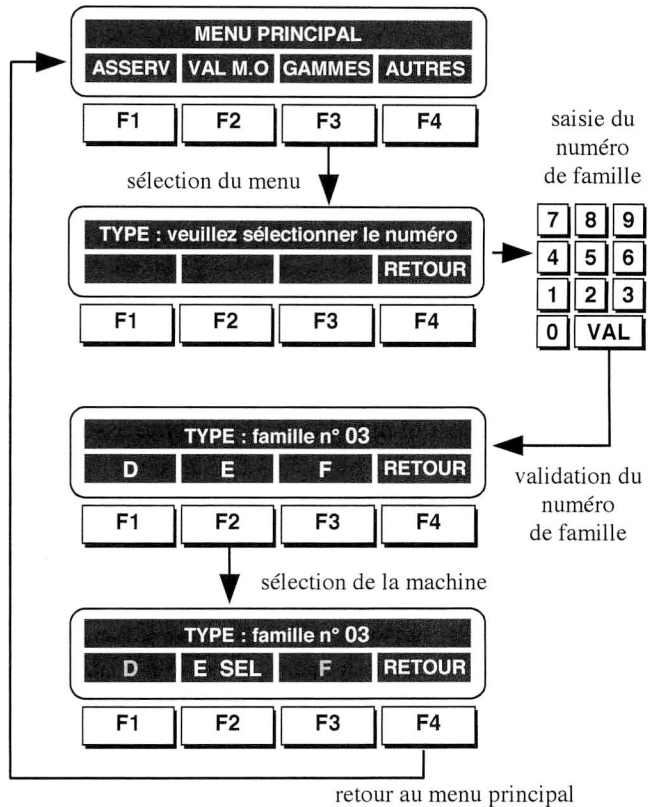
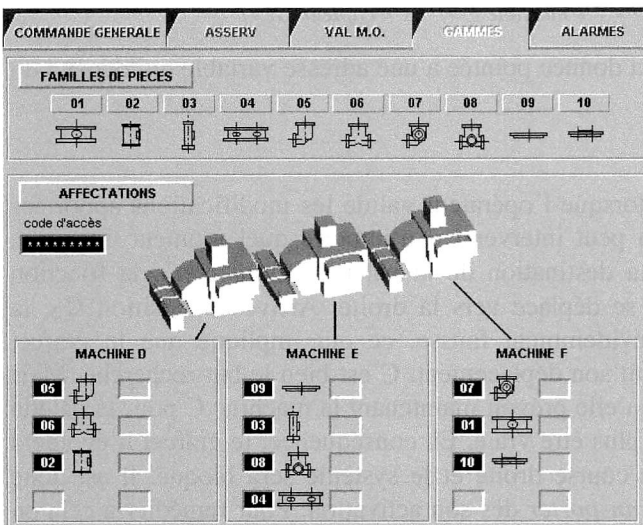


figure C-24

- Supervision



Le réglage des affectations pièces-machines peut également se faire grâce au superviseur. Il offre davantage de convivialité et une meilleure appréciation d'ensemble :

- des objets graphiques rendent la reconnaissance des familles et des machines plus aisée,
- l'affectation se fait simplement en désignant l'une après l'autre : la famille (dans la partie supérieure de l'écran) puis la machine,
- l'écran affiche pour chaque

figure C-25

machine toutes les pièces qui lui sont affectées (partie inférieure de l'écran),

- l'édition se fait entièrement sur une seule page, ce qui évite la recherche dans des menus.

b) Structure de données

Les réglages décrits ci-dessus doivent évidemment être transmis à l'automate de commande du chariot par l'intermédiaire du réseau. Il est nécessaire de structurer les données correspondantes aussi bien du côté des terminaux opérateurs et du superviseur que du côté de l'automate. La nécessité de cohérence entre ces différents partenaires a été démontrée au chapitre VII : cette étude se limite dorénavant au seul automate.

- Mémoire initiale

Le type de structure de données adapté au cas présent est la *recette*. Il s'agit certes d'une recette limitée à un tableau de 10 x 1 entrées puisque l'*information source* est le numéro de famille délivré par l'étiquette magnétique lors du chargement de la palette et que l'*information calculée* est le numéro de la machine associée.

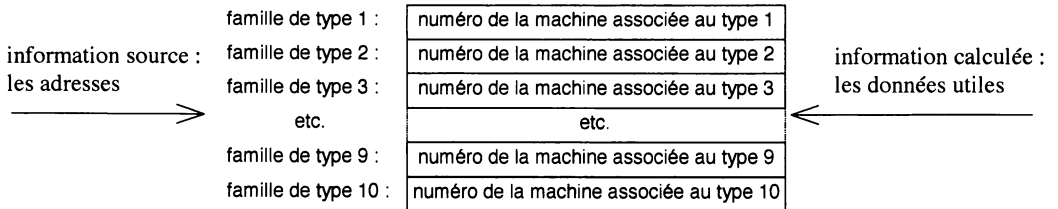


figure C-26

Les numéros de famille constituent les 10 *adresses* de la recette : elles permettent de pointer les *données utiles* à chaque cycle de transit. Les données utiles définissent l'endroit d'arrêt du chariot quand il se déplace vers la droite (fonction *transporter*) : lorsqu'il y a correspondance entre le numéro de machine et le numéro du capteur où se trouve le chariot, celui-ci doit s'arrêter (le grafct devant donc évoluer X4 → X5). Il en résulte que la réceptivité t4-5 s'écrit sous forme logique de la manière suivante :

$$\begin{aligned}
 \text{réceptivité t4-5 :} & \quad [(\text{« donnée pointée »} = \text{« machine D »}) \text{ ET } \text{« capteur } C_D \text{ »}] \\
 & \quad \text{OU} \quad [(\text{« donnée pointée »} = \text{« machine E »}) \text{ ET } \text{« capteur } C_E \text{ »}] \\
 & \quad \text{OU} \quad [(\text{« donnée pointée »} = \text{« machine F »}) \text{ ET } \text{« capteur } C_F \text{ »}]
 \end{aligned}$$

Cette expression est incomplète car la donnée pointée a une adresse variable qui dépend du type de pièce.

- Mémoire de travail

La mise à jour de la recette se fait lorsque l'opérateur valide les modifications apportées aux affectations. Or cette validation peut intervenir à n'importe quel moment du cycle. Supposons que la machine E soit la destination de la palette à transférer : la fonction *transporter* est lancée et le chariot se déplace vers la droite. Arrivé en position C_D , la réceptivité proposée ci-dessus est évidemment fautive, ce qui implique que le grafct n'évolue pas et que le chariot poursuit son déplacement. C'est bien le but recherché. Mais si la recette est alors mise à jour et qu'elle prévoit maintenant la machine C pour la palette en cours, cette réceptivité ne pourra plus être vraie. En conséquence, le grafct n'évoluera plus. Le chariot se perdra en fin de course droite et le système sera bloqué. Il est donc nécessaire de stabiliser la fonction *transporter* dès son activation. Pour remédier à cela on

prévoit une mémoire supplémentaire, appelée *mémoire de travail*. Le rafraîchissement de ce registre constitue une fonction supplémentaire à ajouter sur le grafcet immédiatement à la suite de *identifier*. La stabilité et la cohérence du cycle sont ainsi garanties une fois la palette prise en charge par le chariot : les modifications apportées à la recette seront réellement effectives au cycle de transit suivant.

Par la même occasion, la donnée pointée est maintenant une adresse invariable. La réceptivité t4-5 est modifiée comme suit :

réceptivité t4-5 : [« mém. travail » = « machine D »] ET « capteur C_D »]
 OU [« mém. travail » = « machine E »] ET « capteur C_E »]
 OU [« mém. travail » = « machine F »] ET « capteur C_F »]

6 - ELEMENTS DE PROGRAMMATION

6 - 1. Format des données

Une phase essentielle dans la préparation du grafcet au niveau programmation consiste à procéder au choix des formats des données et de leurs adresses.

- Numéro de famille

Ce numéro variant de 1 à 10, un octet est suffisant. On peut également utiliser un mot de 16 bits selon la préférence de l'utilisateur ou les spécificités de l'automate.

numéro de famille : %MW30

figure C-27

- File d'attente

- Le nombre total de palettes qui peuvent physiquement être stockées en aval des machines A-B-C détermine la taille de la file d'attente : 9 positions sont nécessaires et suffisantes.
- Le nombre de machines et le type de codage retenu déterminent le format minimal des positions. Le choix doit ensuite se porter sur une suite d'octets, de mots ou de double-mots en fonction essentiellement des plages d'utilisation des paramètres (figure C-28-a).

position 9 :	%MW58	<input type="text"/>	famille 1 :	%MW61	<input type="text"/>
position 8 :	%MW57	<input type="text"/>	famille 2 :	%MW62	<input type="text"/>
position 7 :	%MW56	<input type="text"/>	famille 3 :	%MW63	<input type="text"/>
position 6 :	%MW55	<input type="text"/>	famille 4 :	%MW64	<input type="text"/>
position 5 :	%MW54	<input type="text"/>	famille 5 :	%MW65	<input type="text"/>
position 4 :	%MW53	<input type="text"/>	famille 6 :	%MW66	<input type="text"/>
position 3 :	%MW52	<input type="text"/>	famille 7 :	%MW67	<input type="text"/>
position 2 :	%MW51	<input type="text"/>	famille 8 :	%MW68	<input type="text"/>
pointeur :	%MW50	<input type="text" value="machine_appel"/>	famille 9 :	%MW69	<input type="text"/>
			famille 10 :	%MW70	<input type="text"/>

mémoire travail : %MW80

- a - (file d'attente)

- b - (recette)

figure C-28

- Recette

La taille du tableau a déjà été définie. Bien que ce ne soit pas obligatoire, on verra qu'il est intéressant, de même que pour la file d'attente, de choisir des adresses consécutives. La figure C-28-b propose des adresses pour les deux langages.

- Assignation de symboles

Afin de faciliter l'écriture et la maintenance des programmes, la plupart des logiciels de programmation permettent l'utilisation de symboles associés aux adresses : on procède en général par configuration de tables d'assignation. Dans le cas étudié, les symboles qui correspondent aux adresses indicées 30, 50 et 80 en font partie.

- Codification des machines

Les codes associés aux machines sont exprimés selon le tableau ci-contre. Par ailleurs, les logiciels des terminaux opérateurs et du superviseur doivent garantir que toutes les positions de la recette ont bien été saisies avant de permettre le démarrage de la cellule.

machine	code
aucune	0
A	1
B	2
C	3
D	11
E	12
F	13

6 - 2. Grafcet au niveau programmation

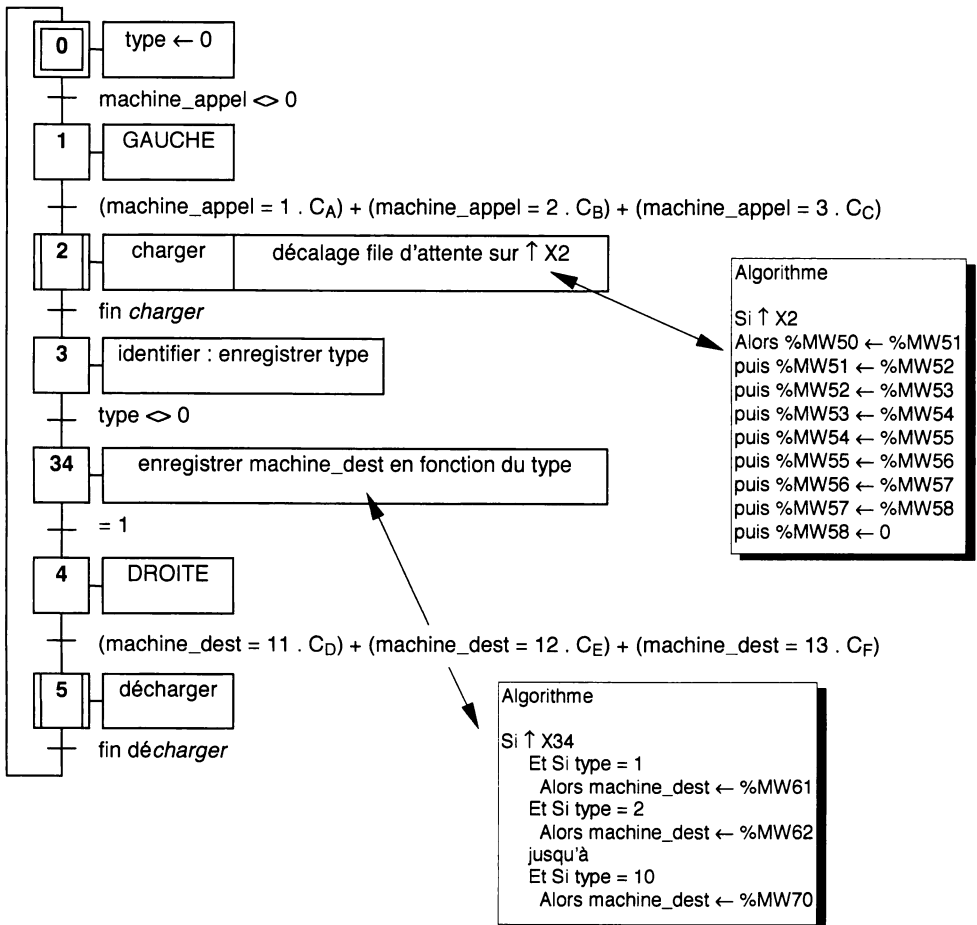


figure C-29

6 - 3. File d'attente

a) Décalage

- Unicité du décalage

Lors de la programmation du décalage, il faut bien veiller à ce qu'il ne soit effectué qu'une seule fois pour que son évolution ne se fasse que d'un seul pas.

Selon l'automate, la technique sera différente :

- En Step5, qui n'intègre pas les fronts en tant que fonctions de base, on peut par exemple activer un bit dès la fin du décalage et n'autoriser celui-ci que si le bit est désactivé. Cette désactivation sera programmée sur X0 par exemple.
- En PI7-Micro, il suffit de programmer le décalage sur front montant de l'étape X2 grâce à la fonction *PI* : *action à l'activation*.

- Remise à zéro de la file d'attente

La mise à zéro systématique de %MW58 permet la purge progressive de la file d'attente lorsque toutes les positions étaient utilisées avant le décalage.

- Sens de programmation

Si par inadvertance l'ordre de transfert des variables était inversé lors de la programmation (%MW58 ← 0, puis %MW57 ← %MW58, etc.) la file d'attente serait immédiatement vidée de tout son contenu par copies successives de la valeur 0 dans toutes les adresses.

- Utilisation d'instructions de base

Les opérations de transfert de variables conduisent à une liste d'instructions de longueur proportionnelle à la taille de la file d'attente. Ci-contre un exemple : duplications successives du contenu de l'adresse *n* à l'adresse *n-1*.

```
%MW50 :=%MW51
%MW51 :=%MW52
etc...
%MW57 :=%MW58
%MW58 :=0
```

- Utilisation d'instructions avancées

Les instructions avancées sont d'utilisation plus aisée.

Par exemple : ROR_ARW est l'instruction de décalage dans le sens souhaité, 1 est le pas du décalage, %MW50 est l'adresse de début (les numéros d'adresses doivent se suivre) et 9 est la taille du registre.

```
ROR_ARW (1,%MW50:9)
%MW58 := 0
```

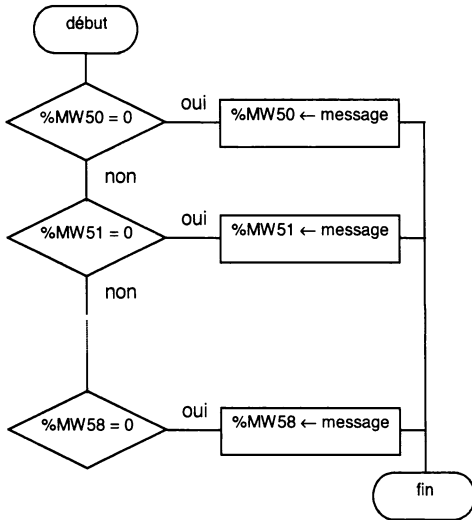
b) Ajout d'une nouvelle valeur

Il s'agit d'enregistrer le message émis par l'une des machines. La position à laquelle doit être inscrite cette information est la première qui est à zéro en partant du pointeur.

- Utilisation d'instructions de base

Attention, bien que cette question puisse être traitée par Grafcet, il est conseillé de tracer un organigramme. En effet, l'opération à réaliser est un calcul et il vaut mieux réserver le Grafcet à sa vocation première : l'exécution du programme sera beaucoup plus rapide.

L'algorithme de recherche est lancé sur réception d'un message émis par une machine : des tests successifs déterminent la première adresse rencontrée dont le contenu est 0 pour y inscrire le numéro de machine. La file d'attente ne peut déjà être saturée puisque dans ce cas les trois stocks seraient également tous saturés, ce qui implique qu'aucune machine ne serait en mesure d'évacuer une palette, et *à fortiori* d'émettre un message.



```

LD [%MW50<0]
JMPC %L10
%MW50 := message
RET
%L10
LD [%MW51<0]
JMPC %L20
%MW51 := message
RET
%L20
etc ...
    
```

Annotations:

- si %MW50 n'est pas = 0
- alors saut conditionnel
- sinon,
- enregistrement
- et retour
- ainsi de suite pour toutes les positions

figure C-30

- Utilisation d'instructions avancées

• Voici à titre d'exemple, un extrait en Step 5 de Siemens, antérieur à la norme sur les langages de programmation. L'instruction de substitution B permet en langage Step5 de réduire considérablement la taille du programme, surtout si la file d'attente comporte un grand nombre de positions. Une boucle itérative scrute les différentes positions les unes après les autres et *s'interrompt naturellement* quand le message a été enregistré.

note 1 : le pointeur MW100 doit être positionné à 50 avant chaque exécution de ce bloc de programme.

note 2 : le bit de contrôle -bit_contr est positionné à un par l'instruction S et est mis à zéro lorsqu'apparaît un nouveau message. Ce bloc n'est appelé que si ce bit = 0 pour éviter l'empilement multiple du message.

```

Exemple en Step5
ETI2 : B MW100
      : L MB0
      : L KF+0
      : ><F
      : SPB= ETI1
      : L -message
      : B MW100
      : T MB0
      : S -bit_contr
ETI1 : BEA
      : L MW100
      : L KF+1
      : +F
      : TMW100
      : SPA= ETI2
      : BE
    
```

Annotations:

- substitution de l'adresse MB0 par le contenu de MW100
- vérification de la position pointée
- enregistrement à la position pointée
- fin de bloc
- incréméntation et mise à jour du pointeur
- itération suivante en cas d'échec

figure C-31

• Les langages normalisés offrent des instructions plus compactes :

```

%MW10 := FIND_EQW(%MW50:9,%KW27)
%MW50[%MW10] := message
    
```

Dans cet exemple, la première ligne opère la recherche et l'adresse calculée est stockée dans le mot %MW10. FIND_EQW réalise la recherche de la première position égale à une valeur donnée, %MW50 est le début du tableau de mots et 9 sa longueur, %KW27 est une constante quelconque qui aura préalablement été configurée à 0, qui est l'objet de la recherche. La deuxième ligne enregistre le message dans le mot d'adresse 50 augmenté du contenu de %MW10.

La deuxième ligne enregistre le message dans le mot d'adresse 50 augmenté du contenu de %MW10.

• Des registres spéciaux peuvent être utilisés, comme par exemple les blocs fonction %R. Les différents paramètres sont réglés par configuration (mode de fonctionnement, variables d'évolution...) la programmation étant alors relativement limitée.

Tous ces programmes traduisent chacun à sa manière l'algorithme de la figure C-31. Ils doivent être appelés à partir d'un programme hiérarchiquement supérieur qui détecte la réception d'un nouveau message et qui doit s'assurer que cet enregistrement ne s'effectue qu'une seule fois.

6 - 4. Recette

a) Mise à jour de la recette

Le rafraîchissement de la recette n'est pas synchronisé avec le cycle du chariot, aussi cette fonction n'apparaît-elle pas sur le grafcet. Un sous-programme exécuté en continu copie dans la recette les informations reçues des terminaux et du superviseur par l'intermédiaire du réseau.

b) Mise à jour de la mémoire de travail

- Utilisation d'instructions de base

« La variable *machine_dest* prend la valeur du mot pointé par la variable *type*, qui dépend du numéro de famille ».

La fonction ainsi définie peut se traduire de différentes manières dont voici un exemple :

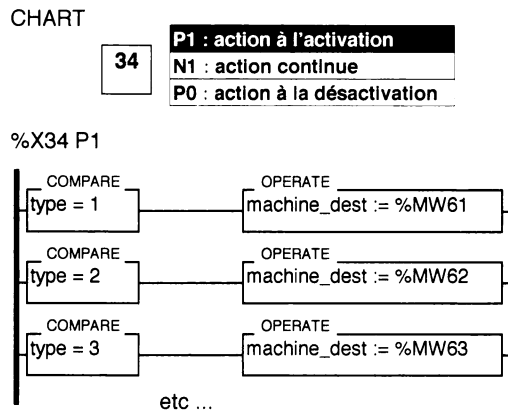


figure C-32

Cette programmation est d'abord très simple et le résultat est très explicite mais s'avère trop lourde lorsque le nombre de données à gérer est important.

- Utilisation d'instructions avancées

La fonction à réaliser peut être exprimée d'une autre façon : « l'opération consiste à affecter à la variable *machine_dest* la valeur du mot dont l'adresse correspond au numéro de famille augmenté de 60 (famille 1 \Leftrightarrow adresse 61) ». Cette manière de définir la fonction se traduit dans les programmes de la figure C-33.

Notons que les adresses de la recette doivent être consécutives. Contrairement à la gestion de la file d'attente, il n'y a aucun inconvénient à ce que la duplication de la donnée utile dans la mémoire de travail se fasse plusieurs fois à chaque appel de cette fonction.

Grâce aux instructions avancées, le nombre d'instructions est limité, et ce quelle que soit la taille du tableau de données : les programmes de la figure C-33 sont directement exploi-

tables si le nombre de familles est augmenté, même de manière importante. Dans d'autres cas, seules des modifications mineures seraient alors à apporter au programme.

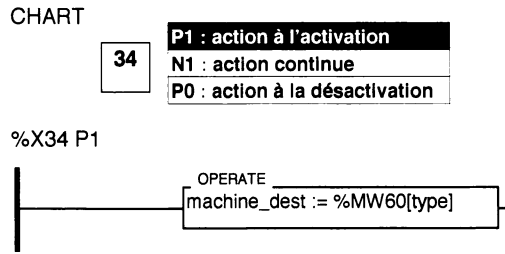


figure C-33

IX . EXERCICES DE SYNTHÈSE

1 - BACS DE TREMPE

1 - 1. Présentation du sujet

Le poste de traitements de surface étudié est composé de 4 bacs de trempe, d'un dispositif automatique d'accrochage munis d'un système de reconnaissance de type de plateau, d'un dispositif automatique de décrochage et d'un portique sur lequel évolue un chariot.

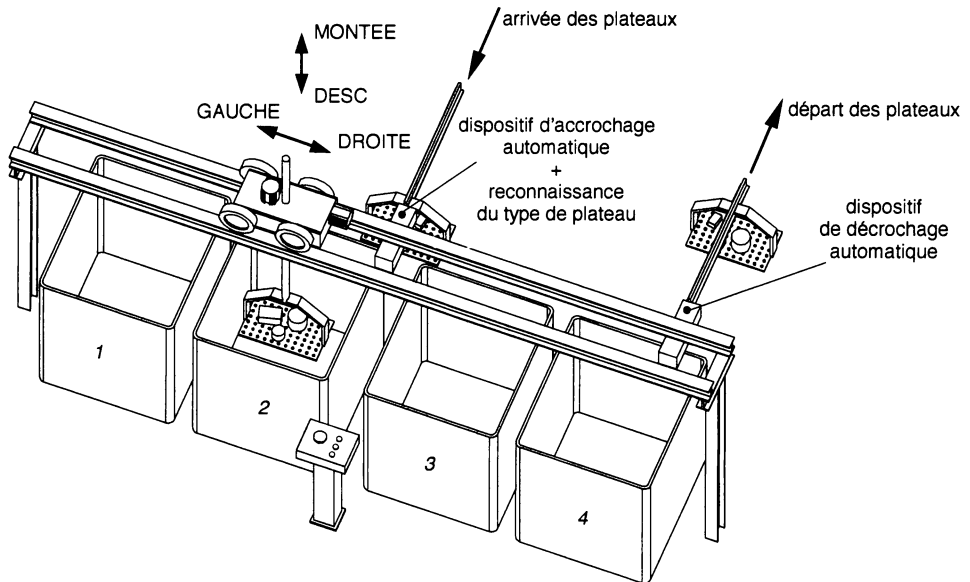


figure D-1

Les plateaux proviennent d'un atelier d'usinage et se présentent dans le désordre à l'entrée de ce poste. Chaque type de plateau nécessite un cycle de trempe spécifique, appelé *gamme*. En voici quelques exemples :

- bac 2 → bac 3 → bac 3
- bac 1
- bac 4 → bac 2 → bac 3 → bac 4 → bac 1

Le nombre total de gammes réalisables dépend du nombre de bacs et du nombre maximum d'opérations. Pour 4 bacs et 8 opérations par exemple, la sélection de la gamme se fait parmi 87380 possibilités ($4^1+4^2+4^3+4^4+4^5+4^6+4^7+4^8$).

On souhaite :

- *pouvoir affecter à chaque type de palette l'une de ces gammes en la décrivant simplement,*
- *modifier la gamme alors que le processus est en mode exploitation,*
- *choisir la durée de trempe individuellement pour chaque opération.*

1 - 2. Fonctionnement du système

A condition qu'un plateau soit présent à l'entrée du poste, un cycle est lancé :

- le plateau est identifié afin de déterminer la gamme associée,
- le plateau est pris en charge par le chariot,
- le chariot se déplace vers le premier bac demandé,
- le plateau plonge dans le bac,
- une temporisation est lancée,
- le délai écoulé, le plateau remonte et le chariot se positionne au-dessus du bac suivant, le cas échéant, pour effectuer une nouvelle trempe,
- lorsque toutes les opérations sont terminées, le plateau est décroché et quitte le poste,
- le chariot se présente immédiatement au dispositif d'accrochage,
- un nouveau cycle démarre si un plateau est présent à l'entrée du poste.

1 - 3. Enoncé du sujet

Les questions ne concernent que le mode de fonctionnement normal : la gestion des modes de marches et d'arrêts ainsi que la surveillance sont gérés à des niveaux hiérarchiques supérieurs.

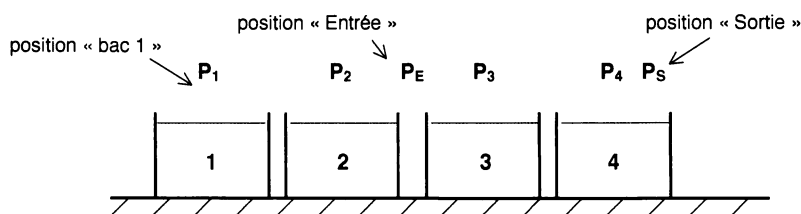


figure D-2

a) Question 1 : analyse des fonctions

Faire l'inventaire des fonctions et les détailler. Une décomposition en plusieurs niveaux est nécessaire : on rappelle que l'analyse est suffisamment fine lorsque les *fonctions* peuvent être transcrites en *actions*.

Note : les dispositifs d'accrochage et de décrochage ne sont pas définis pour l'instant. L'étude détaillée de leur automatisme n'est pas demandée.

b) Question 2 : épure du grafctet

Rechercher les séquences obligées et tracer une épure du grafctet de commande. Ne préciser aucune réceptivité pour l'instant. Note : le trop grand nombre de gammes possibles ne permet évidemment pas de les décrire toutes par autant de séquences de grafctet.

c) Question 3 : structure de données

On souhaite mémoriser les cycles pour 12 types de plateaux. Les bacs sont au nombre de 4. Le nombre maximum d'opérations de trempe est limité à 8. Tracer la structure de données qui permet d'archiver les gammes. Toutes les données sont *résidentes dans l'automate* : celui-ci ne dispose pas de fonction intégrée de type recette, aussi faut-il développer entièrement la structure de données.

d) Question 4 : interface utilisateur

L'accès direct à la structure de données par un personnel non autorisé est une source de danger. En effet, la maîtrise du procédé est indispensable pour paramétrer correctement le système. Proposer une interface utilisateur sûre et facile à mettre en oeuvre sachant qu'aucune connexion informatique ne lie le poste de traitement à une autre machine.

e) Question 5 : réceptivités développées

Ecrire les expressions des réceptivités qui permettent de gérer les déplacements du chariot entre 2 opérations consécutives : utiliser les noms des détecteurs de la figure D-2 et définir des symboles pour les mémoires utiles.

Peut-on simplifier ces expressions de manière algébrique ?

f) Question 6 : gestion des données et équations simplifiées

On constate qu'au lancement d'un cycle de trempe, seules les données relatives au type de plateau traité sont nécessaires. Par ailleurs, seules certaines données sont utiles à chaque opération de trempe. Tracer la *mémoire de travail* qui découle de cette constatation.

Reconsidérer les expressions des réceptivités et compléter le grafcet par des étapes qui commandent les différentes fonctions de gestion des données.

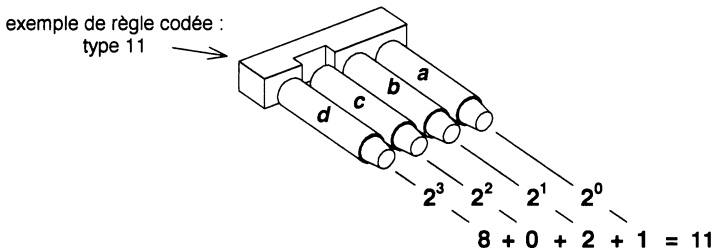
Tracer ensuite le grafcet complet.

g) Programmation

Si le lecteur ne dispose pas du manuel de référence du constructeur, il se reportera directement au corrigé.

- Reconnaissance du type de plateau

Les plateaux sont identifiés grâce à un jeu de 4 détecteurs (solidaires du dispositif d'accrochage) combiné à des règles codées (solidaires des plateaux) :



Les détecteurs sont reliés aux entrées %I2.2, %I2.3, %I2.4 et %I2.5.

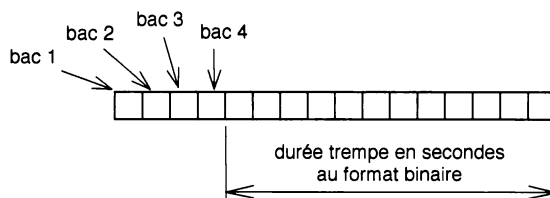
Proposer des solutions pour mémoriser le type de plateau en fonction des états des détecteurs.

- Paramétrage de la durée de trempe

Cette question concerne la contrainte suivante, formulée dans le cahier des charges initial :

La durée de trempe doit pouvoir être spécifiée individuellement pour chaque opération.

On précise que les données relatives à une opération de trempe sont transmises à l'automate sous le format suivant :

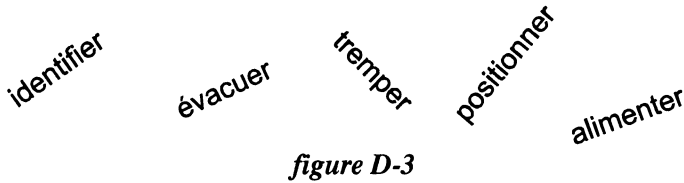


Proposer des solutions pour gérer la durée des trempes.

1 - 4. Corrigé et commentaires

a) Analyse des fonctions (Q.1)

Un premier niveau d'analyse permet de dégager les fonctions inventoriées à la figure D-3 : cette façon de les représenter évite de suggérer un quelconque aspect séquentiel dès ce stade de l'étude.



Fonction : identifier

Il s'agit de déterminer le type de plateau en fonction de la morphologie de la règle codée.

Fonction : évacuer

Quel que soit l'endroit où se trouve le chariot, il se positionne devant le dispositif de décrochage. Celui-ci a lieu. Le chariot se positionne ensuite devant le dispositif d'accrochage.

Fonction : tremper

Quel que soit le bac où se trouve le chariot, le plateau descend. Dès qu'il est arrivé en position basse, une temporisation est lancée : sa durée dépend de la valeur prévue pour la trempe en cours. Une fois le délai écoulé, le plateau remonte. Arrivé en position haute, la fonction de trempe est terminée.

Fonction : positionner

Quel que soit le bac où se trouve le chariot, il se positionne au-dessus du bac requis pour la trempe de l'opération suivante. Si le prochain bac est le même que celui où une trempe vient d'avoir lieu, aucun déplacement n'est associé à cette fonction.

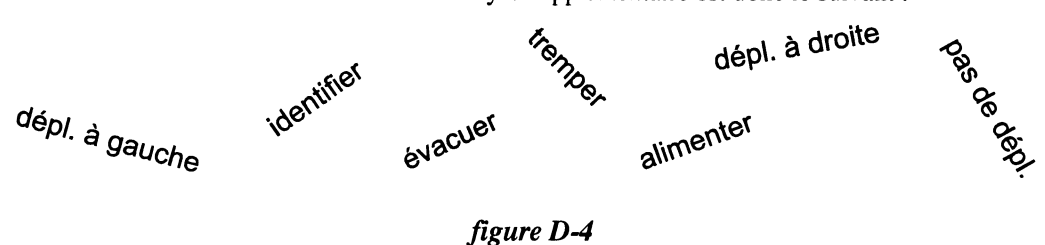
Fonction : alimenter

Cette fonction consiste seulement à accrocher le plateau au bras du chariot.

- Décomposition supplémentaire

- La fonction *identifier* est étudiée en détail à la question 7.
- Les fonctions *évacuer*, *tremper* et *alimenter* feront l'objet d'un grafcet de tâche chacune. Il ne subsiste aucune difficulté pour mettre en place les actions.
- Par contre, aucune action ne peut être associée à la fonction *positionner* au niveau d'analyse actuel. En effet, l'exécution de cette fonction nécessite 3 comportements différents :
 - le chariot se déplace vers la droite,
 - le chariot se déplace vers la gauche,
 - le chariot ne se déplace pas.

L'inventaire des fonctions à un niveau d'analyse supplémentaire est donc le suivant :



b) Epure du grafcet (Q.2)

L'organisation chronologique des fonctions peut être étudiée très librement dans un premier temps en s'aidant d'un diagramme :

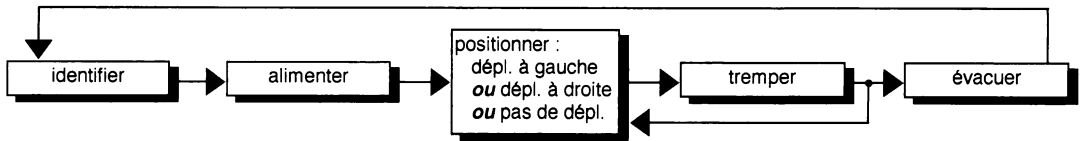


figure D-5

Un grand nombre d'opérations peut être géré grâce à cette structure générale par itérations. L'épure du graficet en est déduite aisément (figure D-6).

c) Structure de données (Q.3)

La structure de données initiales doit être conçue afin qu'elle puisse mémoriser toutes les informations nécessaires à la gestion des plateaux. On recense l'ensemble des paramètres :

- on souhaite traiter 12 types de plateaux,
- un traitement nécessite un maximum de 8 opérations, chacune pouvant s'effectuer dans l'un des 4 bacs.

L'affectation d'un bac à chaque opération peut être réalisée de deux manières, soit en utilisant un quartet dont chaque bit est l'image d'un bac, soit un octet (ou un mot) qui contient le numéro de bac. La figure D-7 propose la structure sous les deux représentations.

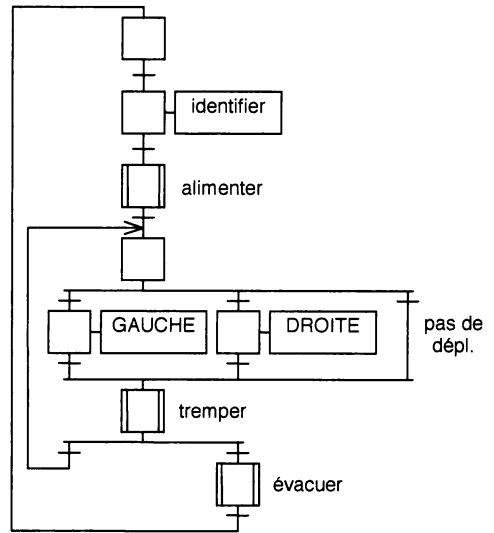


figure D-6

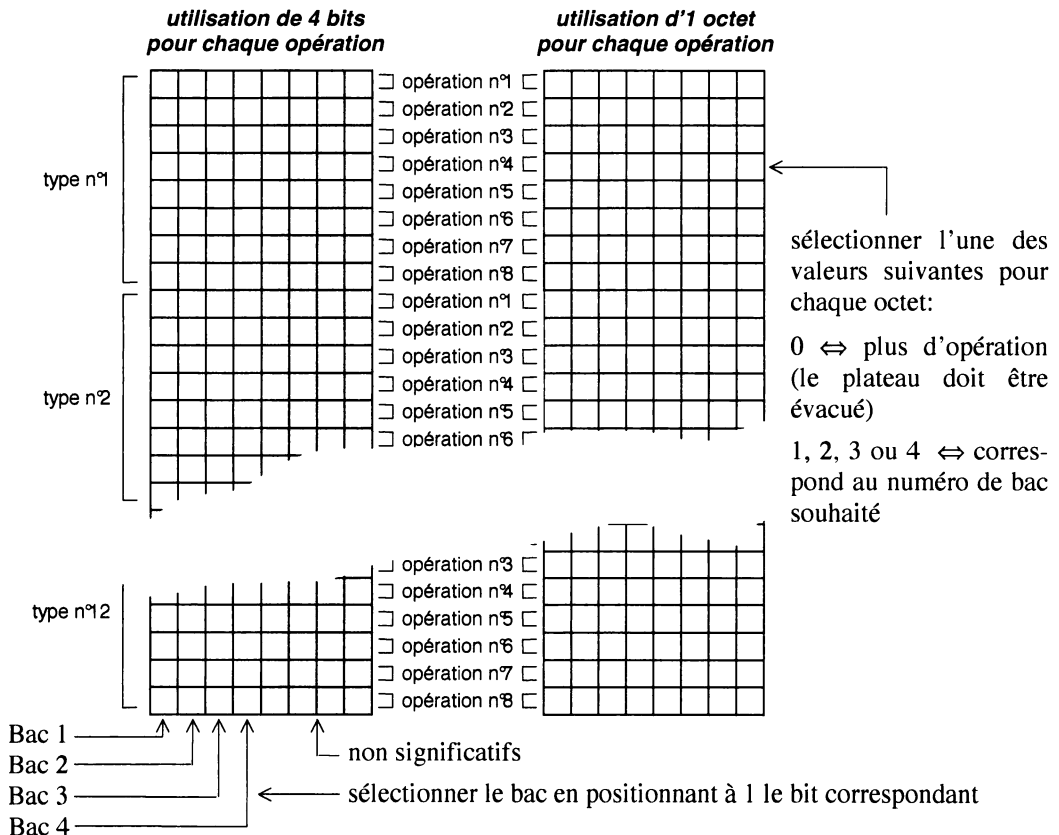


figure D-7

La figure D-8 montre l'exemple de la gamme : bac 3 → bac 4 → bac 4 → bac 1 → bac 2

d) Interface utilisateur (Q.4)

Le personnel de conduite de ce poste n'est pas forcément qualifié pour configurer les données sous la forme présentée : une interface plus conviviale est nécessaire. Il est judicieux de prévoir un mot de passe pour réserver l'accès de cette configuration au seul personnel compétent. Le poste de traitement n'étant pas connecté à une plus vaste structure, un terminal opérateur est indiqué pour la saisie des gammes. Il est également chargé de la mise en forme automatique des données et de leur mise à jour dans la mémoire de l'automate. L'illustration ci-contre reprend le principe du terminal déjà décrit au chapitre VIII (page 137, figure C-24).

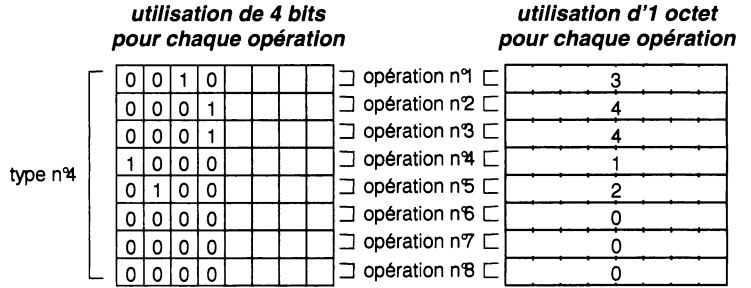


figure D-8

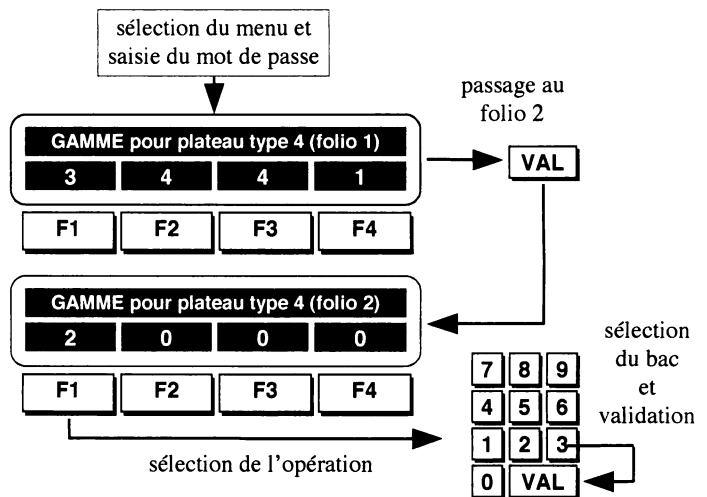


figure D-9

Soit à nouveau la gamme de la

figure D-8 : on souhaite modifier la cinquième opération en remplaçant le bac 2 par le bac 3. La figure D-9 indique la manipulation à effectuer pour répondre à ce besoin. Si c'est l'une des quatre premières opérations qui doit être modifiée, l'appui sur la touche de fonction correspondante permet la sélection du bac lorsque le folio 1 est affiché.

e) Réceptivités développées (Q.5)

Le grafcet de la figure D-6 ne décrit que les possibilités d'évolution du système mais ne peut pas, dans son état actuel, gérer de cycle complet relativement à une gamme. Les réceptivités jouent le rôle de règles qui permettront la réalisation effective d'un tel cycle.

On renvoie le lecteur à la page 80. Les réceptivités sont déterminées en réponse aux interrogations suivantes :

- ⇒ quelles informations permettent le départ des tâches,
- ⇒ quelles informations permettent leur arrêt.

En premier lieu, un certain nombre de réceptivités sont d'écriture immédiate :

- la fonction *identifier* démarre lorsqu'un plateau est présent et s'arrête dès que le type de plateau est enregistré, ce qui provoque la tâche d'alimentation,
- en aval des étapes qui font appel à des tâches, les réceptivités comportent l'étape de comptendu de fin de tâche.

Il reste à définir les conditions qui permettent de sélectionner le sens de déplacement (ou l'absence de déplacement) du chariot. La sélection en aval de l'étape qui appelle la tâche de *trempe* se fait selon le numéro de bac de la prochaine opération : si ce numéro est nul, le plateau doit être évacué.

Le sens de déplacement dépend de deux paramètres :

- l'endroit où doit se rendre le chariot pour effectuer l'opération suivante
 ☛ *celui-ci est matérialisé par la donnée correspondante*
- l'endroit où se trouve le chariot à l'instant considéré
 ☛ *celui-ci est matérialisé par le détecteur correspondant*

Le recensement de tous les cas de figure permet l'écriture de l'expression générique des réceptivités.

Plusieurs possibilités sont à l'origine du déplacement vers la gauche :

- si la prochaine opération a lieu dans le bac 1
et que le chariot se trouve au-dessus du bac 2, ou du bac 3, ou du bac 4, ou à l'entrée du poste
- ou si la prochaine opération a lieu dans le bac 2
et que le chariot se trouve au-dessus du bac 3, ou du bac 4, ou à l'entrée du poste
- ou si la prochaine opération a lieu dans le bac 3
et que le chariot se trouve au-dessus du bac 4

Plusieurs possibilités sont à l'origine du déplacement vers la droite :

- si la prochaine opération a lieu dans le bac 2
et que le chariot se trouve au-dessus du bac 1
- ou si la prochaine opération a lieu dans le bac 3
et que le chariot se trouve au-dessus du bac 1, ou du bac 2, ou à l'entrée du poste
- ou si la prochaine opération a lieu dans le bac 4
et que le chariot se trouve au-dessus du bac 1, ou du bac 2, ou du bac 3, ou à l'entrée du poste

Il n'y a pas de déplacement lorsque :

- la prochaine opération a lieu dans le bac (n)
et que le chariot se trouve au-dessus du bac (n)

On choisit la structure de données par bits pour la suite de l'étude. Compte tenu des différents type de plateaux ainsi que des opérations successives, la réceptivité en amont de X_{GAUCHE} s'écrit donc :

$R_{\text{GAUCHE}} =$

$$\begin{aligned} & \{ \text{type}=1 . \text{op}=1 . [M_{111} . (P_2+P_3+P_4+P_E)] + [M_{211} . (P_3+P_4+P_E)] + [M_{311} . P_4] \} \\ & + \{ \text{type}=1 . \text{op}=2 . [M_{112} . (P_2+P_3+P_4+P_E)] + [M_{212} . (P_3+P_4+P_E)] + [M_{313} . P_4] \} \\ & \text{etc...} \\ & + \{ \text{type}=1 . \text{op}=8 . [M_{118} . (P_2+P_3+P_4+P_E)] + [M_{218} . (P_3+P_4+P_E)] + [M_{318} . P_4] \} \\ & \quad + \{ \text{type}=2 . \text{op}=1 . [M_{121} . (P_2+P_3+P_4+P_E)] + [M_{221} . (P_3+P_4+P_E)] + [M_{321} . P_4] \} \\ & \quad + \{ \text{type}=2 . \text{op}=2 . [M_{122} . (P_2+P_3+P_4+P_E)] + [M_{222} . (P_3+P_4+P_E)] + [M_{322} . P_4] \} \\ & \quad \text{etc...} \\ & \quad + \{ \text{type}=2 . \text{op}=8 . [M_{128} . (P_2+P_3+P_4+P_E)] + [M_{228} . (P_3+P_4+P_E)] + [M_{328} . P_4] \} \\ & \quad + \{ \text{type}=3 \dots \} \end{aligned}$$

ainsi de suite pour les 12 types de plateaux

Avec les mémoires M_{ijk} :
 i varie de 1 à 4 : nom du bac
 j varie de 1 à 12 : type de plateau
 k varie de 1 à 8 : numéro d'opération

extrait de la structure de données
pour les deux premières opérations du type 1

M_{111}	M_{211}	M_{311}	M_{411}				
M_{112}	M_{212}	M_{312}	M_{412}				

Le choix de la structure de données par octets conduit à une expression légèrement différente, mais tout aussi dense. Il n'est pas possible de simplifier algébriquement cette expression car toutes les mémoires sont indépendantes.

f) Gestion des données et réceptivités simplifiées (Q.6)

A chaque opération, seuls sont nécessaires les 4 bits qui appartiennent à la fois au plateau en cours de traitement et à l'opération à réaliser :

- les plateaux se présentent dans un ordre aléatoire
 ☞ *la zone mémoire à isoler dépend du type de plateau à chaque cycle complet*
- les opérations sont systématiquement réalisées dans l'ordre 1 à 8
 ☞ *une file d'attente permet de dérouler les opérations au fur et à mesure*

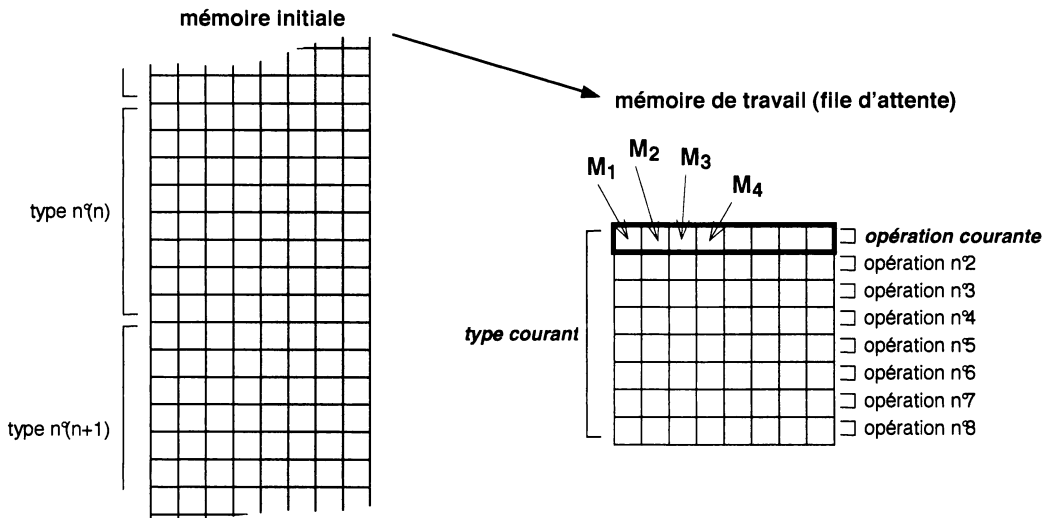


figure D-10

Le grafcet est complété par les fonctions suivantes :

- dupliquer les données relatives au type de plateau immédiatement après sa reconnaissance,
- à chaque nouvelle opération, décaler la file d'attente d'un pas vers le haut et enregistrer la valeur 0 dans l'octet du bas (pour purger les données pour le cas où 8 opérations sont demandées).

La réduction du nombre de mémoires à tester simplifie considérablement les réceptivités. Leur développement complet conduit aux expressions suivantes :

$R_{\text{GAUCHE}} = [M_1 \cdot (P_2+P_3+P_4+P_E)] + [M_2 \cdot (P_3+P_4+P_E)] + [M_3 \cdot P_4]$
$R_{\text{DROITE}} = [M_2 \cdot P_1] + [M_3 \cdot (P_1+P_2+P_E)] + [M_4 \cdot (P_1+P_2+P_3+P_E)]$
$R_{\text{pas de dépl.}} = [M_1 \cdot P_1] + [M_2 \cdot P_2] + [M_3 \cdot P_3] + [M_4 \cdot P_4]$

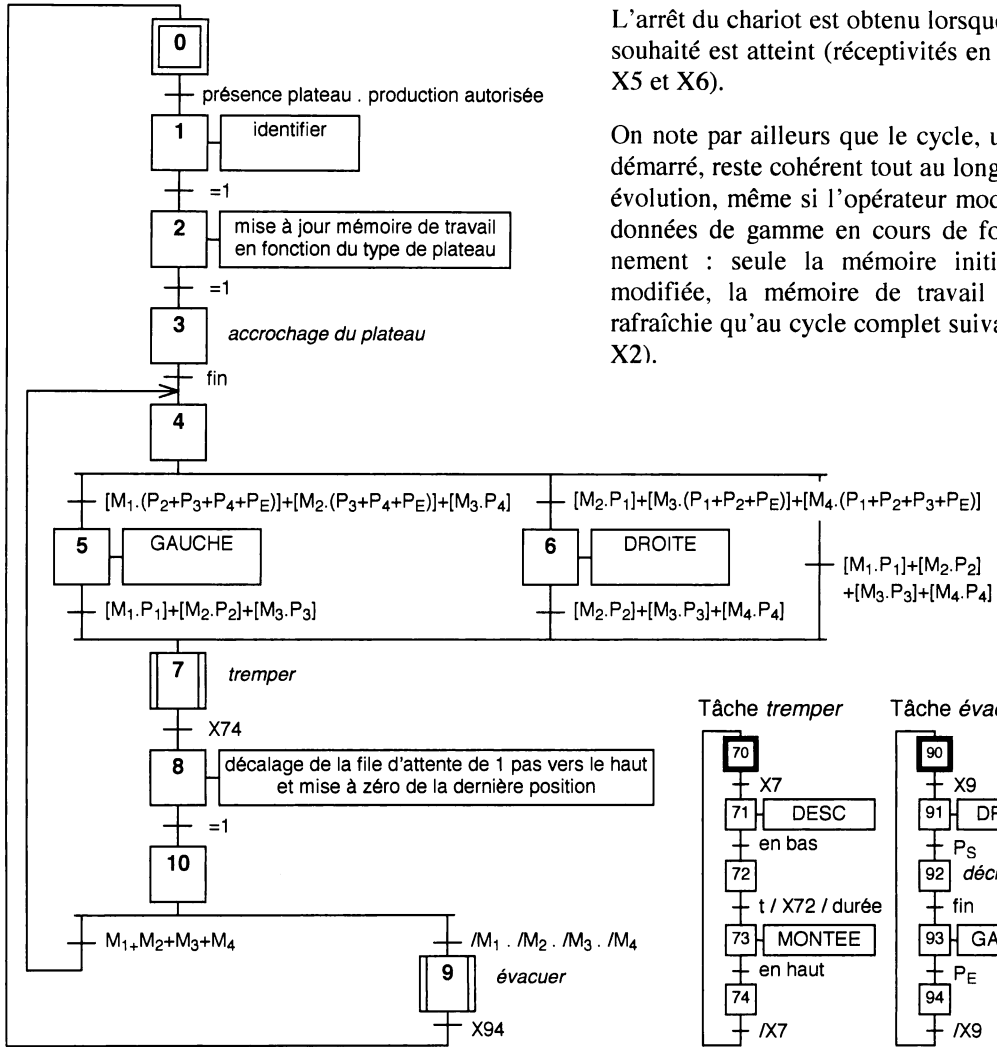


figure D-11

g) Programmation

- Reconnaissance du type de plateau

L'utilisation d'instructions de base n'est pas très élégante : pour les 12 types de plateaux, il faut en effet tester quasiment toutes les combinaisons des 4 entrées qui correspondent aux détecteurs. L'organigramme général est représenté ci-dessous :

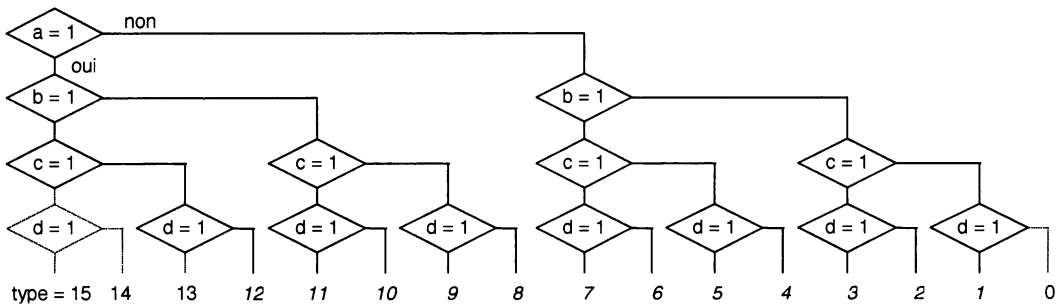


figure D-12

Le programme peut avantageusement être simplifié en utilisant des *instructions avancées*. Le principe consiste à lire le module d'entrées complet sur lequel sont reliés les détecteurs et de déduire directement le type de plateau à partir du profil binaire obtenu.

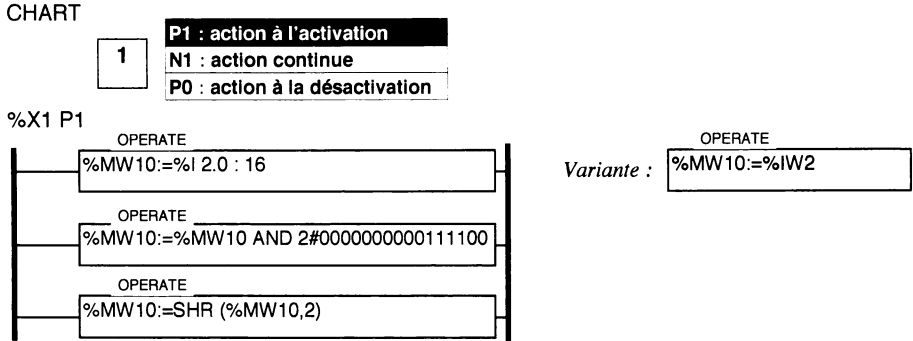


figure D-13

La procédure est la suivante :

- 1) lecture du mot associé au module d'entrées : les 16 bits consécutifs à %I2.0
- 2) masquage des informations indésirables et filtrage des informations significatives (1 pour les données significatives, 0 pour les données non significatives) réalisés par la fonction ET SUR MOT (fonction logique effectuée bit par bit) : AND
- 3) décalage de 2 bits vers la droite pour faire correspondre les poids des bits : SHR
- 4) et mémorisation dans le mot d'indice 10 (%MW10)

En fonction de l'état réel des capteurs (détecteurs de type de plateau et autres capteurs liés au processus) le module d'entrées est par exemple dans l'état de %IW2 ci-dessous au moment de la réalisation de cette opération :

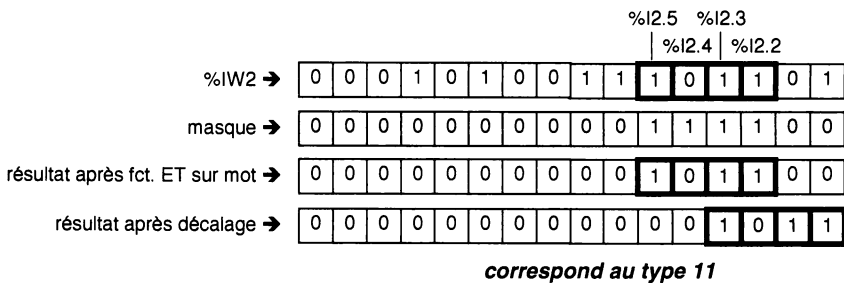
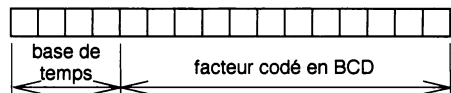


figure D-14

- Paramétrage de la durée de trémie (exemple en Step5)

En Step5, le format de temporisation est le suivant :



- la base de temps est codée sur les 4 bits de gauche : 0000 = 1/100° sec.
0001 = 1/10° sec.
0010 = 1 sec.
0011 = 10 sec.
- le facteur est codé en BCD sur 3 quartets, sa valeur maximale est donc 999
- la durée réelle est facteur x base de temps (la précision de la tempo équivaut à la base de temps)

Différentes manipulations sont donc nécessaires :

- 1) lecture du mot associé à l'opération de trempe à réaliser
- 2) isolement et transcodage du facteur de temps
- 3) ajout de la base de temps
- 4) départ de la temporisation sur activation de X72

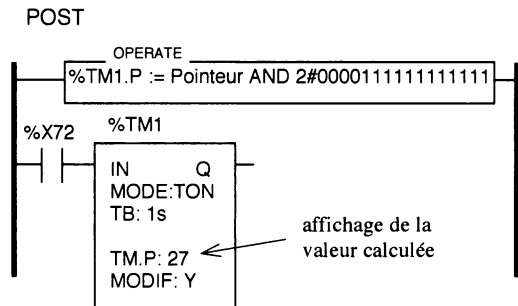
```

L -pointeur
L KM0001111111111111
UW
T MW50
SPA FB241
L MW50
L KM0010000000000000
OW
T MW50
U M7.2
L MW50
SE T1
  
```

Le premier masque, en combinaison avec la fonction ET SUR MOT, permet d'isoler le facteur de temps contenu dans le pointeur. FB241 est un bloc fonctionnel intégré qui permet le transcodage de valeurs décimales en codes BCD : il suffit de paramétrer le mot d'entrée et ensuite un mot + un octet de sortie lors de la saisie du programme (MW50 est le mot d'entrée dans cet exemple et étant donné que la valeur est limitée à 999, seulement le mot de sortie est significatif : on choisit MW50). Le deuxième masque, en combinaison avec la fonction OU SUR MOT, permet d'ajouter la base de temps (1 sec. dans cet exemple).

- Paramétrage de la durée de trempe

La temporisation est réalisée par un bloc fonction, par exemple %TM1, qu'il suffit de paramétrer : TB est la base de temps, directement exprimée dans sa valeur, TM.P est le facteur exprimé en décimal. Celui-ci peut être modifié par configuration (avec la console) ou par programme (ce qui est l'objectif souhaité). Il faut également isoler au préalable le facteur de temps contenu dans le pointeur. On utilisera ensuite %TM1.Q dans la réceptivité X72 → X73.



Dans les deux cas, tous les octets des mémoires initiales et de travail sont remplacés par des mots.

2 - PREPARATION DE MEDICAMENTS

2 - 1. Présentation du sujet

Le pharmacien d'un hôpital est responsable de la préparation des boîtes de médicaments prêtes à l'emploi qui répondent aux traitements prescrits par les médecins. Le patient est guidé jour par jour :

- la première rangée de la boîte contient les médicaments du premier jour et ainsi de suite,
- les 7 rangées d'une boîte correspondent à une semaine,
- aucun traitement n'excède 3 semaines,
- un lot comporte toujours 3 boîtes au cas où la deuxième semaine par exemple ne nécessite aucun médicament (exemple du traitement n°2 ci-dessous).

Un certain nombre de traitements standards sont fréquents. Des traitements spécifiques peuvent être créés au fur et à mesure du besoin. Pour garantir la traçabilité des interventions médicales, il est indispensable de mémoriser le traitement prescrit à chaque patient.

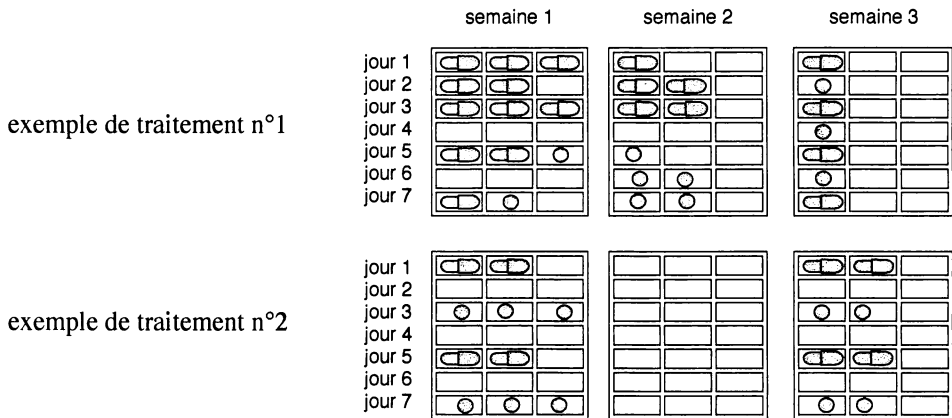


figure D-15

- Reconditionnement préalable

Jusqu'à 120 références, limitées à des gellules, pillules et comprimés, interviennent dans les traitements. Pour que la préparation des boîtes puisse être réalisée par un système automatique, il est nécessaire que les médicaments soient au préalable reconditionnés dans des emballages standards (ce poste de reconditionnement n'est pas étudié). Ces nouveaux emballages se présentent sous forme de rouleaux marqués d'un code-barre qui sont ensuite installés sur un magasin. Chaque numéro de rouleau a sa place réservée dans le magasin.

2 - 2. Fonctionnement du système (figure D-16)

Les médicaments sont prélevés individuellement par le dispensateur au fur et à mesure du besoin. A chaque fois, le code-barre imprimé sur le rouleau est comparé au numéro de médicament demandé : en cas d'erreur, l'opération est interrompue et une alarme est déclenchée (si le rouleau est placé au mauvais emplacement par exemple).

Selon le traitement à préparer, le bras manipulateur saisit un à un les médicaments dans le dispensateur et les dépose dans les boîtes aux emplacements adéquats.

Le bouchage hermétique des boîtes et la pose des étiquettes d'identification se font ensuite sur un poste indépendant (non étudié).

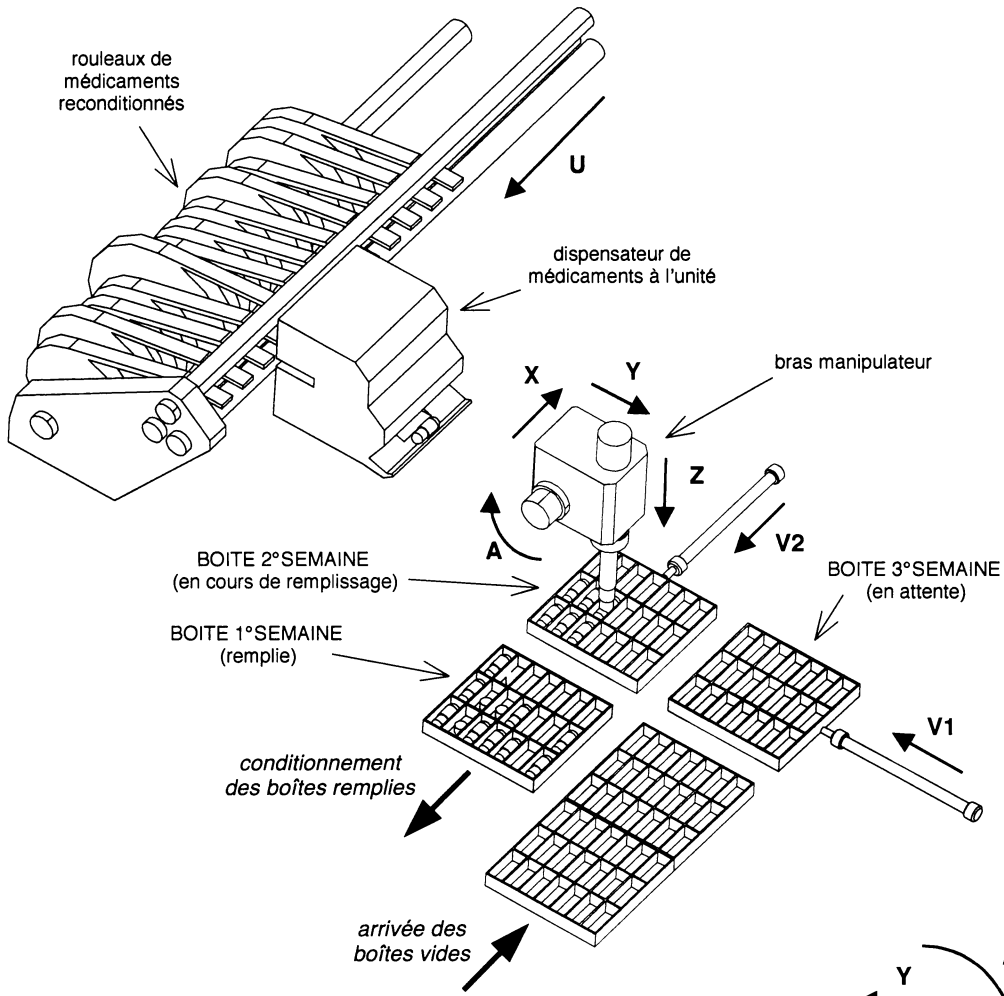


figure D-16

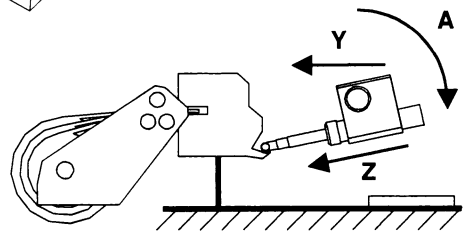


figure D-17

La figure D-17 montre le bras manipulateur vu de profil entrain de saisir un médicament dans le dispensateur. Un cycle complet pour un lot de boîtes est le suivant :

- amenée d'une boîte vide à l'aire de remplissage (vérin V1)
- positionnement X et Y du bras manipulateur et rotation A pour l'amener dans l'axe du médicament à saisir, Z étant en position arrière
- pendant ce temps : préparation du magasin de rouleaux à la bonne position et prélevement d'un médicament par le dispensateur
- lorsque ces deux opérations sont terminées : avancée suivant Z et saisie d'un médicament (l'organe de préhension n'est pas étudié)
- retour de Z, puis de A
- positionnement X et Y en fonction de l'endroit où déposer le médicament
- avancée de Z, dépose, retour de Z
- mise en place de tous les médicaments de la boîte
- évacuation de la boîte par le vérin V2
- préparation des deux boîtes suivantes

2 - 3. Enoncé du sujet

a) Question 1 : premier niveau d'analyse

- Faire l'inventaire des fonctions à un premier niveau d'analyse et les détailler.
- Rechercher les séquences et tracer l'épure du grafcet de commande.

b) Question 2 : structures de données et grafcet de coordination

On prévoit de mémoriser jusqu'à un million de prescriptions puisque la nature exacte du traitement administré à chaque patient doit être conservée.

- Proposer la structure matérielle du système de commande ainsi que l'organisation générale des mémoires. Justifier.
- Proposer une interface utilisateur.
- Tracer la structure de données.
- Prévoir la gestion des données.
- Tracer le grafcet de coordination complet.

c) Question 3 : analyse détaillée des tâches

On note X+ l'avance du bras suivant cet axe et X- son recul. L'axe est positif dans le sens de la flèche sur la figure D-16. On note tous les autres mouvements selon le même principe. Les fins de course de V1 sont notés v11 en position avancée et v10 en position reculée. On note de même les fins de course des mouvements V2, A et Z. Les axes U, X et Y sont contrôlés par codeurs incrémentaux.

Magasin : le pas entre les rouleaux est de 30 mm. Il correspond à 100 points du codeur incrémental qui contrôle l'axe du magasin. L'origine de l'axe U se situe à 30 mm à gauche du premier rouleau de gauche. Les numéros des rouleaux correspondent aux numéros des médicaments et sont placés dans l'ordre croissant dans le magasin.

Bras manipulateur : les empreintes des boîtes ont un entre-axe de 20 mm suivant X et de 15 mm suivant Y. Le dispensateur se trouve dans le prolongement des empreintes du milieu. Suivant Y, le déplacement nécessaire entre le dispensateur et la rangée du premier jour est de 40 mm. Les codeurs suivant X et Y délivrent 10 points par mm de déplacement. Leur origine est la suivante :

- suivant X : 20 mm à gauche de la colonne d'empreintes de gauche
 - suivant Y : en position de prise de médicament dans le dispensateur
- Traiter l'alarme.
 - Choisir des symboles pour les variables significatives et les définir.
 - Tracer les grafcets de tâches au niveau pré-actionneur.

2 - 4. Corrigé et commentaires

a) Premier niveau d'analyse (Q.1)

- *Inventaire et définition des fonctions*

palettiser

déposer

saisir

prélever

figure D-18

Fonction : palettiser

Cette fonction consiste à évacuer la boîte qui vient d'être remplie et ensuite à présenter une boîte vide à l'aire de remplissage.

Fonction : déposer

Le bras manipulateur se positionne au-dessus de l'emplacement où il doit déposer le médicament. Le bras descend, le médicament est déposé et le bras remonte.

Fonction : saisir

Quelle que soit la position du bras, il se positionne en face du dispensateur. Le bras avance et le médicament est saisi s'il est disponible. Puis le bras recule. Il bascule ensuite en position verticale.

Fonction : prélever

En fonction de la référence de médicament demandée, le magasin se positionne pour que le rouleau adéquat soit engagé dans le dispensateur. Le code du rouleau est vérifié. S'il est correct, un médicament est prélevé et mis à disposition du bras manipulateur. Si le code n'est pas correct, l'alarme est déclenchée et le mode de production est interrompu. Un opérateur change les rouleaux mal placés et retire les boîtes éventuellement remplies ainsi que la boîte qui était en cours de remplissage. Après acquittement du défaut, le cycle complet est relancé.

- Recherche des séquences

Les fonctions telles qu'elles ont été définies, conduisent à la structure suivante :

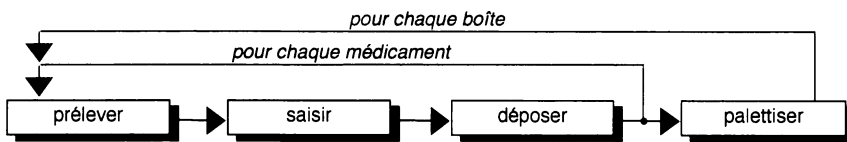


figure D-19

Une analyse plus fine montre que la cadence peut être augmentée en scindant la fonction *saisir* en deux parties dont la première (notée *préparer le bras*) peut être réalisée en même temps que la fonction *prélever*. Le grafcet au niveau *fonction* de la figure D-20 traduit le cycle complet.

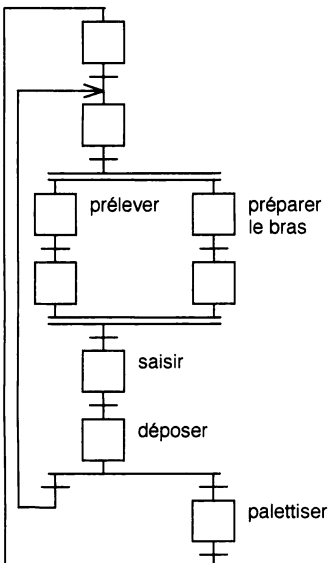


figure D-20

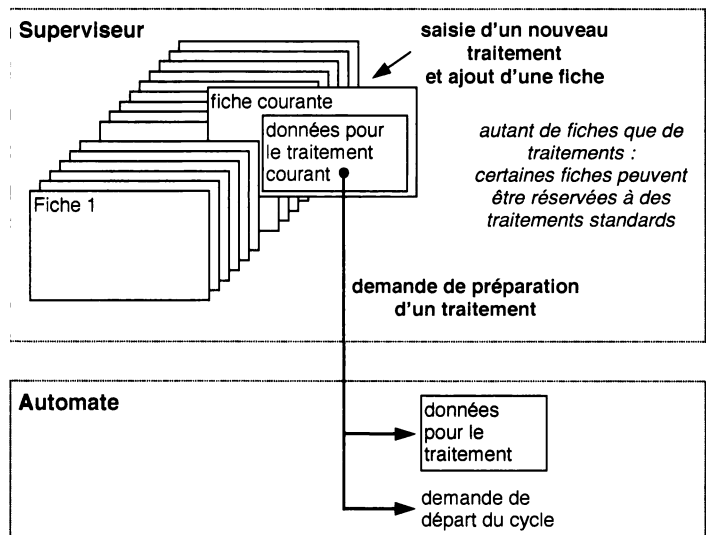


figure D-21

b) Structure de données et grafset de coordination (Q.2)

La quantité d'informations à archiver n'est pas compatible avec la taille mémoire d'un automate. Les données sont gérées dans ce cas par un superviseur en ayant recours à une recette. Chaque fiche de cette recette est réservée à un patient et comporte toutes les informations : nature exacte du traitement, identité complète du patient...

La mise en forme des données est faite par le superviseur en fonction de la nature du traitement définie par le médecin et les données d'identification du patient. Seules les informations relatives au traitement à préparer sont transmises à l'automate au début de chaque cycle.

Voir la figure D-21.

Un fragment de l'interface utilisateur est présenté figure D-22. La configuration des boîtes est représentée à l'écran et pour chaque emplacement, deux champs de saisie sont disponibles : l'un pour le numéro du médicament, l'autre pour son nom en clair. Chaque fiche est identifiée par son numéro auquel est associée l'identité du patient.

The image shows a graphical user interface for a medical system. It features several input fields and buttons. At the top, there is a field for 'numéro patient' with the value '001.356' and two radio buttons labeled 'appeler' and 'enregistrer'. To the right, the text 'identité patie.' is partially visible. Below this is a field for 'numéro traitement pré-régulé' with the value '000.000' and similar radio buttons. Underneath, it says 'semaine 1'. The main part of the interface is a grid for medication scheduling. It has two columns for 'jour 1' and 'jour 2'. For 'jour 1', the first row shows '083' in the first column and '083' in the second, with 'trevlielos 1000' written below each. For 'jour 2', the first row shows '049' in the first column and an empty field in the second, with 'carbo...' written below the first column.

figure D-22

La structure de données détaillée est la même pour chaque fiche de la recette du côté du superviseur et est également reproduite du côté de l'automate (figure D-23).

Toutes les lignes de la mémoire peuvent être scrutées systématiquement les unes après les autres :

- si les pointeurs affichent des valeurs non nulles, ils servent à désigner le médicament à saisir et l'endroit de sa dépose,
- si les pointeurs sont nuls, aucune saisie de médicament n'a lieu,
- après 21 de ces opérations, la tâche de palettisation échange les boîtes,
- au bout de 3 palettisations, le cycle complet est terminé : le superviseur doit en être informé.

Les fonctions liées à la gestion des données sont les suivantes :

- enregistrement des données issues du superviseur en début de cycle,
- décalage de la file d'attente systématiquement après chaque opération de saisie-dépose,

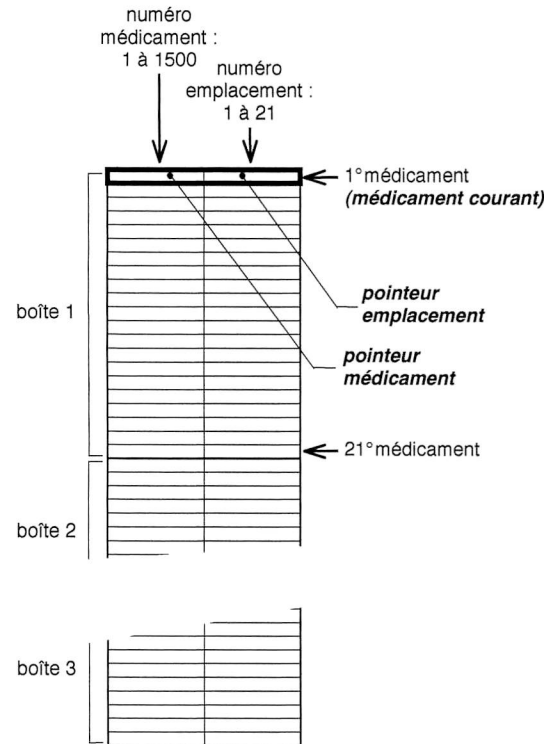


figure D-23

- incrémentation d'un compteur à chaque opération (noté Cop),
- incrémentation d'un second compteur à chaque nouvelle boîte (noté Cb).

Les compteurs sont initialisés en début de cycle. Le compteur Cop est de plus réinitialisé à chaque nouvelle boîte.

- Grafcet de coordination complet

Le cycle démarre sur demande du superviseur. Celui-ci transmet en même temps les données relatives au traitement à préparer: une première étape consiste à les enregistrer.

Les pointeurs correspondent alors au premier médicament à déposer. Les 21 décalages successifs permettent le remplissage de la première boîte.

Le compteur des opérations est réinitialisé et le compteur de boîtes est incrémenté.

Lorsque les 3 boîtes ont été remplies, le cycle est terminé et le superviseur en est informé.

Le saut d'étapes X2 → X10 a lieu lorsqu'une opération doit rester sans effet. De ce fait, toutes les 63 positions de la file d'attente sont scrutées.

Remarque : le nombre de décalages étant géré par compteurs, il n'est pas utile de forcer la dernière position à zéro.

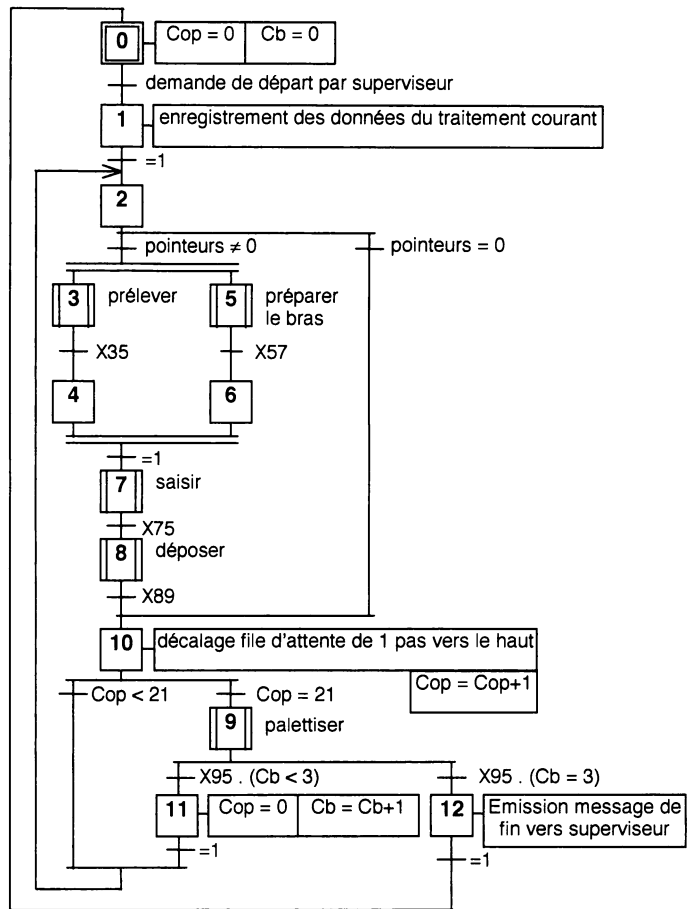


figure D-24

c) Analyse détaillée des tâches (Q.3)

- Alarme

L'alarme est générée par l'étape X33 de la tâche *prélever* en cas d'erreur de code (figure D-28). Le grafcet de surveillance X90 est de niveau hiérarchique le plus élevé et commande l'interruption du cycle grâce au forçage de tous les autres grafcets dans la situation vide. Après intervention de l'opérateur (changement de place des rouleaux par exemple et enlèvement des boîtes), le cycle complet doit reprendre. Tous les grafcets de tâche sont réinitialisés et le grafcet de coordination du mode de production doit démarrer à condition d'en refaire la demande à partir du superviseur. Le grafcet de surveillance complet se limite aux étapes de la figure D-25.

L'architecture hiérarchisée est résumée par le diagramme suivant :

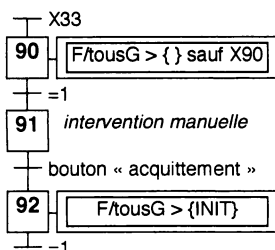


figure D-25

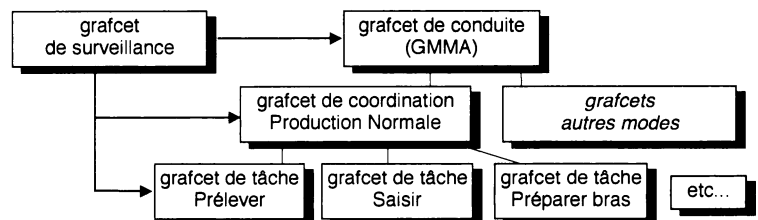


figure D-26

- Définition des variables

Les variables numériques sont de deux natures :

- les compteurs qui évoluent avec les codeurs incrémentaux, ils sont notés Cu, Cx et Cy.
- les valeurs utiles de la structure de données, il s'agit :
 - du numéro de médicament, noté Nm : la position réelle à atteindre suivant l'axe U est :

$$\{Nm * pas \text{ des rouleaux}\}$$
 - de la position de dépose du médicament. Elle est déduite des numéros de colonnes et de lignes, notées Pmx et Pmy. Ces numéros sont calculés par le superviseur en fonction de la place occupée par chaque médicament : le pointeur *emplacement* est donc composé de deux variables.

Selon X, la coordonnée réelle est :

$$\{Pmx * pas \text{ suivant X}\}$$

Selon Y, elle est :

$$(40 \text{ mm} + \{Pmy * pas \text{ suivant Y}\})$$

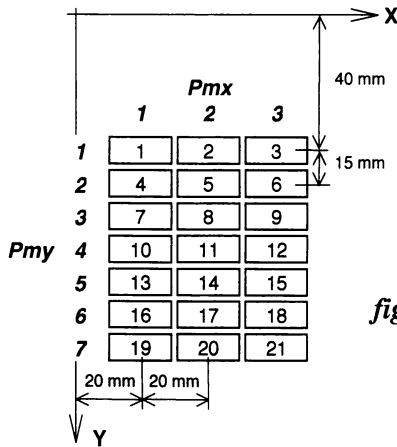


figure D-27

- Grfcets de tâches

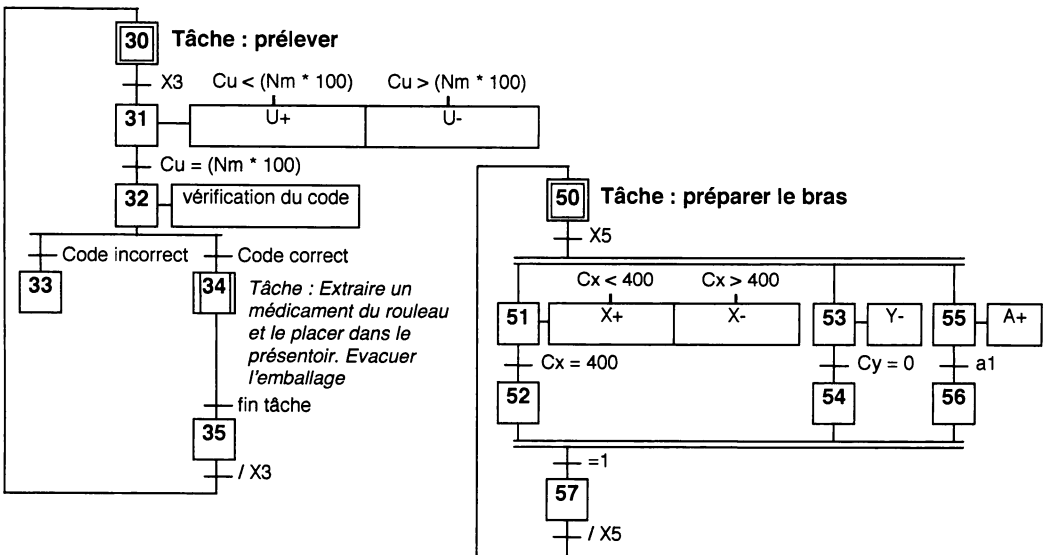


figure D-28

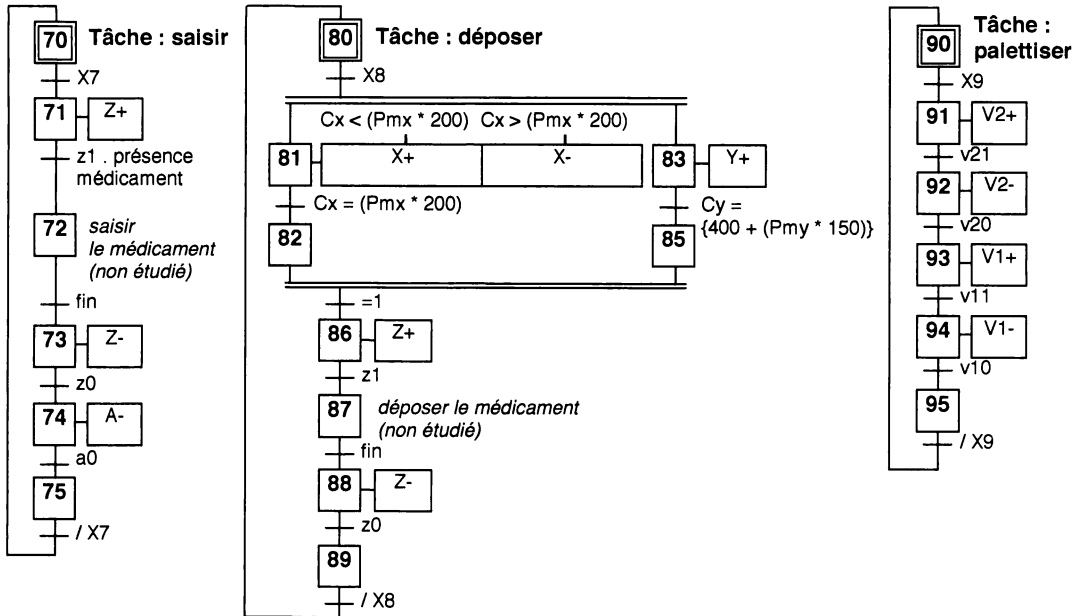


figure D-29

Remarque 1 : la discrimination du sens de marche des axes numériques peut également se faire par sélection de séquences. Cette solution avait été proposée en corrigé de l'exercice précédent (bacs de trempe).

Remarque 2 : l'écriture des grafjets au niveau *automate* mérite quelques précautions, illustrées par l'exemple suivant.

Supposons qu'un médicament doit être saisi et que le bras se trouve au-dessus de l'empreinte de gauche. Un déplacement suivant X+ positionne le bras en face du dispensateur. Le passage de l'étape X51 vers l'étape X52 (qui provoque l'arrêt du bras) se fait lorsque la coordonnée X400 est atteinte. Or l'inertie mécanique du bras et le temps interne de traitement de l'automate ne permettent pas l'arrêt *immédiat* du bras : un léger dépassement a lieu. Si le médicament doit ensuite être déposé dans la deuxième rangée (alignée avec le dispensateur), Pmx est = 2. L'étape X81 provoquera ainsi un léger retour suivant l'axe X. Ce mouvement n'est pas nécessaire et réduira la durée de vie du moteur. Pour éviter cela, on peut adapter la réceptivité en aval de X81 comme suit :

$$R_{81-82} = [Cx \geq \{(Pmx * 200) - 10\}] \cdot [Cx < \{(Pmx * 200) + 10\}]$$

si l'on considère que le dépassement est proche de 1 mm sans dépasser cette valeur.

De surcroît, le positionnement sera de meilleure précision car l'arrêt est commandé 10 points avant la coordonnée à atteindre, ce qui compense le défaut de dépassement.

3 - CELLULE DE POINÇONNAGE-CONTROLE

3 - 1. Présentation du sujet

La cellule comporte 3 poinçonneuses à commande numérique et un poste de contrôle automatisé. Un robot est chargé de dispatcher des pièces en tôle en fonction de leur type. Un code-barre imprimé sur chaque tôle permet leur reconnaissance à l'entrée de la cellule (en position de saisie par le robot) et également au niveau des trois poinçonneuses.

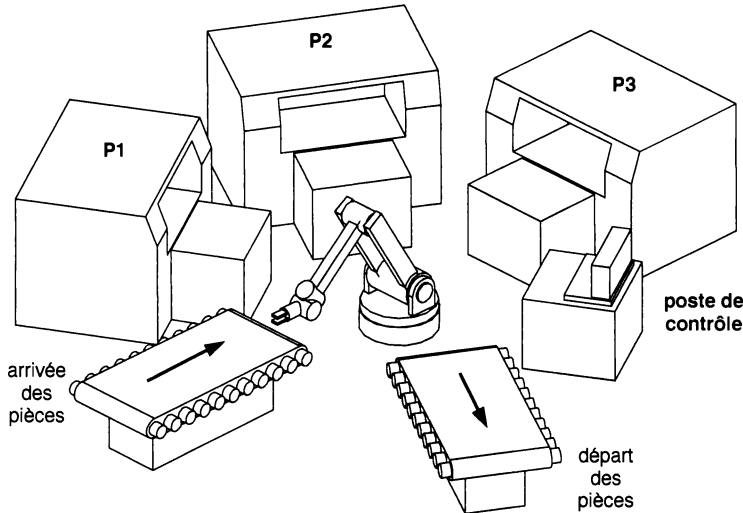


figure D-30

3 - 2. Fonctionnement du système

Selon le type de tôle, le Bureau des Méthodes définit si la gamme nécessite une ou deux opérations :

- l'opération (a) est obligatoire \Leftrightarrow passage sur l'une des trois poinçonneuses
- l'opération (b) n'est pas obligatoire \Leftrightarrow passage sur le poste de contrôle

Pour l'opération (a), le Bureau des Méthodes définit les poinçonneuses qui sont autorisées à recevoir les différents types de pièces. Les choix possibles sont :

- le passage est autorisé sur la machine 1
- le passage est autorisé sur la machine 2
- le passage est autorisé sur la machine 3
- le passage est autorisé sur la machine 1 ou sur la machine 2
- le passage est autorisé sur la machine 1 ou sur la machine 3
- le passage est autorisé sur la machine 2 ou sur la machine 3
- le passage est autorisé sur l'une des trois machines

Pendant le fonctionnement, lorsque plusieurs machines sont permises, la sélection définitive se fait selon leur disponibilité. Cette condition étant évolutive, son résultat varie en temps réel.

Un numéro de programme doit en outre être affecté à chaque opération de chaque type de pièce. Ce numéro sera pris en compte par la commande de la machine en question lors des opérations de poinçonnage et de contrôle.

On souhaite pouvoir modifier les gammes sans devoir provoquer l'interruption de la production. Le taux d'occupation de la cellule doit être le plus élevé possible afin d'assurer une productivité maximale.

3 - 3. Architecture du système

Les commandes du robot, des 3 poinçonneuses et du poste de contrôle sont toutes reliées directement aux entrées-sorties d'un automate qui gère l'ensemble de la cellule. Par ailleurs, les commandes sont reliées à un serveur de programmes par l'intermédiaire d'un réseau (figure D-31).

3 - 4. Énoncé du sujet

a) Question 1 : étude dynamique

Soit la situation suivante :

- arrivée de la pièce 1 (de type 9, seule la machine 1 est autorisée, le contrôle n'est pas demandé),
- arrivée de la pièce 2 (de type 3, toutes les machines sont autorisées, le contrôle est demandé),
- arrivée de la pièce 3 (de type 9).

Envisager plusieurs évolutions différentes en faisant varier la fréquence d'arrivée des pièces et tracer un graphique pour deux d'entre-elles. On précise que les temps des opérations sont différents selon le programme exécuté. Conclure sur la prévisibilité des cycles.

b) Question 2 : analyse

- Faire l'inventaire des fonctions et les détailler. Le niveau d'analyse doit être suffisamment précis pour pouvoir mettre en place les mouvements du robot.
- Rechercher les séquences et tracer l'épure des grafjets.

c) Question 3 : structures de données et grafjets

Les pièces sont regroupées en 20 familles. On souhaite pouvoir affecter un millier de programmes différents aux opérations de poinçonnage et autant au contrôle.

- Tracer la structure des données initiales.
- Prévoir la gestion des données.
- Tracer les grafjets complets.

d) Programmation

Si le lecteur ne dispose pas du manuel de référence du constructeur, il se reportera directement au corrigé.

On envisage de décentraliser la commande en affectant un micro-automate à chaque constituant de la cellule (figure D-32). Un réseau local véhicule alors les signaux de synchronisation.

A titre d'exemple, traiter la demande de départ cycle de la machine 1 (le téléchargement des programmes de poinçonnage n'entre pas dans le cadre de cette étude). Proposer une solution pour le réseau Fipway.

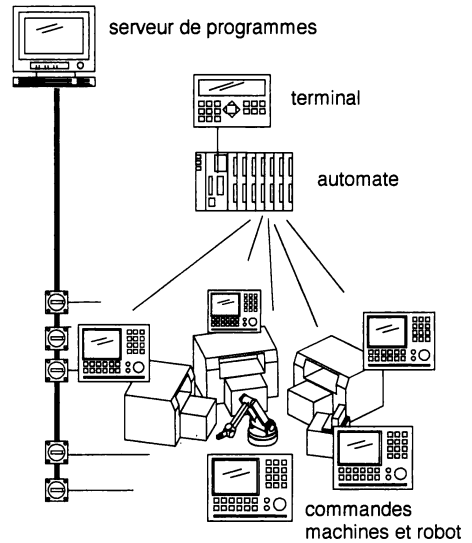


figure D-31

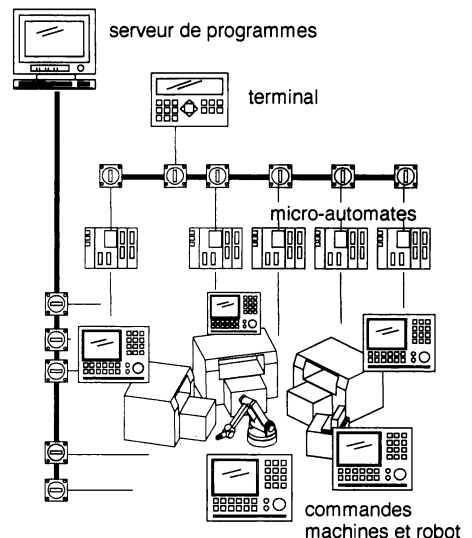


figure D-32

3 - 5. Corrigé et commentaires

a) Etude dynamique (Q.1)

L'axe du temps des diagrammes de la figure D-33 évolue de haut en bas. En situation initiale, aucune pièce n'est présente à l'entrée de la cellule. L'arrivée de la première pièce déclenche le fonctionnement. Les deux simulations diffèrent en raison de la fréquence d'arrivée des pièces : un rythme lent pour la première simulation, un rythme plus soutenu pour la seconde.

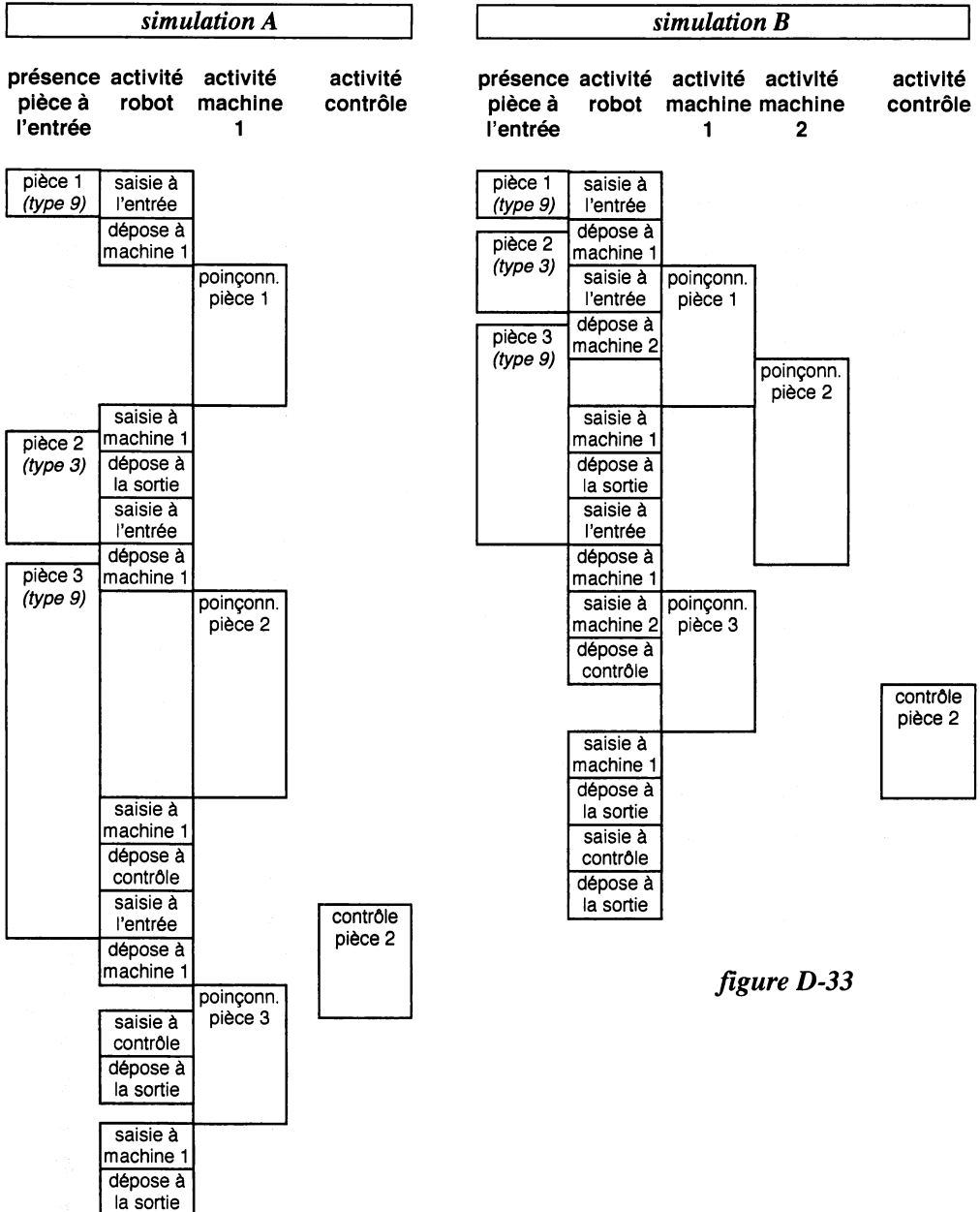


figure D-33

Un nombre quasi illimité de simulations est possible. On se heurte à des difficultés si l'on essaie de traduire le fonctionnement du robot par un grafcet de description selon la méthode traditionnelle. En effet, l'observation du robot dans différentes simulations ne permet pas de dégager de séquence *a priori*.

On constate également que l'activité des machines ne suit pas la même évolution dans les deux cas. Par ailleurs, la pièce n°2, de type 9, n'est pas traitée par la même poinçonneuse. Dans d'autres simulations, elle pourrait également être traitée par la machine 3. Toutes les combinaisons sont susceptibles de se produire à un moment où à un autre si la cellule fonctionne suffisamment longtemps.

Même s'il est de type déterministe (les mêmes effets provoquant les mêmes causes), le système n'a pas un comportement prévisible. La fréquence d'arrivée des pièces, leur ordre d'apparition, les durées d'exécution des programmes, la disponibilité au fil du temps des machines, du poste de contrôle et du robot sont autant de paramètres qui influencent la progression des activités.

b) Analyse (Q.2)

Malgré le caractère aléatoire du fonctionnement, il faut construire le programme de l'automate qui commande la cellule. Procédons selon la méthode proposée.

- Décomposition fonctionnelle

Un premier niveau de décomposition permet d'isoler le robot et chaque poste de travail puisque ces cinq constituants sont indépendants et ne sont pas synchronisés de manière immuable. On trace un grafcet autonome pour chacun d'eux.

Les 4 postes

Chaque poste peut avoir 3 états différents :

- il est *libre* (aucune pièce ne se trouve sur le poste, le poste est à l'arrêt),
- il est en *travail* (un cycle de poinçonnage ou de contrôle est en cours),
- il est en *attente* de déchargement (le cycle de travail est terminé mais la pièce est encore présente sur le poste).

Ces 3 états s'enchaînent toujours de la même façon. Le tracé des grafcets correspondants est donc immédiat :

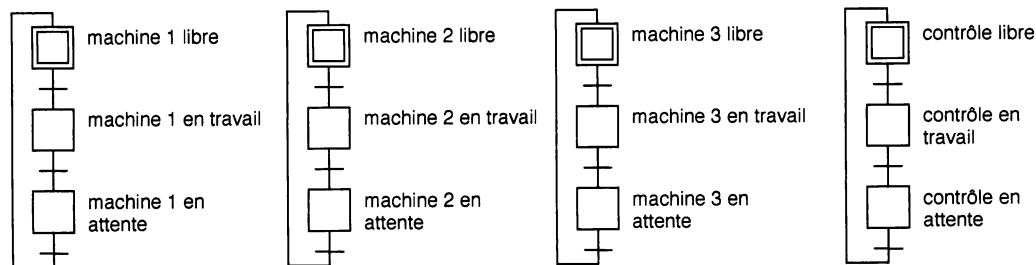


figure D-34

Le robot

De manière similaire à l'étude de cas du chapitre VI (cellule double), deux phases se succèdent invariablement : la saisie d'une tôle puis la dépose de la tôle, puis à nouveau une saisie. Le grafcet de commande du robot est ainsi construit très globalement en deux phases :

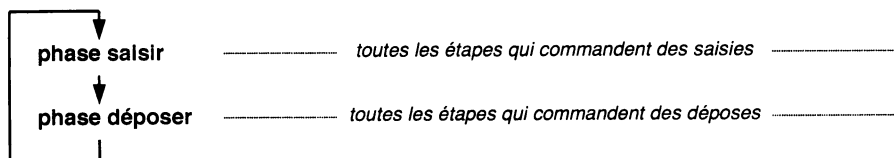


figure D-35

Les fonctions liées au robot, de niveau de précision suffisant pour sa gestion, sont les suivantes :

PHASE SAISIR		PHASE DEPOSER	
SE	saisir à l'entrée de la cellule	DP1	déposer au poste 1
SP1	saisir au poste 1	DP2	déposer au poste 2
SP2	saisir au poste 2	DP3	déposer au poste 3
SP3	saisir au poste 3	DC	déposer au poste de contrôle
SC	saisir au poste de contrôle	DS	déposer à la sortie de la cellule

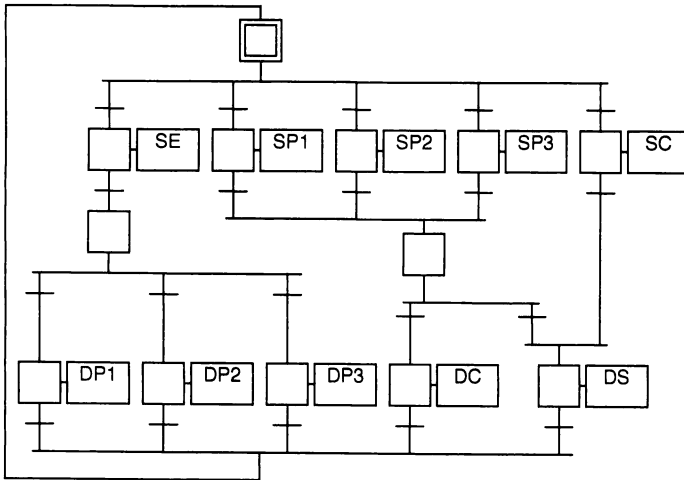


figure D-36

La recherche des séquences obligées permet de tracer simplement une première épure du grafset de commande du robot. Toutes les saisies sont autorisées lorsque le robot est disponible. Lorsqu'une saisie a eu lieu à l'entrée de la cellule, la dépose se fait obligatoirement sur l'une des poinçonneuses. Après saisie sur une poinçonneuse, la tôle peut être déposée au poste de contrôle ou être évacuée directement. Après saisie sur le poste de contrôle elle est obligatoirement évacuée. Aucune contrainte ne conduit à exécuter une saisie particulière après une dépose.

c) Structures de données et grafsets (Q.3)

- Mémoire Initiale

Chacune des 20 familles de pièces nécessite jusqu'à 2 opérations. La première opération est paramétrée et la deuxième est facultative. Un numéro de programme est affecté individuellement à chaque opération. On propose la mémoire initiale suivante :

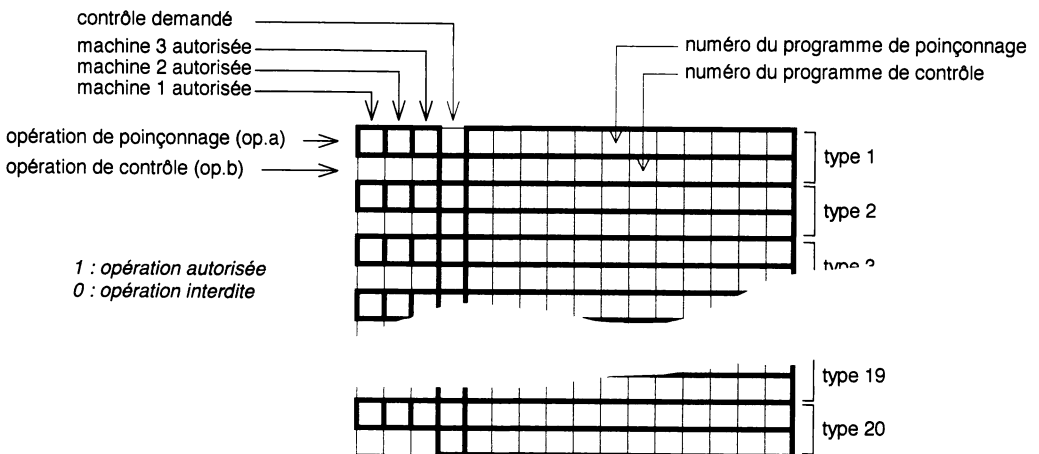


figure D-37

Les 4 bits de poids fort permettent de spécifier la nature des opérations. Pour les opérations de poinçonnage, on a soit un seul, soit 2, soit 3 bits à un. Les numéros de programmes sont codés sur les 12 bits de poids faible.

- Gestion des données

L'utilisation d'une mémoire de travail facilite l'écriture des réceptivités en réduisant le nombre de paramètres à tester. La mise à jour d'une seule opération est suffisante, la signification des 4 bits de poids le plus fort étant fonction du type d'opération en cours.

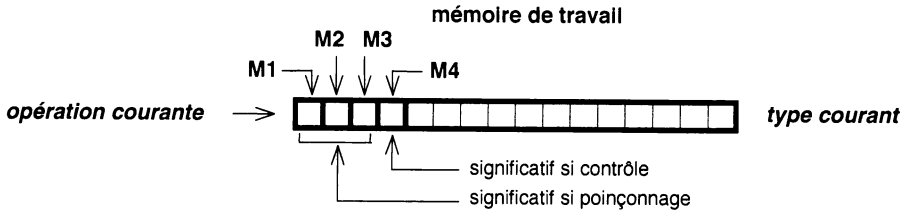


figure D-38

La sélection des informations à dupliquer dépend du type de tôle et du type d'opération à réaliser.

- Grafecets complétés

Synchronisation entre le robot et les postes

Lorsque le robot a déposé une tôle sur un poste, celui-ci doit démarrer en utilisant le numéro de programme indiqué par la mémoire de travail. A cet effet, des étapes en aval de celles qui commandent les déposes sont ajoutées au grafecet du robot (il s'agit de X74, X76, X78 et X80).

Lorsque le programme est terminé, et quelle que soit sa durée, l'étape de compte-rendu du poste est activée (X12, X22, X32 ou X42). Elle génère une demande d'évacuation, interprétée par le grafecet de commande du robot au niveau des réceptivités en amont de X54, X56, X58 et X64. Lorsque la pièce est enlevée, le grafecet du poste est réinitialisé pour indiquer qu'il est de nouveau libre.

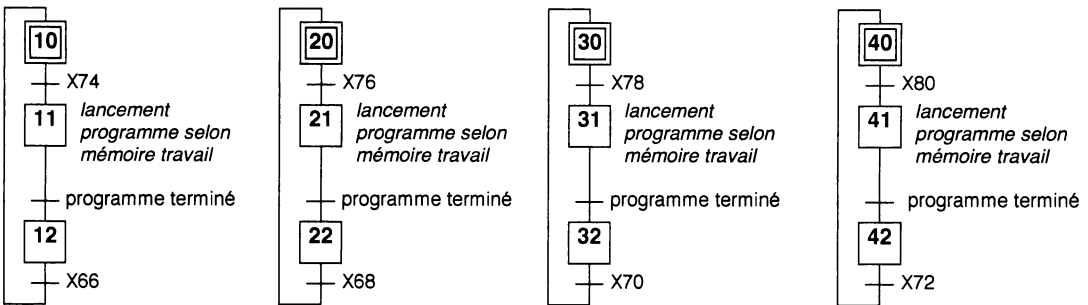


figure D-39

Fonctionnement du robot

Le fonctionnement normal de la cellule est autorisé ou non par un grafecet de niveau hiérarchique supérieur.

Pour éviter de bloquer la cellule, on retient le principe qui interdit au robot de saisir une pièce tant que sa dépose ne peut pas être garantie :

- dans le cas où une saisie est envisagée à l'entrée de la cellule, il est donc nécessaire d'identifier le type de pièce (étape X52), de consulter quels sont les postes de poinçonnage autorisés et vérifier la liberté de ces derniers (réceptivités en aval de l'étape X53),
- si une saisie est envisagée sur une poinçonneuse, il faut vérifier si, pour le type de pièce en question, l'opération de contrôle est demandée, et, le cas échéant, vérifier la disponibilité de ce poste (réceptivités en aval des étapes X55, X57 et X59).

En cas d'échec, aucune action n'a lieu : le grafecet est immédiatement réinitialisé. Au niveau *automate*, il faut veiller à ce que le grafecet puisse quitter la boucle [X50-51-52-53] afin de permettre l'évolution vers les séquences X54, X56, X58 ou X64.

Afin de ne pas surcharger la figure, la gestion des priorités n'est pas représentée. On prévoira par exemple de favoriser l'entrée d'une pièce dans la cellule plutôt qu'une évacuation pour maximiser le taux d'occupation des machines et donc la cadence générale.

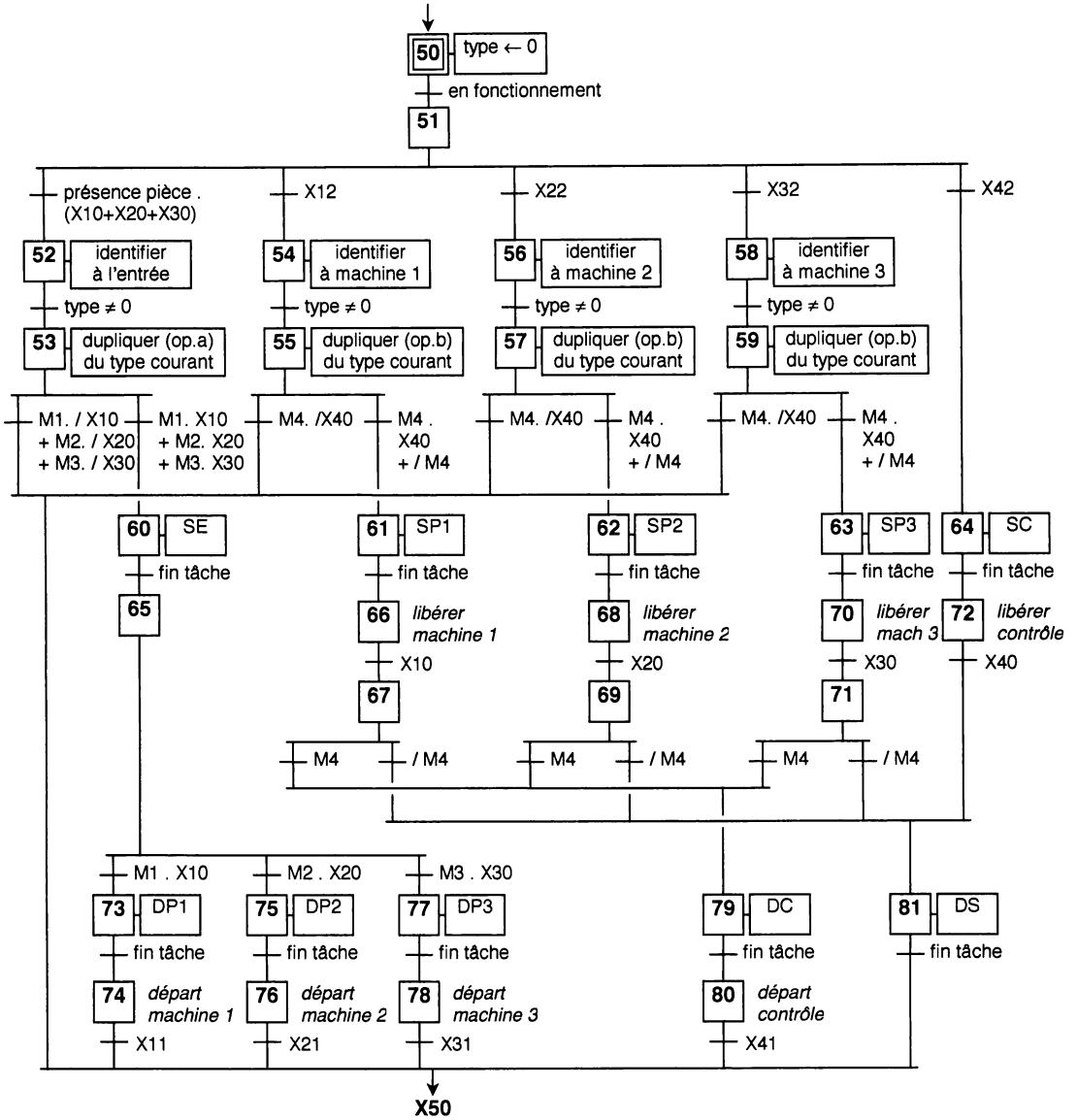


figure D-40

d) Programmation

Dans la nouvelle configuration de la figure D-32, les graficets ci-dessus sont tous implantés dans des automates différents. On demande de traiter le départ cycle de la machine 1 :

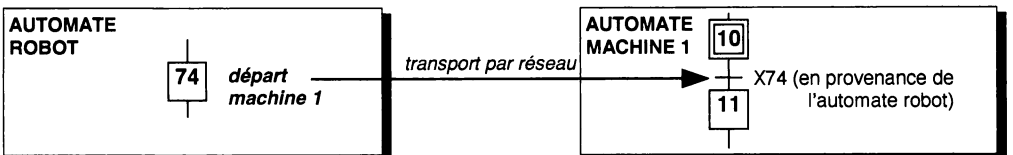


figure D-41

Réseau local Fipway

Il existe 3 manières de programmer des échanges d'informations entre des équipements raccordés à un réseau Fipway :

- les instructions Uni-Telway (READ_VAR pour lire un objet, WRITE_VAR pour positionner un objet...),
- les mots communs (chaque station dispose de 4 mots %NW qui sont diffusés sur tout le réseau et donc exploitables par toutes les autres stations),
- la table partagée.

L'utilisation de la table partagée permet une programmation transparente, comme si les données manipulées se trouvaient toutes dans chaque automate. On configure dans chaque automate une table qui débute toujours à la même adresse et qui est toujours de même longueur. Cette table est accessible en lecture par tous les automates. Par contre, chaque automate ne dispose que d'une plage limitée spécifique pour l'accès en écriture. La configuration de l'exemple pourrait être la suivante :

Automate robot	Automate machine 1	Automate machine 2	Automate machine 3	Automate contrôle
%MW200 ↓ %MW209	%MW200	%MW200	%MW200	%MW200
	%MW210 ↓ %MW219	%MW220 ↓ %MW229	%MW230 ↓ %MW239	%MW240 ↓ %MW249
	%MW249			
		%MW249	%MW249	
	%MW249			

Chaque automate a accès en lecture aux adresses %MW200 à %MW249. Les zones mémoires en gras indiquent les adresses d'écriture pour chaque automate. Ainsi, les ordres de départ des machines que doit émettre l'automate du robot sont inscrits dans la zone %MW200 à %MW209. Le dialogue demandé peut être programmé ainsi :

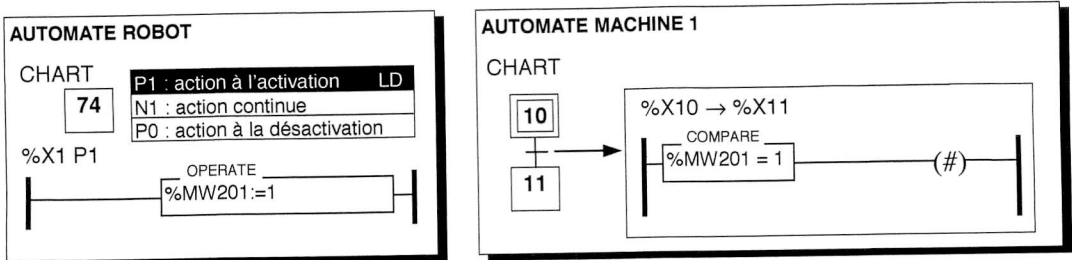
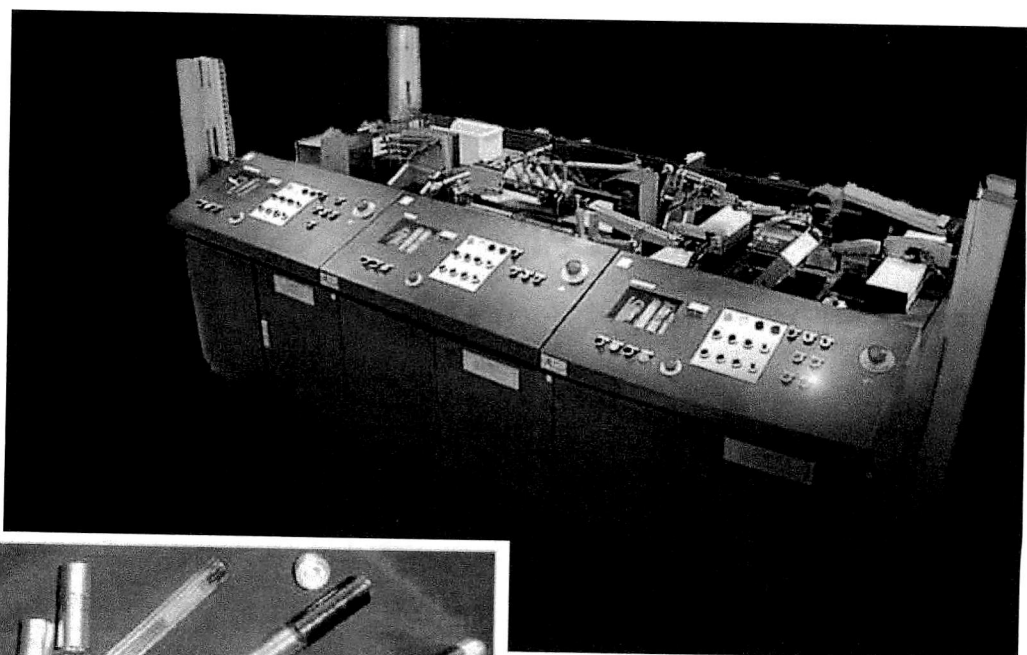
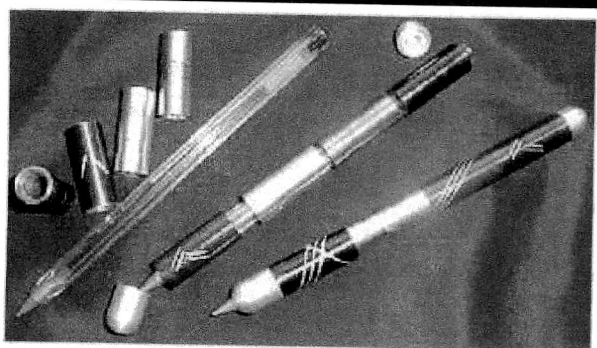


figure D-42

Dans l'automate robot, il faut repositionner %MW201 à Zéro après désactivation de X74. On utilisera %MW202 à %MW204 respectivement pour les machines 2 et 3 et le poste de contrôle.



Laboratoire didactique



Stylos à décor personnalisable

X . APPLICATION

1 - PRESENTATION DU LABORATOIRE

1 - 1. La machine et son environnement

Ce laboratoire didactique développé à l'*IUT de Mulhouse, dépt. GMP*, est une unité de production automatisée complète destinée à l'enseignement dans le domaine des automatismes, de la supervision, des réseaux locaux, de la gestion de production intégrée et du pilotage de systèmes. Tous les concepts abordés dans le présent ouvrage, des éléments de base aux notions les plus élaborées, y sont mis en application pratique en milieu pédagogique à travers un produit support : des stylos à décor personnalisable.

1 - 2. L'architecture du système et le flux de production

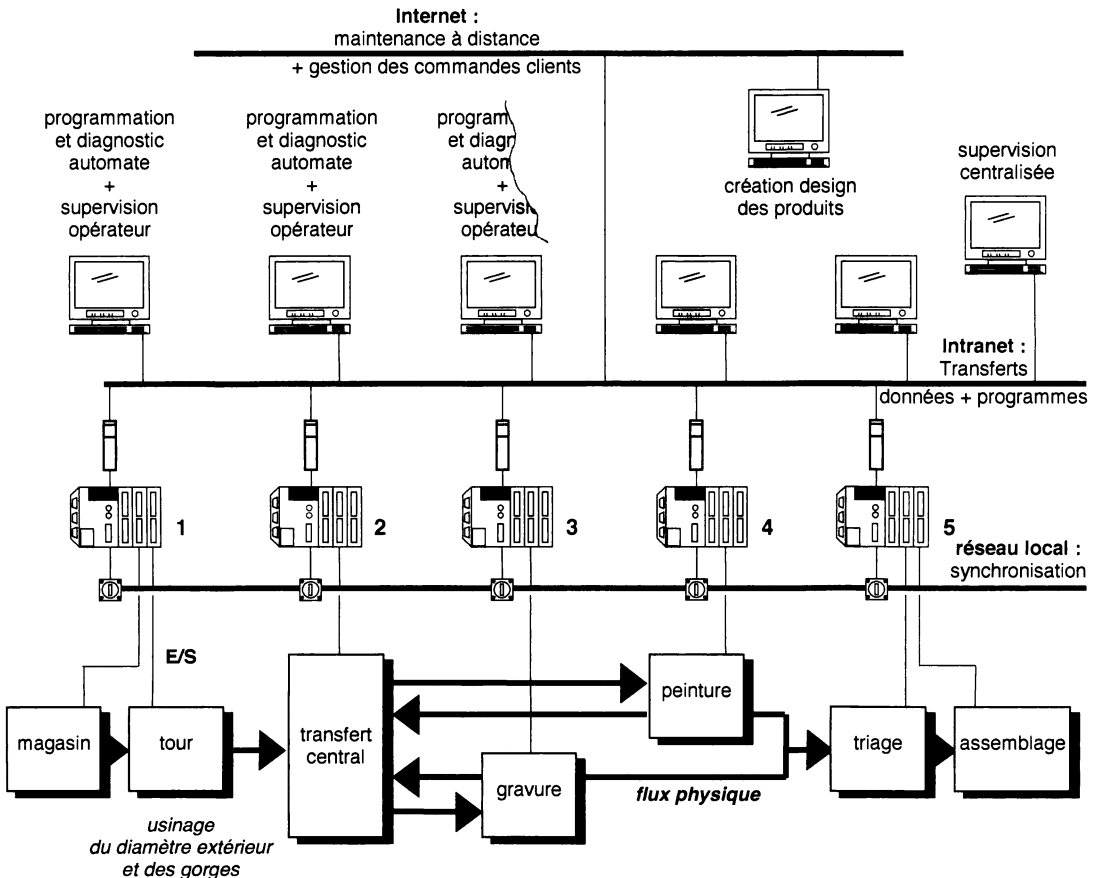


figure D-43

1 - 3. Les 5 sous-systèmes

Chaque élève dispose d'un sous-système complet :

- un poste informatique,
- un automate programmable,
- un pupitre qui comporte des boutons poussoirs, interrupteurs, potentiomètres et voyants programmables ainsi que des éléments câblés (arrêt d'urgence, mise en/hors énergie, volt-mètres...),
- une partie opérative complète : capteurs, actionneurs et effecteurs utilisables dans le cadre de la fabrication de stylos aussi bien que pour des travaux pratiques variés.

Les cinq automates communiquent par réseau local industriel pour assurer la synchronisation de production. Ils sont connectés à l'Intranet ainsi que les 5 postes de programmation-diagnostic et les 2 postes supplémentaires, l'un dédié à la supervision centralisée et l'autre à la création du design des produits (également disponible à distance par Internet). Chacun des 5 postes-élèves dispose également d'un superviseur de type « opérateur ». La connexion du système à Internet permet la télémaintenance grâce à des pages web implantées dans les coupleurs des automates et permet la relation avec les clients, réception des commandes et suivi de production. Il est entendu qu'il s'agit d'un exercice pédagogique et que les produits ne sont pas réellement commercialisés. Tous les procédés ont soigneusement été choisis dans cette perspective, au détriment de performances de productivité qui seraient ici sans objet.

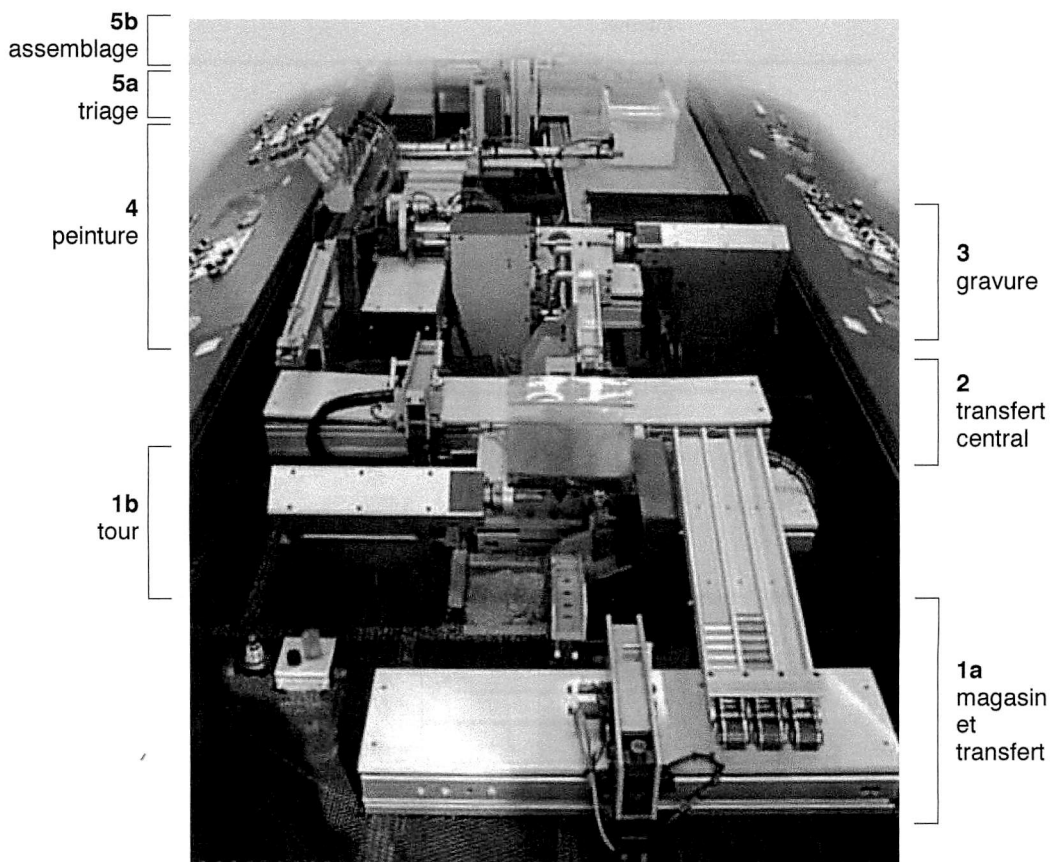


figure D-44

1 - 4. Les modes d'utilisation

Le laboratoire est conçu pour répondre à 3 modes d'utilisation :

<i>modes d'utilisation</i>	<i>mise en oeuvre</i>	<i>objectifs pédagogiques</i>
production effective de biens de consommation	programmes complets prêts à l'emploi	<ul style="list-style-type: none"> • conduite de systèmes automatisés • maintenance • gestion de production avec simulation de goulets, gestion des stocks... • lecture et modifications mineures de programmes • analyse de grafquets, de variables, de programmes • fonctions de diagnostic, configuration automate...
élaboration progressive d'applications	programmes entièrement à développer travaux pratiques basés sur des programmes préparés à modifier ou à compléter <i>dans cette configuration, tout le potentiel de la machine n'est pas obligatoirement utilisé : ceci permet une approche très progressive des difficultés</i>	<ul style="list-style-type: none"> • mise en oeuvre de l'algèbre logique • prise en compte de la sécurité câblée et programmée • édition en langages normalisés CEI 61131-3 • mise en évidence de la scrutation cyclique ou périodique, des interruptions, des mémoires-images... • utilisation de symboles, de tables d'animation, de forçages, de pages de mise au point et de configuration, des données système... • apprentissage du Grafcet et de la gestion des modes de marches et d'arrêts • grafquets hiérarchisés et paramétrés • fonctions : temporisateurs, compteurs, structures de données • comptage rapide, tâche rapide, traitements événementiels et algorithmiques, interfaces analogiques, horodatage, formats numériques et alphanumériques... • communication par réseau local • édition d'applications de supervision : configuration de tables et de variables, paramétrage d'objets, scripts en langage C, recettes, événements...
développements	apport de compléments aux programmes existants	réservés aux activités de longue durée de type projet : optimiser la gestion de production, affiner les modes de défaillances, ajouter des fonctions de supervision...

1 - 5. Le produit

Le produit support est un stylo à décor personnalisable de conception originale, spécialement étudié pour un usage didactique. Le processus consiste à fabriquer des bagues décoratives et à en habiller des stylos standards du commerce. Un stylo comporte 4 bagues. Des embouts montés serrés garantissent le maintien de l'ensemble.

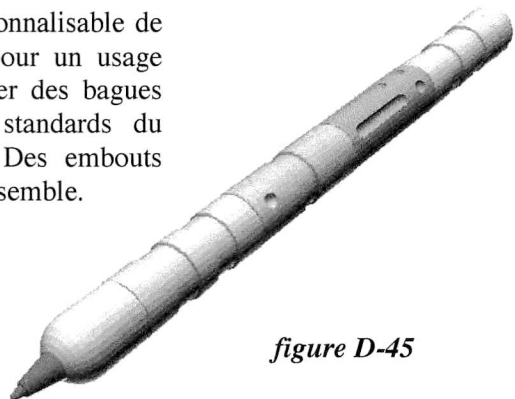


figure D-45

En fonction des préférences du client, chaque bague du stylo peut être :

- soit en alliage d'aluminium, soit en laiton,
- usinée ou non de gorges,
- colorée ou non (le client sélectionne les couleurs de son choix),
- gravée d'un motif pré-réglé ou entièrement créé par le client (les motifs sont limités à une combinaison de points et de lignes).

On peut choisir l'aspect des motifs parmi deux possibilités :

- si la gravure a lieu avant la mise en couleur, le motif est de la même couleur que le restant de la bague,
- si le processus est inverse, c'est le matériau de la bague qui apparaît.

Ce dernier paramètre aura une importance considérable sur le flux de production et apporte un intérêt particulier à la gestion du processus.

Enfin, la combinaison de l'assemblage des bagues est également définie par le client.

1 - 6. La création des modèles

La création des différents modèles par les clients se font chez les marchands, situés n'importe où dans le monde, grâce à l'écran de saisie présenté ci-dessous. Les données sont transférées à l'unité de production via Internet. Un écran complémentaire permet de sélectionner les motifs pré-réglés. En raison de la vocation exclusivement pédagogique du laboratoire, un certain nombre d'écrans supplémentaires permettent l'analyse en temps réel des variables du programme en complément de la documentation.

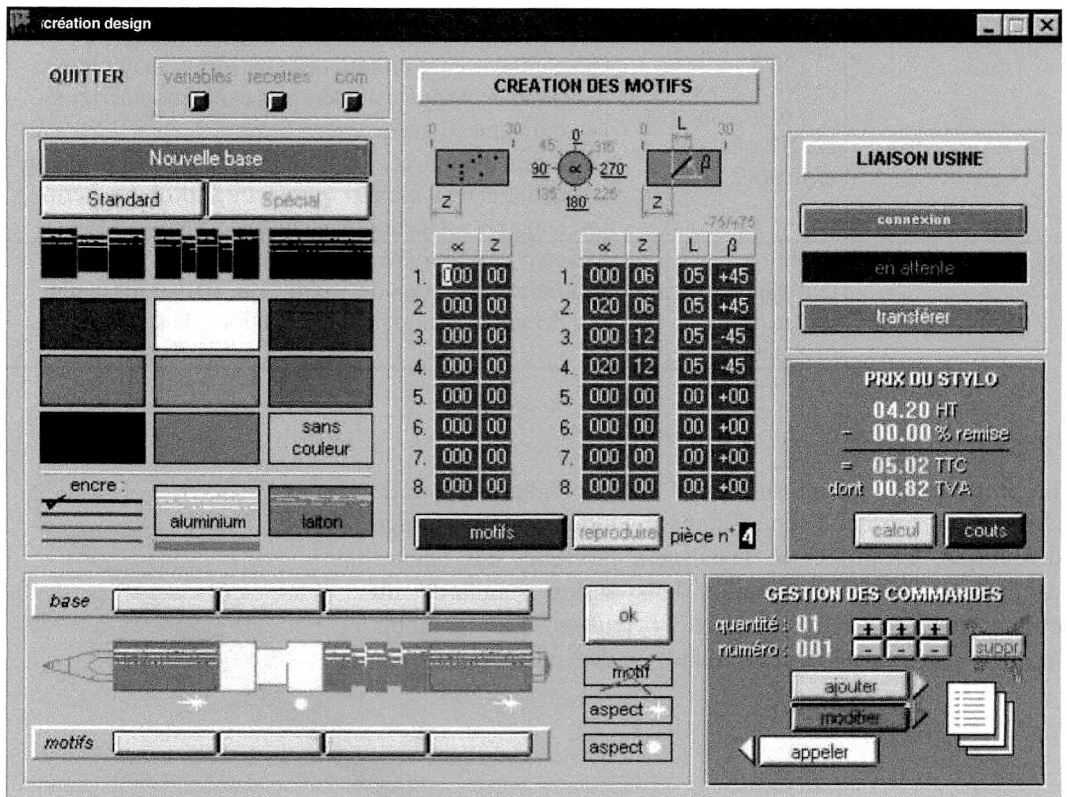


figure D-46

Cet écran d'édition est divisé en plusieurs zones :

- une zone où s'affiche le design du stylo au fur et à mesure de sa création,
- une zone où l'on sélectionne les paramètres de base : la morphologie des bagues et leurs couleurs,
- une zone d'édition des motifs,
- une zone de calcul du prix des stylos,
- une zone de gestion des commandes,
- une zone de dialogue avec le superviseur de l'unité de production,

A cela s'ajoutent les possibilités de définir les aspects de chaque bague, la couleur de l'encre du stylo et d'appeler les pages annexes (sélection des motifs pré-réglés, mise à jour des coûts, analyse du programme en temps réel)

2 - LE SUIVI DE PRODUCTION

Ce chapitre donne une vue synthétique du flux des informations qui permettent de gérer la production d'une manière automatique. Au lancement d'une campagne de production, certaines données de production sont transférées du superviseur vers les automates. D'autres données sont transmises en temps réel au fur et à mesure des besoins.

Numérotation des gammes

Le cheminement d'une pièce dans l'unité de production dépend de la gamme de fabrication à réaliser. Les gammes dépendent :

- de la présence ou non de motif,
- de la présence ou non d'une couleur,
- de l'aspect (ordre gravure-peinture ou peinture-gravure).

	phase 10	phase 20	phase 30	phase 40
gamme 1	tournage	assemblage		
gamme 2	tournage	gravure	assemblage	
gamme 3	tournage	peinture	assemblage	
gamme 4	tournage	gravure	peinture	assemblage
gamme 5	tournage	peinture	gravure	assemblage

Numérotation des destinations

Sur les postes *transfert central*, *gravure* et *peinture*, les pièces peuvent prendre des chemins différents. On numérote les destinations possibles de la manière suivante :

	poste de destination			
	transfert central	gravure	peinture	trriage-assemblage
numéro de destination	2	3	4	5

Ce numéro sert à orienter les pièces à la sortie des postes de travail. Il fait référence au numéro de l'automate de destination (voir la figure D-43).

2 - 1. Automate 1 : magasin des bruts et opération de tournage

Ordre de sortie magasin d'une pièce brute

La file d'attente « magasin » peut être complétée par un ordre de fabrication (O.F.) selon 2 moyens :

- automatiquement par l'automate 5
si la file d'attente est vide, l'automate 1 émet une autorisation vers l'automate 5. En fonction du besoin, un O.F. est alors généré et transmis à l'automate 1

- par insertion manuelle d'un O.F.
une commande effectuée sur le superviseur local permet d'insérer jusqu'à 6 O.F. qui seront prioritaires par rapport aux O.F. générés par l'automate 5

Dans tous les cas, le matériau est sélectionné grâce au tableau d'association type-matériau. Ce tableau est mis à jour par le superviseur en début de campagne, et ce en fonction des caractéristiques du stylo.

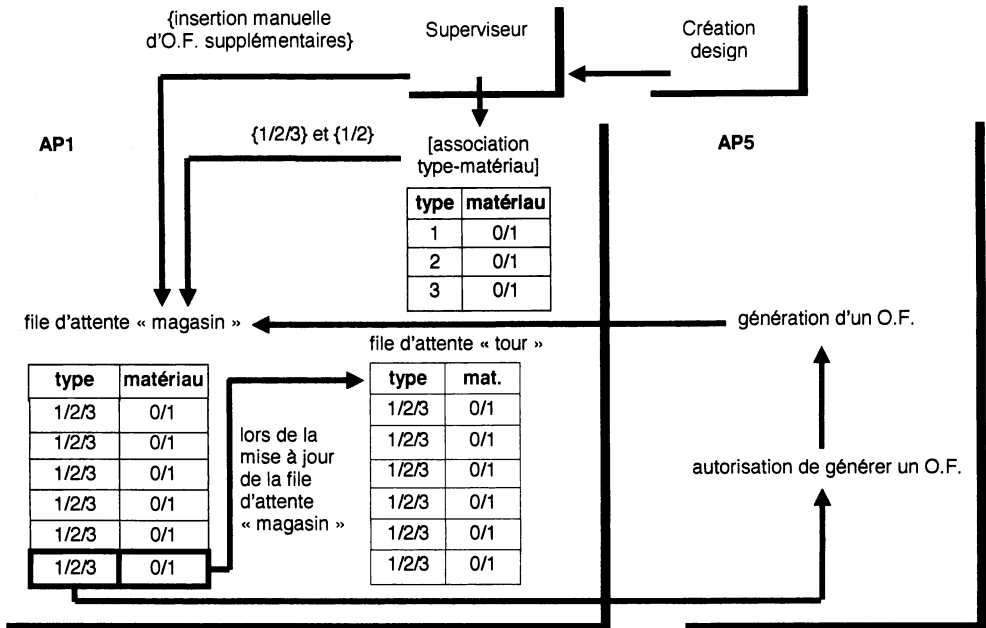


figure D-47

Transport de la pièce brute vers le tour

La pièce brute est dirigée vers le tour et la file d'attente « magasin » est mise à jour.

Lien entre les files d'attente

La file d'attente « tour » contient les mêmes informations, dans le même ordre, que la file d'attente « magasin ». Le retard que peut cependant prendre la file d'attente « tour » provient du fait qu'il est possible que le chariot qui prélève les pièces dans le magasin peut être amené à exécuter plusieurs cycles pendant l'exécution d'un seul cycle d'usinage (le chariot et le tour ne sont pas synchronisés). L'empilement de la file d'attente « tour » se fait lors de la mise à jour de la file d'attente « magasin ».

Sélection du type d'usinage de gorge

Des valeurs indiquées par le pointeur de la file d'attente « tour » dépendent :

- le type d'usinage : 1 gorge pour les pièces de type 1,
2 gorges pour les pièces de type 2,
aucune gorge pour les pièces de type 3.
- le mode de pulvérisation : avec pulvérisation pour l'alliage d'aluminium,
sans pulvérisation pour le laiton.

Lorsque le tour évacue une pièce vers le transfert central, l'automate 1 émet vers l'automate 2 une information qui lui signale l'arrivée d'une pièce. De plus, la file d'attente « tour » est mise à jour à ce moment.

2 - 2. Automate 2 : transitique au transfert central

Sélection de la pièce à prélever

L'information qui indique qu'une pièce est évacuée du tour porte la valeur numérique 1 en considération du numéro de l'automate émetteur. Elle est enregistrée dans la file d'attente de l'automate 2 au même titre que des informations similaires émises par les automates 3 et 4 (voir les § suivants). Le poste auquel une pièce doit être prélevée est déterminé par lecture du pointeur de cette file d'attente.

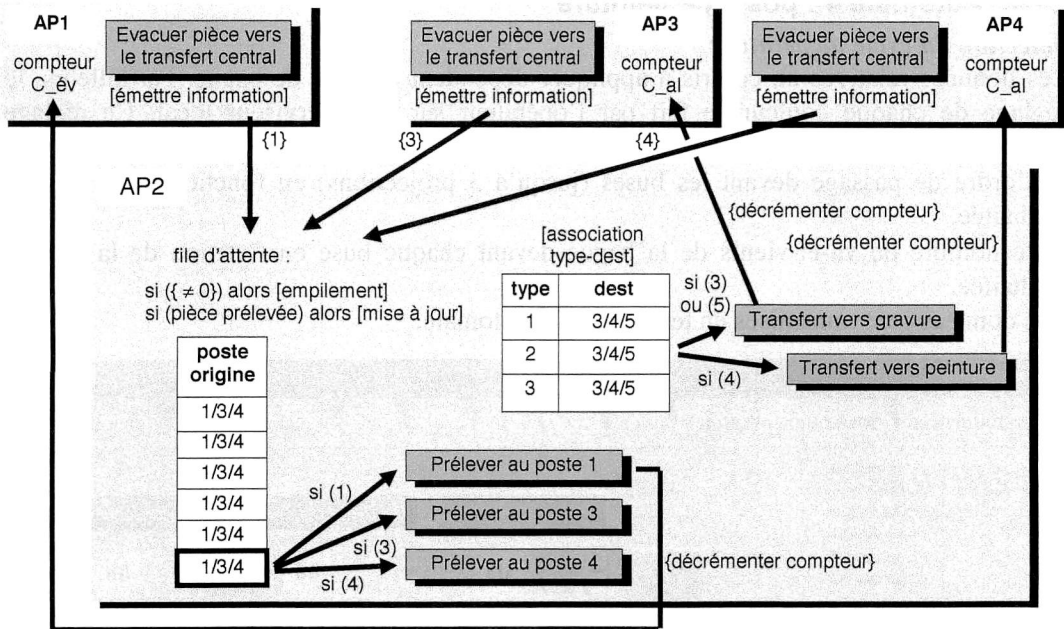


figure D-48

Lorsque le chariot de transfert a prélevé une pièce à la sortie du tour, l'automate 2 émet un ordre de décrémentation vers l'automate 1. Cet ordre de décrémentation permet la mise à jour du compteur relatif au stock d'évacuation du tour.

Sélection de la destination de la bague prélevée

- Si la bague est en provenance du tour, c'est le *numéro de destination* qui lui est associé qui définit quelle doit être sa goulotte de destination.
- Si la bague est en provenance de la peinture, elle rejoint systématiquement le poste de gravure.
- Si elle provient du poste de gravure, elle rejoint systématiquement le poste de peinture.

Avant de libérer la bague vers le stock d'alimentation des postes de gravure et de peinture, l'automate 2 vérifie si ces stocks ne sont pas saturés. Lorsque la bague est déposée, il émet un ordre d'incrémation vers l'automate correspondant.

2 - 3. Automate 3 : poste de gravure

Sélection du motif à graver

Au début de *chaque* cycle de gravure, l'automate 3 demande au superviseur local de lui transmettre les coordonnées du motif à graver en fonction du type de bague en présence.

Sélection du poste de destination

A la fin de la gravure, la bague est évacuée soit directement vers le poste de triage, soit de nouveau vers le transfert central à destination de la peinture (dans ce dernier cas, la valeur numérique 3 est émise vers l'automate 2).

La sélection du chemin se fait en fonction du numéro de destination indiqué par le superviseur local en début de campagne.

2 - 4. Automate 4 : poste de peinture

Sélection du cycle de peinture

Les données relatives au coloris à appliquer dépendent du type de bague. Par ailleurs, le réglage de chaque couleur se fait par l'opérateur sur le superviseur local. Ce réglage concerne :

- l'ordre de passage devant les buses (jusqu'à 3 projections) en fonction de la nuance souhaitée,
- le nombre de va-et-vients de la bague devant chaque buse en fonction de la densité souhaitée.

Ces données sont transmises en temps réel à l'automate.

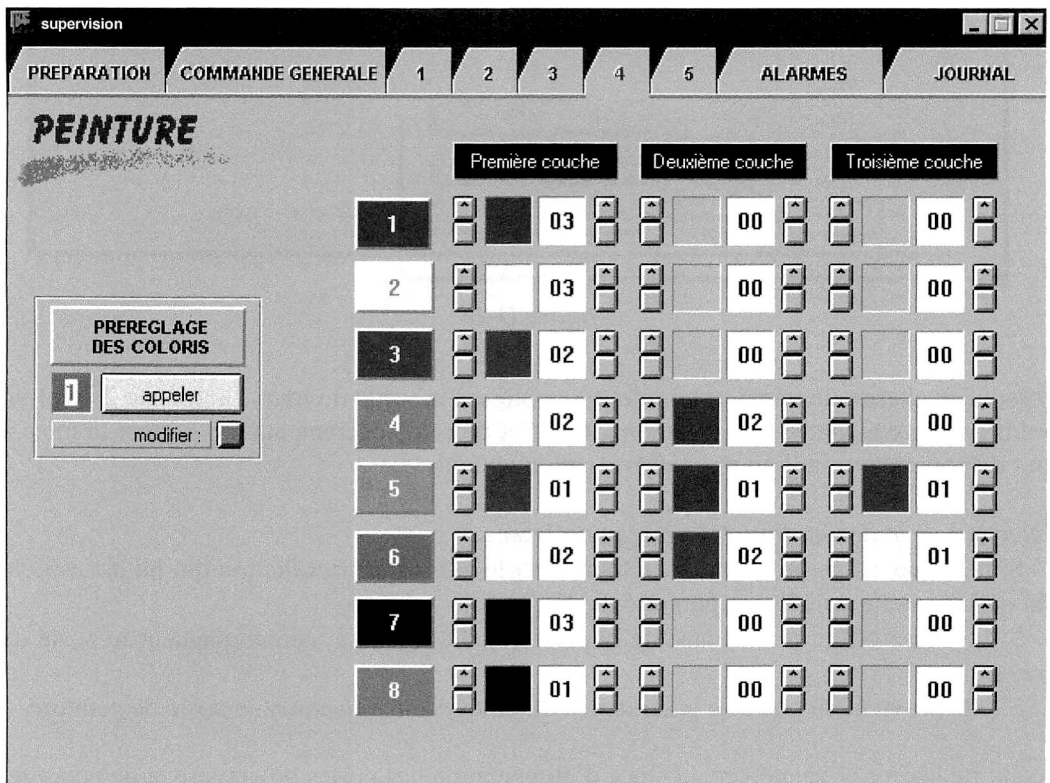


figure D-49

Sélection du poste de destination

A la fin de la peinture, la bague est évacuée soit directement vers le poste de triage, soit de nouveau vers le transfert central à destination de la graveuse (dans ce dernier cas, la valeur numérique 4 est émise vers l'automate 2).

La sélection du chemin se fait en fonction du numéro de destination indiqué par le superviseur en début de campagne.

2 - 5. Automate 5 : assemblage du stylo

Empilement des bagues décoratives

- Préparation de l'empilement par le manipulateur :

L'empilement à réaliser est déterminé en fonction du stylo tel qu'il a été créé par le client. Une table de données contient la liste des bagues à monter sur le stylo. La sélection du type de bague à prélever est déterminée par le pointeur de la file d'attente si des bagues sont présentes.

- Emission des ordres de réapprovisionnement vers l'automate 1 :

L'automate 1 prend l'initiative d'une requête lorsque sa file d'attente « magasin » est vide. Ces requêtes sont prises en compte :

- et génèrent des O.F. dans l'ordre de la composition du stylo,
- tant que le nombre de stylos n'est pas atteint.

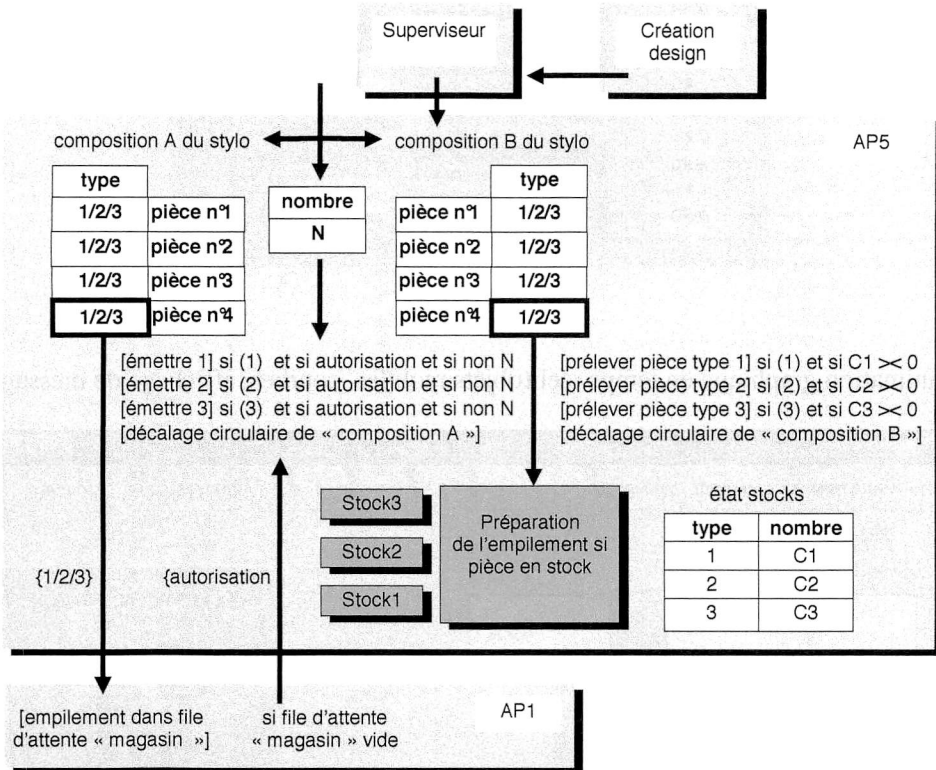


figure D-50

Etat des stocks « bagues finies »

Les compteurs C1, C2, et C3 « état stocks » sont régulièrement tenus à jour. A chaque fois qu'une bague est ajoutée au stock, le compteur correspondant est incrémenté. Lorsqu'une bague est prélevée, le compteur correspondant est décrémenté.

Assemblage final

Les quatre bagues décoratives sont empilées automatiquement selon la configuration voulue. Le déroulement de cette opération est tributaire de l'état des stocks.

Les embouts sont ensuite montés sur le stylo d'une manière semi-automatique : l'opérateur prépare l'ensemble et une presse réalise le sertissage.

3 - LA SUPERVISION

Plusieurs fonctions de supervision ont déjà été abordées ci-dessus. Elles se complètent de divers écrans qui illustrent les services habituellement rendus par cet outil :

- le pilotage centralisé de l'unité de production (ci-dessous un extrait de cette page)

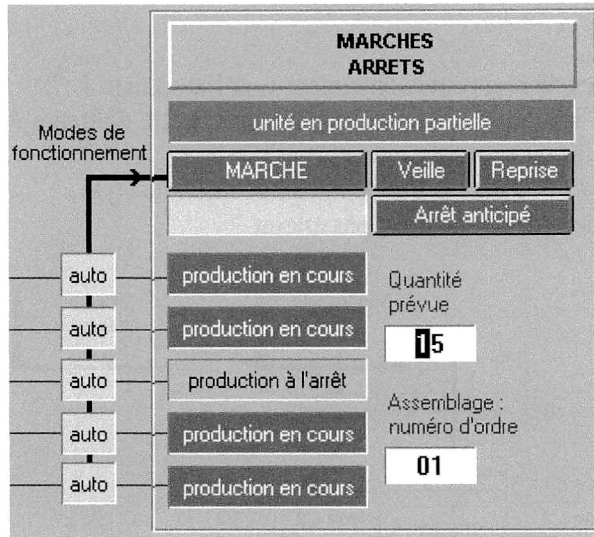


figure D-51

- L'animation graphique en temps réel (objets mobiles, courbes, affichage de messages...)

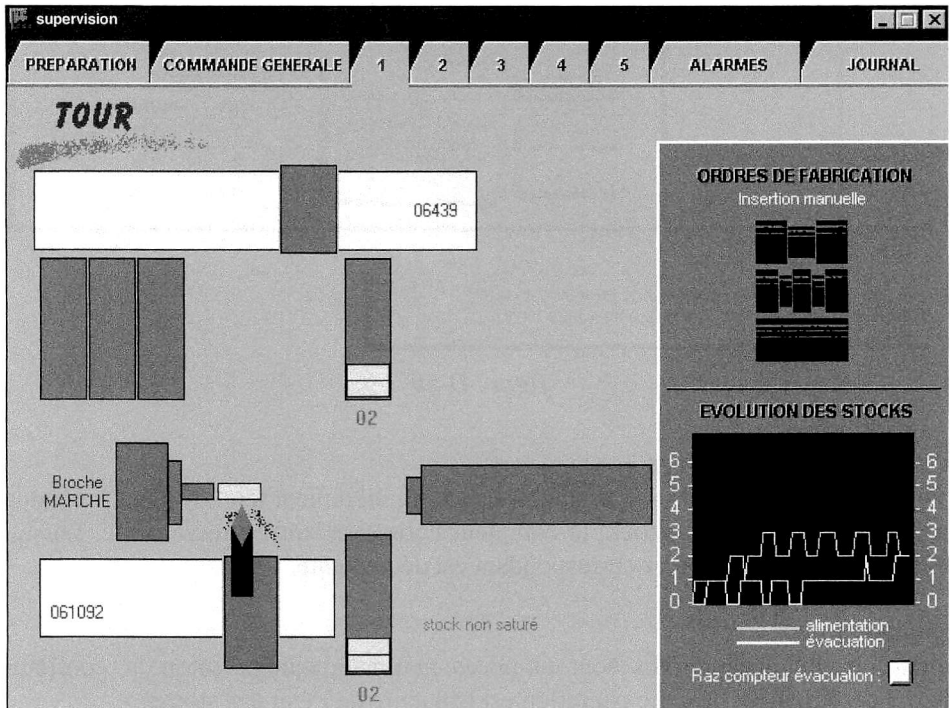


figure D-52

- La surveillance centralisée des défauts

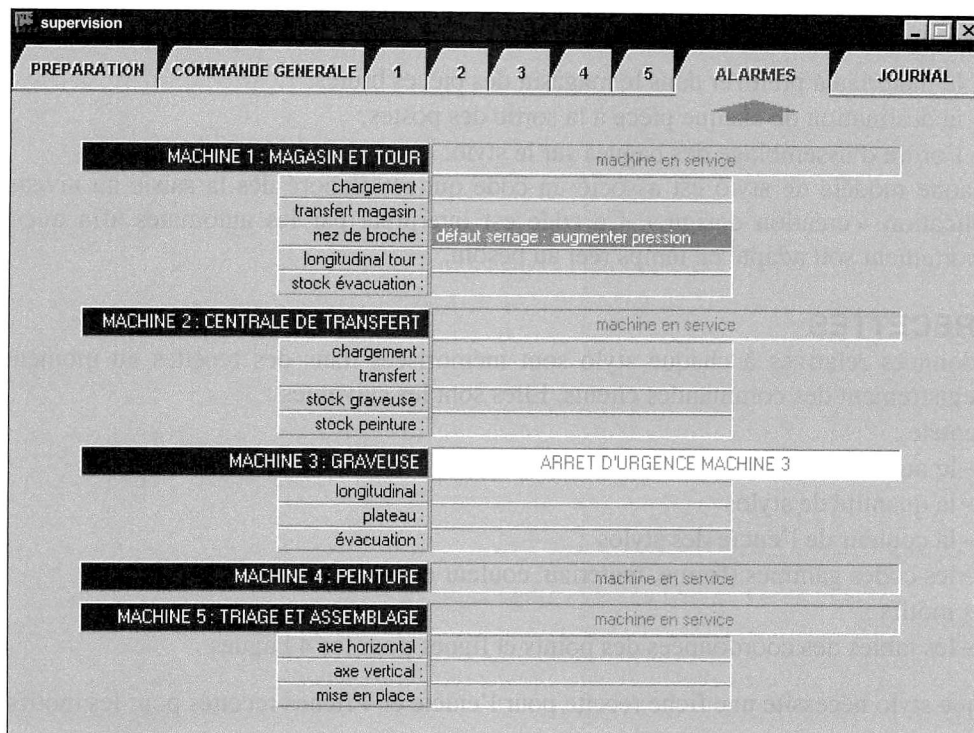


figure D-53

- Le journal des événements et la tenue de statistiques

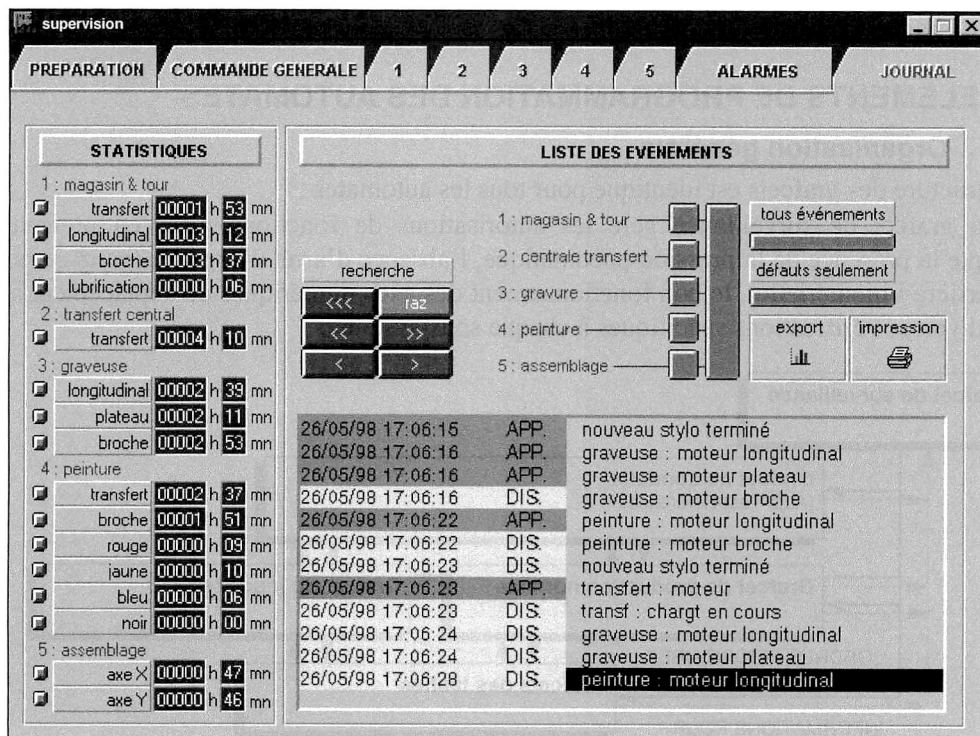


figure D-54

4 - LES DONNEES DE GAMMES

Pendant le fonctionnement, les données de gammes permettent aux automates de déterminer

- le matériau à prélever dans le magasin des pièces brutes,
- la destination de chaque pièce à la sortie des postes,
- l'ordre d'assemblage des bagues sur le stylo.

A chaque modèle de stylo est associé un code qui est élaboré dès la saisie au niveau de l'application « création design ». Ce code est interprété par les automates afin que leur comportement soit adapté en temps réel au besoin.

5 - RECETTES

Les données relatives à chaque stylo sont mémorisées dans des recettes au moment de l'enregistrement des commandes clients. Elles sont les suivantes :

- l'entête :
 - le numéro de client
 - la quantité de stylos
 - la couleur de l'encre des stylos
 - les codes gammes (forme, matériau, couleur et aspect)
- les motifs :
 - les tables des coordonnées des points et lignes pour les 4 bagues

Chaque stylo nécessite une fiche recette pour l'entête et 4 fiches recettes pour les motifs.

Les entêtes sont transmises aux automates en début de campagne alors que les coordonnées des motifs sont transmises au fur et à mesure du besoin sur requête de l'automate du poste de gravure. La fiche recette du motif est sélectionnée en fonction du numéro de stylo et du type de bague à graver.

6 - ELEMENTS DE PROGRAMMATION DES AUTOMATES

6 - 1. Organisation générale

La structure des grafquets est identique pour tous les automates :

- un grafquet de surveillance gère les autorisations de fonctionnement en prenant en compte la présence de la pression pneumatique, l'absence d'arrêt d'urgence (coup de poing et barrière immatérielle), le bon fonctionnement des axes numériques de déplacement ainsi que diverses vérifications spécifiques à chaque sous-système,

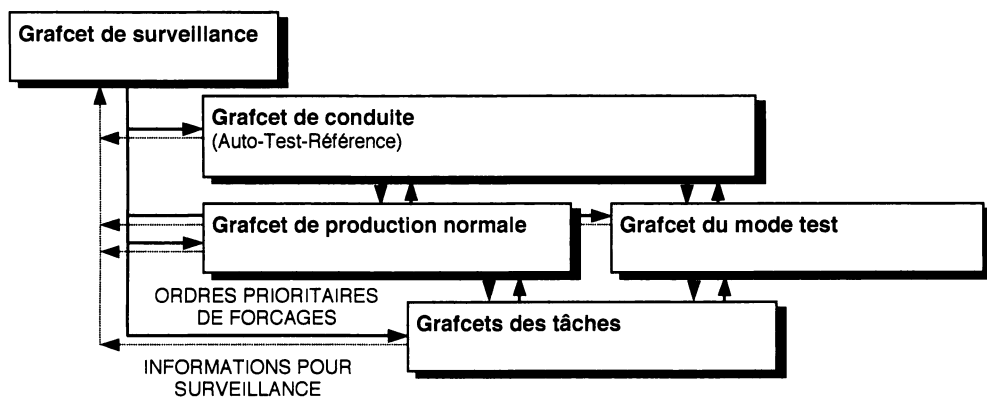


figure D-55

- un grafcet de conduite gère les 3 modes de fonctionnement : automatique, tests unitaires et mise en référence de la partie opérative (prise d'origine des axes numériques notamment),
- les modes de production normale et de test appellent en général des grafquets de tâches.

A titre d'exemple, on représente ci-dessous le grafcet de surveillance au niveau *automate* du sous-système 1 (magasin et tour) :

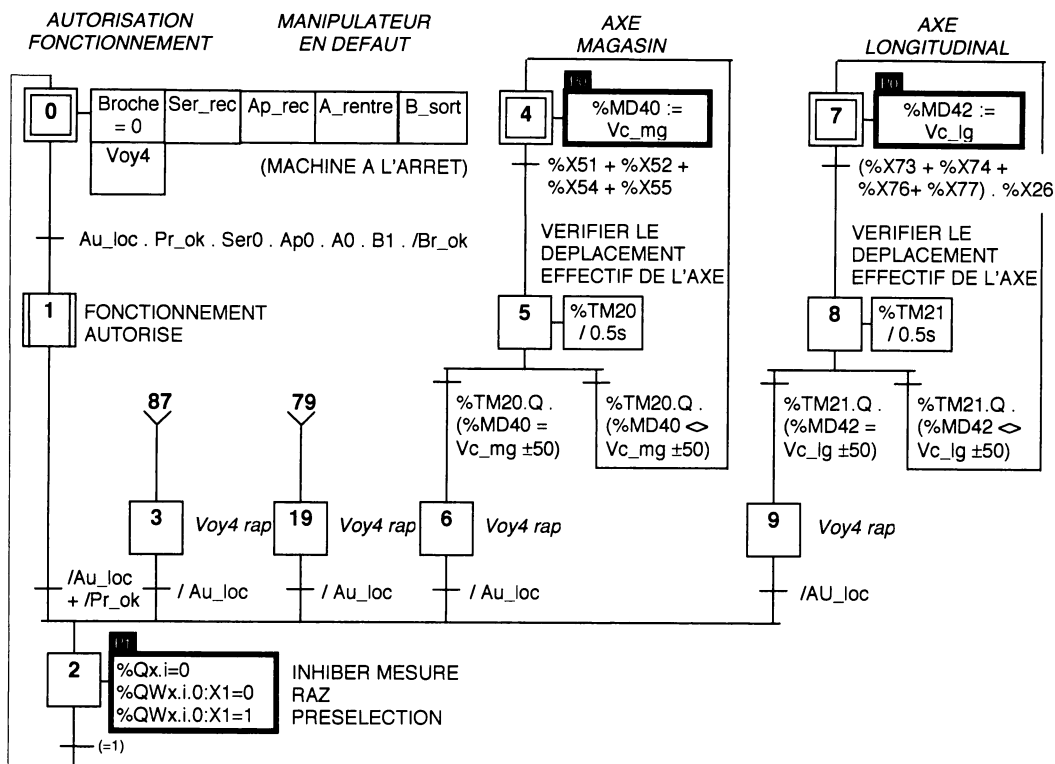


figure D-56

Actions supplémentaires

- sur %X2 : - forcer les grafquets en situation initiale (↵ Prel : %S21=1)
- sur %X19 : - interdire le redémarrage du cycle de transfert magasin
- forcer Lg_p , Lg_n , Plongée , Gorge , Lub et Broche à zéro
- sur %X6 : - interdire le redémarrage des cycles d'usinage et du manipulateur
- forcer Mg_p et Mg_n à zéro
- sur %X9 : - interdire le redémarrage des cycles de transfert magasin et du manipulateur
- forcer Lg_p , Lg_n , Plongée , Gorge , Lub , Séchage et Broche à zéro
- sur %X0 : - émission message d'AU vers superviseur (%M114)

6 - 2. Vue d'ensemble des modes de marche

A titre d'exemple, on indique ci-dessous le diagramme GEMMA pour le sous-système 2 (transfert central) :

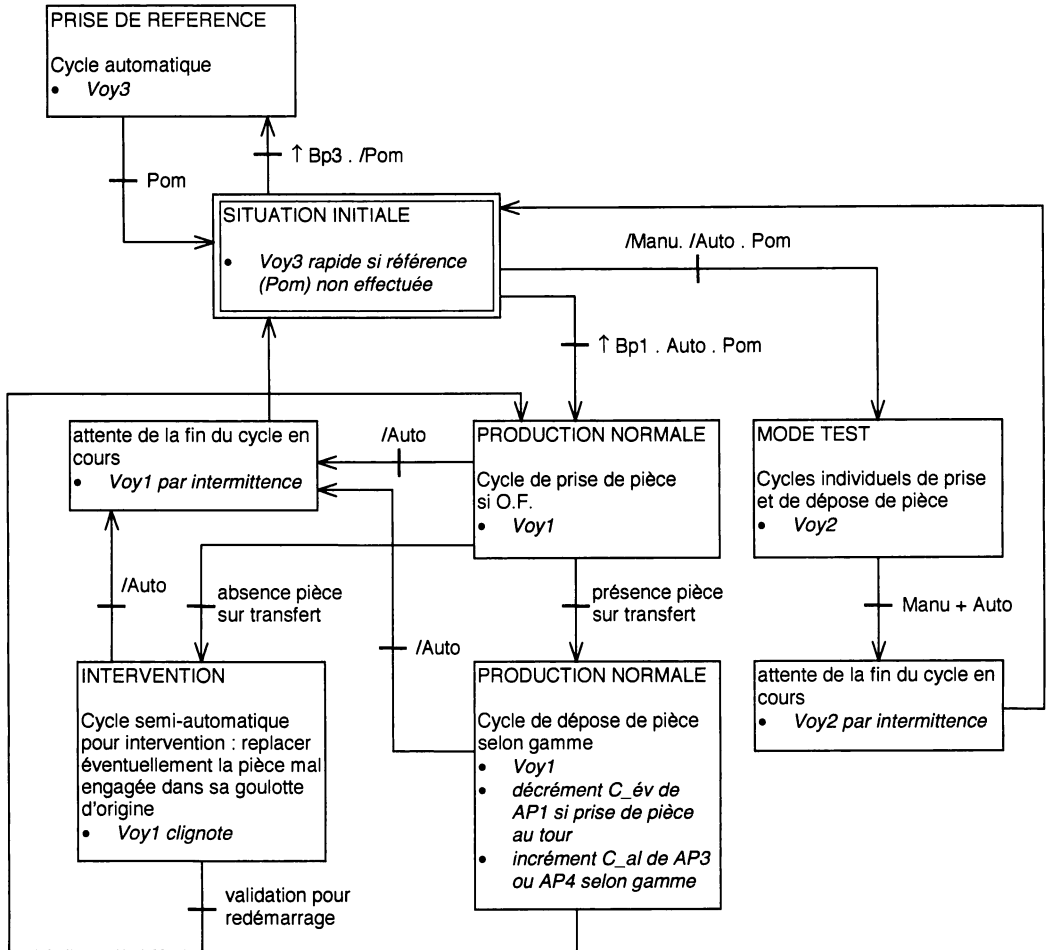


figure D-57

La production normale est décomposée en deux parties : l'une pour la prise de pièce et l'autre pour la dépôt de pièce. Cette organisation permet une gestion des stocks plus sûre.

Le diagramme de la figure D-59 se traduit par le grafcet au niveau automate suivant :

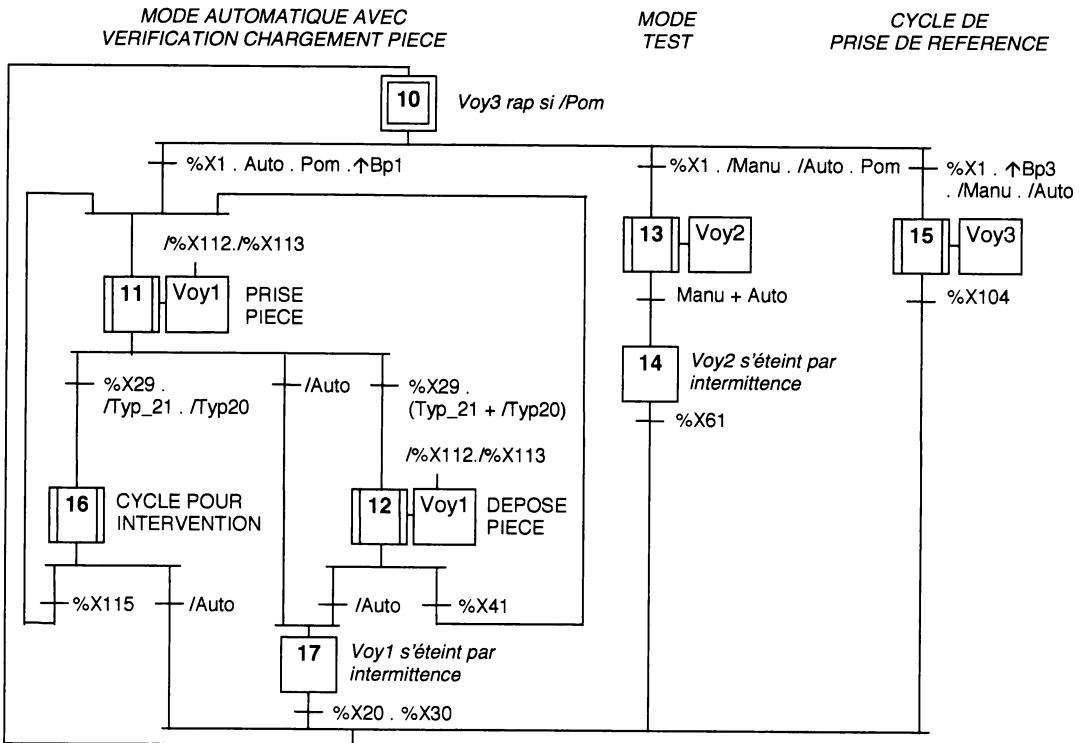


figure D-58

6 - 3. Cycles de prise de référence

Dans le cas du grafcet de conduite ci-dessus, la prise d'origine de l'axe numérique est réalisée par le cycle X100 : l'axe se déplace vers la butée de fin de course jusqu'à immobilisation. Celle-ci est détectée lorsque la coordonnée courante est égale à la coordonnée de la séquence X101-102 précédente. La prise de référence se fait ensuite par passage devant un capteur de référence.

Tous les axes numériques sont gérés de cette même manière.

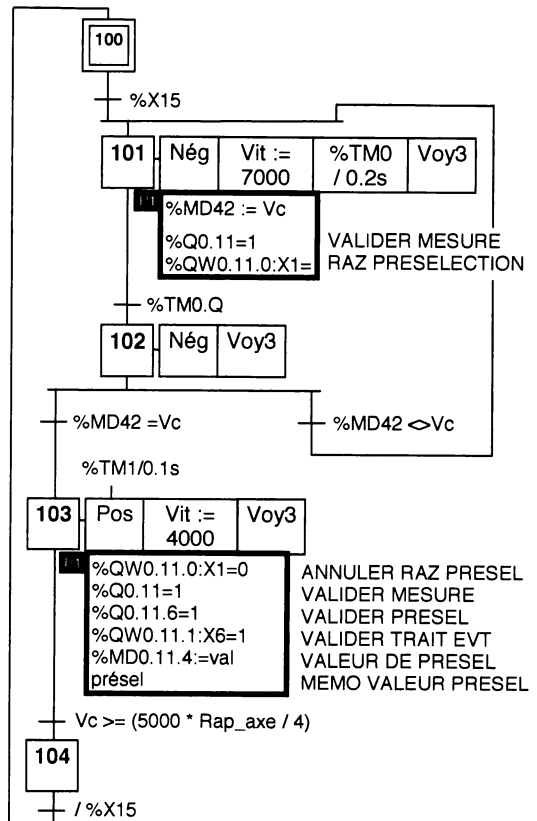


figure D-59

6 - 4. Mode de production normale

Les modes de production normale des différents sous-systèmes sont plus ou moins complexes. Une étude complète de la partie opérative est nécessaire pour pouvoir en interpréter les grafquets.

Le fonctionnement du sous-système 4 (peinture) est décrit à titre d'exemple puisqu'il s'apparente de très près à l'exercice *bacs de trempe* du chapitre IX : en lieu et place des 4 bacs sont installées les buses qui projettent les couleurs sur les bagues (rouge, bleu, jaune et noir). Une cinquième buse sert au séchage de la bague. Par comparaison aux *bacs de trempe*, on constate que la gestion des positions est très nettement simplifiée lorsqu'on utilise un axe de type numérique.

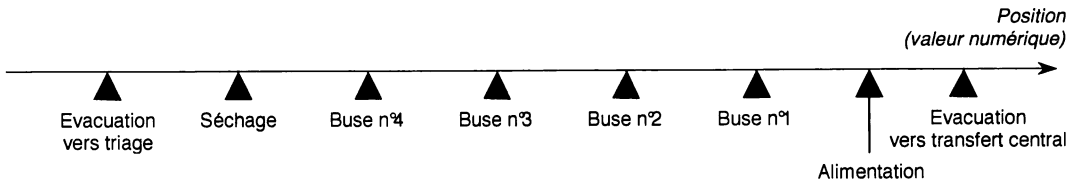


figure D-60

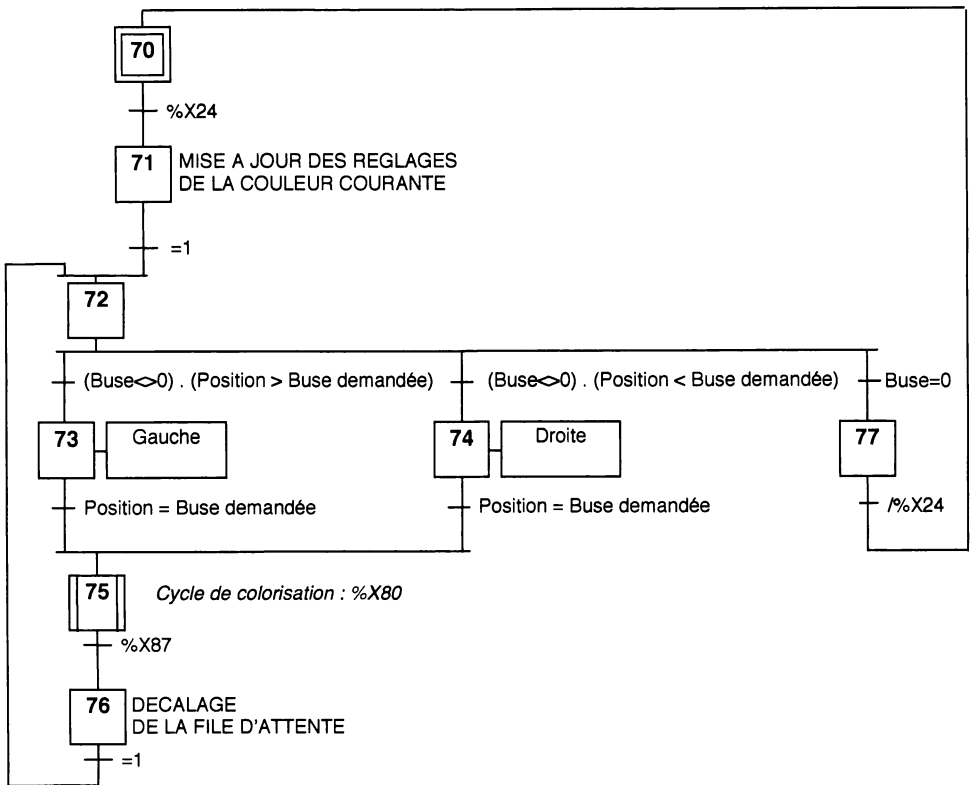


figure D-61

Dans la COLLECTION TECHNOSUP :

Niveau A : Approche

Niveau B : Bases

Niveau C : Compléments

calcul scientifique :

- VBA pour Excel. Bibliothèque mathématique avec applications pratiques 600 p. (C)
- Faire des maths avec *Mathematica*. Initiation, thèmes d'étude 160 p. (B)
- Éléments d'analyse. Calcul différentiel et intégral 240 p. (A)
- Mathématiques des sciences appliquées 216 p. (B)
- Suites et séries. Cours et exercices corrigés 192 p. (B)
- Analyse harmonique. Cours et exercices 192 p. (B)
- Calcul scientifique avec *Matlab* 288 p. (C)
- Modélisation et analyse des systèmes linéaires 224 p. (C)
- Tenseurs, variations et milieux continus 288 p. (C)

D. ROUX
N. VERDIER
B. RADJ, A. EL HAMI
Ph. GOLDNER
A.-J. RIQUET
B. ROSSETTO
J. KOKO
J.F. MASSIEU, Ph. DORLÉANS
J.-F. GANGHOFFER

mesure :

- Mesure physique et instrumentation 192 p. (A)
- Capteurs électrochimiques. Fonctionnement, utilisation, conception 288 p. (C)
- Systèmes d'acquisition de données 240 p. (B)
- Traitement des mesures. Interprétation, modélisation 384 p. (B)
- Traitement statistique du signal 216 p. (C)
- Analyse spectrale Cours Supélec 192 p. (C)

D. BARCHIESI
G. GONDRAN, P. FABRY
E. ETIEN
R. JOURNEAUX
M. BARRET
G. FLEURY

probabilités :

- Probabilités pour modéliser et décider 256 p. (A)
- Calcul des probabilités 224 p. (B)
- Modélisation probabiliste pour l'ingénieur 312 p. (C)

N. SAVY
J.-P. BOULAY
A. SMOLARZ

statistiques :

- Statistique sans mathématique 224 p. (A)
- Assimiler et utiliser les statistiques 288 p. (A)
- Statistiques et expérimentation en biologie 192 p. (A)
- Statistique mathématique 360 p. (B)

J. BADIA, R. BASTIDA, J.R. HAÏT
L. PIBOULEAU
J.-Cl. LABERCHE
J.-P. BOULAY

mécanique quantique ::

- La mécanique quantique et ses applications Cours Supélec 224 p. (C)

A. et M.-F. CHARLIER

ondes, corpuscules :

- Ondes et matière 352 p. (C)
- Physique pour l'électronique. Corpuscule, onde, état quantique, structures 320 p. ((B))
- Éléments de propagation électromagnétique 160 p. (B)
- Techniques micro-ondes. Dispositifs passifs et tubes micro-ondes Cours Supélec 320 p. (C)
- Optoélectronique. Composants photoniques et fibres optiques Cours Supélec 320 p. (C)
- Le radar. Théorie et pratique Cours Supélec 160 p. (C)

D. BARCHIESI, M. LAMY DE LA CHAPELLE
A. et D. DEVILLE
Ph. ROSNET
M. HÉLIER
Z. TOFFANO
J.-M. COLIN

optique :

- Optique physique. Interférences, diffraction, holographie. Cours et exercices 192 p. (A)
- Optique moderne. Polarisation, lasers, fibres optiques. Cours et exercices 224 p. (B)
- Exercices corrigés d'optique. Optique instrumentale. Optique de Fourier 192 p. (B)

FI. WEIL
FI. WEIL
J. SURREL

acoustique :

- Biophysique de l'environnement sonore 192 p. (B)
- Acoustique générale 352 p. (C)

C. GELIS
C. POTEI, M. BRUNEAU

géophysique :

- Murmures ionosphériques. Techniques de réception sous le seuil de 100 kHz 216 p. (C)

J.-J. DELCOURT

géologie :

- Pétrologie sédimentaire 264 p. (B)
- Géologie de terrain 160 p. (A)

F. BOULVAIN
F. BOULVAIN, J. VANDER AUWERA

environnement :

- Les traitements de l'eau. Cours et problèmes résolus 256 p. (B)
- Techniques appliquées au traitement de l'eau 256 p. (B)
- Pollution atmosphérique. Causes, conséquences, solutions, perspectives 224 p. (B)
- La lutte biologique. Application aux arthropodes et adventices. 320 p. (B)
- Biodégradation des matériaux. (288 p. (C))
- Gestion des déchets. Réglementation, organisation, mise en œuvre 224 p. (A)
- Traitement des déchets. Valorisation, élimination. 288 p. (A)

Cl. CARDOT
(coord.) Cl. CARDOT
P. MASCLET
B. PINTUREAU
A. CORNET, F. FEEUGEAS, B. TRIBOULET
Th. ROGAUME
A. ADDOU

chimie :

- Révisions et autoévaluation en chimie structurale. 240 p. (A)
- Comprendre la chimie organique - 2e édition 240 p. (A)
- Assimiler la chimie organique 192 p. (A)
- La chimie en IUT et BTS. Cours et exercices résolus 224 p. (A)
- Chimie des solutions. Résumés de cours et exercices corrigés 224 p. (B)

C. WATERLOT
A. LASSALLE, D. ROBERT
A. LASSALLE, D. ROBERT
F. VAISSIAUX
P.-L. FABRE

génie chimique :

- Thermodynamique et cinétique chimique. Résumés de cours et exercices 224 p. (B)
- Cinétique et catalyse hétérogènes 320 p. (C)
- Réactions et réacteurs chimiques 288 p. (B)
- Génie chimique. Les opérations unitaires 312 p. (C)

P.-L. FABRE
B. GILOT, R. GUIRAUD
M. GUISNET, S. LAFORGÉ, D. COUTON
D. MORVAN

analyse physico-chimique :

- Les techniques de laboratoire 160 p. (A)
- Spectrométrie de masse 320 p. (C)
- Séparation et analyse des biomolécules. Méthodes physico-chimiques 256 p. (B)

E. BOURGUET, C. AUGÉ
G. DUGUAY
J.-P. SINE

mécanique :

- Mécanique générale. Cours, exercices et problèmes corrigés 288 p. (B) CI. CHÈZE, H. LANGE
- Actions mécaniques. Statique. Inertie. De la théorie aux applications 224 p. (A) CI. CHÈZE, F. BRONSARD
- Vibrations des structures 224 p. (B) G. VENIZELLOS

milieux continus :

- Mécanique des milieux déformables 288 p. (B) M. FOURAR, CI. CHÈZE
- Théorie microscopique des liquides 480 p. (C) J.-L. BRETONNET
- Les écoulements de fluides newtoniens 384 p. (C) J.-N. GENCE
- Écoulements et transferts 312 p. (C) R. BRUN, N. BELOUAGGADIA

génie mécanique

- Détermination des éléments de machines 360 p. A. BOURDON, L. MANIN, D. PLAY
- Conception et construction des moteurs alternatifs 288 p. (C) Ph. ARQUÈS
- Transmissions mécaniques de puissance. Boîtes de vitesses automatiques 288 p. (C) Ph. ARQUÈS
- Ingénierie des turbomachines. Cours et exercices résolus 288 p. (C) M. PLUVIOSE

structures :

- Dimensionnement des structures. Résistance des matériaux 224 p. (B) D. CHÈZE
- Mécanique des structures. Du calcul analytique au calcul matriciel 288 p. (B) J.-Ch. CRAVEUR, CI. CHÈZE
- Vibrations des structures pour l'ingénieur et le technicien 264 p. (C) B. COMBES
- Comprendre les éléments finis. Structures 288 p. (C) A. CHATEAUNEUF

thermodynamique :

- La thermodynamique des principes aux applications 312 p. (C) CI. CHEZE, P. BAUER
- Réactions thermiques en phase gazeuse. Modélisation 288 p. (C) G.-M. CÔME
- La méthode modale en thermique 320 p. (C) G. LEFEBVRE
- Transferts thermiques, application à l'habitat. Méthode nodale 224 p. (C) H. CORTES, J. BLÔT
- Combustion. Inflammation, combustion, pollution, applications 320 p. (C) Ph. ARQUÈS

énergétique :

- Moteurs alternatifs à combustion interne. De la théorie à la compétition 288 p. (B) Ph. ARQUÈS
- Machines à fluides. Principes et fonctionnement 288 p. (C) M. PLUVIOSE
- Conversion d'énergie par turbomachines 288 p. (C) M. PLUVIOSE
- Propulseurs aéronautiques et spatiaux 288 p. (C) P. BAUER
- Piles à combustible. Principes, modélisation, applications 192 p. (B) B. BLUNIER, A. MIRAQOUI
- Énergie solaire. Calculs et optimisation 320 p. (B) J. BERNARD
- Énergie nucléaire 1. De la théorie aux applications 256 p. (B) J. BERNARD
- Énergie nucléaire 2. Les réacteurs nucléaires électrogènes 288 p. (B) J. BERNARD

science des matériaux :

- Introduction à la cristallographie. Solides cristallisés et empilements compacts 160 p. (A) D. RIOU
- Propriétés et comportements des matériaux - 2e édition 336 p. (B) A. CORNET, F. HLAWKA
- Métallurgie mécanique - 2e édition 320 p. (B) A. CORNET, F. HLAWKA
- Fatigue des structures. Endurance, rupture, critères, contrôle, durabilité 320 p. (C) G. HENAFF, F. MOREL
- Endommagement interfacial des métaux 256 p. (C) G. SAINDRENAN, R. LE GALL, F. CHRISTIEN
- Cycle de vie des surfaces industrielles (312 p. (C) F. HLAWKA, A. CORNET

génie civil :

- Béton armé. Application de l'eurocode 2 224 p. (B) R. NICOT
- Analyse et dimensionnement sismiques 224 p. (B) P. LESTUZZI

productique :

- Méthodes, productique et qualité 224 p. (B) J.-M. CHATELET
- Maintenance industrielle 288p. (B) J.M. AUBERVILLE
- Analyse et maintenance des automatismes industriels 192 p. (B) A. REILLER
- TGAO La technologie de groupe 288 p. (C) A. NADIF
- Réseaux de Petri 216 p. (B) M. BOURCERIE

génie industriel :

- Maîtriser la conduite de projet 192 p. (A) C. ALONSO
- Management de projet technique 192 p. (B) C. CAZAUBON, G. GRAMACIA, G. MASSARD
- Organisation et génie de production 224 p. (B) F. LAMBERSEND
- Méthode d'aide à la décision. Approche théorique et études de cas 192 p. (B) R. LABBÉ
- Exercices de mathématiques. BTS industriels 288 p. (A) J. BLANC, R. ROY

électricité générale :

- Vade-mecum d'électrotechnique 312 p. (A) C. LE TRIONNAIRE, J.-P. PICHENY
- Circuits électriques, Régimes continu, sinusoïdal et impulsionnel 192 p. (A) J.-P. BANCAREL
- Les lois de l'électricité 288 p. (A) M. PIOUS
- Annales d'électrotechnique BTS Maintenance 256 p. (A) D. VINCENT, N. ORTEGA
- Annales d'électrotechnique BTS Maintenance 1997/2008 224 p. (A) D. VINCENT
- T.P. d'électrotechnique par simulation avec PSIMDEMO 224 p. (A) F. LEPLUS

machines électriques :

- Moteurs à courant alternatif 288 p. (A) D. JACOB
- Le moteur asynchrone. Régimes statique et dynamique 160 p. (C) L. MUTREL
- Machines à courant alternatif 240 p. (B) D. NAMANE
- Modélisation et commande des moteurs triphasés 256 p. (C) G. STURTZER, E. SMIGIEL
- Électrotechnique. Machines et réseaux Cours Supélec 256 p. (C) J.-P. FANTON
- Machines électriques. Théorie et Mise en oeuvre Cours Supélec 256 p. (C) Ph. BARRET
- Machines électriques tournantes 384 p. (C) B. LAPORTE

électronique de puissance :

- Électronique de puissance, Principes, fonctionnement, dimensionnement 256 p. (A) D. JACOB
- Les redresseurs, Redresseurs à diodes, à thyristors et mixtes 336 p. (B) J. MIGNARD, C. PIN

électronique :

- Les fondamentaux en électronique 224 p. (A) P. ROCHETTE
- Des clés pour l'électronique. Travaux dirigés illustrés par simulation 160 p. (A) B. GIRAULT
- Les oscillateurs en électronique. Cours et exercices corrigés 160 p. (B) G. COUTURIER
- L'outil graphique en électronique et automatique 224 p. (B) J. BAILLOU, G. CHAUVAT, C. PEJOT
- Modulation d'amplitude, Cours et exercices 352 p. (C) F. BIQUARD
- Amplificateurs fondamentaux et opérationnels. Cours et exercices corrigés 352 p. (B) A. LANTZ
- Electronique radiofréquence Cours Supélec 256 p. (C) A. PACAUD
- Electronique analogique rapide 216 p. (C) A. FABRE
- Circuits spécialisés de l'électronique actuelle 320 p. (C) A. et D. DEVILLE

semi-conducteurs :

- Composants à semiconducteurs Cours Supélec 256 p. (C) O. BONNAUD, A. PACAUD
- Technologie mirco-électronique. Du silicium aux circuits intégrés 160 p. (B) O. BONNAUD
- Détecteurs à semi-conducteurs, Principes et matériaux 224 p. (C) J.-P. PONPON
- Les semiconducteurs de puissance Cours Supélec 256 p. (C) P. ALOÏSI

électronique numérique :

- Électronique numérique. Fonctionnement, modélisation, circuits intégrés 320 p. (B) A. et D. DEVILLE
- Circuits intégrés numériques. Du transistor au microprocesseur 224 p. (A) A.-Riadh BABA-ALI

traitement du signal :

- Signaux et systèmes continus et échantillonnés 192 p. (B) M. VILLAIN
- Signaux et systèmes linéaires Cours Supélec 192 p. (B) A. PACAUD
- Traitement du signal analogique. Cours 224 p. (A) T. NEFFATI
- Traitement du signal analogique. Exercices et problèmes résolus 224 p. (B) T. NEFFATI
- Ingénierie du signal. Théorie et pratique 224 p. (B) Ph. COURMONTAGNE
- Analyse et traitement du signal. Approches pour l'ingénieur 320 p. (B) Ph. GAILLARD, R. LENGELLE
- Théorie et pratique du signal déterministe et aléatoire, continu et discret 384 p. (B) J.-P. TANGUY

filtrage numérique :

- Débuter en traitement numérique du signal 224 p. (A) J.-N. MARTIN
- Analyse et contrôle numériques du signal 192 p. (B) Ph. DESTUYNDER, F. SANTI
- Traitement numérique du signal. Théorie et applications 256 p. (C) K. KPALMA, V. HAESE-COAT

automatique :

- Ce qu'il faut savoir sur les automatismes, Fiches-résumés 256 p. (A) P. GRARE, I. KACEM
- Systèmes asservis linéaires 224 p. (B) M. VILLAIN
- Asservissements linéaires continus 288 p. (B) P. ROUSSEAU
- Commande analogique et numérique des systèmes 384 p. (B) R. KONN
- Systèmes et asservissements continus 320 p. (C) É. OSTERTAG
- Régulation PID en génie électrique. Étude de cas 256 p. (C) D. JACOB
- Problèmes résolus d'automatique 288 p. (B) Ch. BURGAT
- Problèmes résolus d'automatique par thèmes et par types d'application 256 p. (B) R. HUSSON
- Développement des grafocets - 2^e édition 192 p. (B) B. REEB

robotique :

- Traité de robotique. 1- Les architectures, conception, modélisation, équations 432 p. (C) C. BOP
- Traité de robotique. 2- Les parties opératives 336 p. (C) C. BOP
- Commande numérique des systèmes Cours Supélec 256 p. (C) E. GODOY, É. OSTERTAG
- Commande et estimation multivariables 288 p. (C) É. OSTERTAG
- Commande automatique des systèmes linéaires, utilisation de MATLAB 256 p. (C) V. MINZU, B. LANG
- Commande et diagnostic des systèmes dynamiques 320 p. (C) R. TOSCANO
- Ingénierie de la commande des systèmes 256 p. (C) A. CROSNIER, G. ABBA, B. JOUVENCEL, R. ZAPATA

informatique industrielle :

- Bit après bit. Computers 1 320 p. (B) J.-J. MERCIER
- Séquence après séquence. Computers 2 288 p. (B) J.-J. MERCIER
- Instruction après instruction. Computers 3 320 p. (C) J.-J. MERCIER
- Circuits logiques programmables. Mémoires, PLD, CPLD, FPGA 256 p. (B) A. NKETSA
- Du binaire au processeur 320 p. (BC) E. MESNARD
- Concevoir son microprocesseur 256 p. (B) J.-Ch. BUISSON
- Logique combinatoire et séquentielle. Système, méthodes, réalisations 320 p. (C) Cl. BRIE
- Architecture des systèmes sur puce 320 p. (C) A. ATTOUI

image :

- Traitement de l'image et de la vidéo, avec exercices pratiques en Matlab et C++ 240 p. (C) R. BELAROSSI

communication, réseaux :

- Lignes de transmission 224 p. (B) R. MEYS
- Transmission de l'information 192 p. (B) Ph. FRAISSE, D. MARTY-DESSUS, R. PROTIÈRE
- Architectures des réseaux et télécommunications 192 p. (B) P. LORENZ
- Réseaux Intranet et Internet, Architecture et mise en œuvre 336 p. (B) J. PHILIP
- Codes correcteurs. Principes et exemples 192 p. (C) J. BADRIKIAN

bases de données :

- Bases de données. Implémentation avec Access 256 p. (B) J. AUBERT
- Conception méthodique des bases de données. Un guide de bonne pratique 224 p. (A) G. BUENO
- Gradualité et imprécision dans les bases de données 320 p. (B) P. BOSCH, L. LIÉTARD, O. PIVERT, D. ROCACHER

bureautique :

- Excel pour l'ingénieur - 2e édition 336 p. (B) Ph. BELLAN

processus temps réel :

- Approche du temps réel industriel 160 p. (A) J.-M. DE GEETER
- Gestion des processus industriels temps réel 224 p. (B) J.-J. MONTOIS

programmation :

- Algorithmes fondamentaux et langage C 320 p. (B) J.-L. IMBERT

- Le langage C par l'exemple 320 p. (A) Ph. ROBINET
 - Du procédural à l'objet: les langages C et C++ 352 p. (B) J. PHILIPP
 - Belle programmation et langage C Cours Supélec 192 p. (C) Y. NOYELLE
 - Programmation avec le langage Python 336 p. (C) X. DUPRE
 - Compilations des langages de programmation 192 p. (C) M. GAUTIER
- génie logiciel :**
- Méthode orientée objet intégrale MACAO 320 p. (B) J.-B. CRAMPES
 - La conception orientée objet, évidence ou fatalité 160 p. (B) J.-L. CAVARERO, R. LECAT
 - Conception des systèmes d'information. Méthodes et techniques 320 p. (B) P. ANDRÉ, A. VAILLY
 - Spécification des logiciels. Deux exemples : Z et UML 320 p. (C) P. ANDRÉ, A. VAILLY
 - Exercices corrigés de conception logicielle. Modélisation par la pratique 320 p. (B) P. ANDRÉ, A. VAILLY
 - Exercices corrigés d'UML. Passeport pour une maîtrise de la notation 320 p. (C) P. ANDRÉ, A. VAILLY
 - Exercices corrigés en langage Z. Spécifications formelles par l'exemple 256 p. (C) P. ANDRÉ, A. VAILLY
- ergonomie :**
- Interfaces graphiques ergonomiques. Conception, modélisation 192 p. (B) J.-B. CRAMPES
 - Analyse des tâches en ergonomie 160 p. (A) M. MOSCATO
- logistique :**
- Logistique interne 160 p. (A) L. AMODEO, F. YALAOUI
- sécurité :**
- Sécurité des ouvrages. Risques. Géotechnique 320 p. (C) J.-L. FAVRE
 - Risques et sécurité 224 p. (C) J.-F. GUYONNET
- économie :**
- Gestion financière. Analyse et politique financières de l'entreprise 256 p. (B) A. RIVET
 - Méthodes mathématiques pour les finances 384 p. (C) J.-Ph. ARGAUD, O. DUBOIS
 - Les marchés à terme agricoles 256 p. (B) N. HABERT
 - La Bourse et les produits boursiers 320 p. (B) D. ARNOULD
- législation :**
- La réglementation du travail 160 p. (A) Ph. MALINGREY
 - Le travail salarié 256 p. (A) P. IRIART
 - Connaître et comprendre le droit. Principes et cas pratiques 256 p. (B) C. GABET
- éthique :**
- Science, technologie et éthique 288 p. (B) S. LAVELLE

Cet ouvrage a été imprimé
en février 2011 par

CPI

FIRMIN-DIDOT

27650 Mesnil-sur-l'Estrée
N° d'impression : 103880
Dépôt légal : février 2011

Imprimé en France