METHODOLOGIE DE SPECIFICATION DU MODELE GENERIQUE DE SYNCHRONISATION POUR LA CONDUITE D'UN SYSTEME FLEXIBLE DE PRODUCTION

L. ZERGUINI [a] & P. MARTINEAU[b]

[a] Institut de Mathématiques, U.S.T.H.B., BP. 32, El Alia, 16111 Alger, Algérie.
E - mail: Zerguini @ Yahoo.com

[b] E3I/ LI, 64, Av Jean Portalis, 37200 Tours, France.

RESUME: Cet article vise à définir une méthodologie de spécification dédiée à la commande temps réel des systèmes flexibles de production couvrant le niveau de coordination qui s'occupe de gérer les interactions entre les différentes ressources de l'atelier, c'est-à-dire principalement des problèmes de synchronisation.

MOTS-CLES: Systèmes Flexibles de Production, Modèle générique de synchronisation, spécification fonctionnelle, réseau de Petri à objets.

1. INTRODUCTION

Les systèmes à événements discrets (SED) sont un type particulier de système dynamique dont l'évolution est dirigée par l'occurrence asynchrone de différents événements *Ho(1989)*. Les systèmes automatisés de production y compris les systèmes flexibles de production (FMS) sont des exemples de SED. En général, un FMS est conçu pour fabriquer un ensemble de produits. La fabrication d'un produit nécessite une suite de transformations élémentaires appelées opérations. Chaque opération peut être exécutée par une ou plusieurs machines. L'exécution d'une opération nécessite éventuellement la présence d'un ou plusieurs outils. les déplacements d'un produit au sein du système s'effectuent grâce à des moyens de transports. Des systèmes de stockage sont nécessaires pour réguler la production.

L'application du modèle d'une unité de stockage à la chaîne de production en réduction *Proust(1989)* consiste à apparenter toute entité opérationnelle de la chaîne à un moyen de stockage *Martineau(1996)*. Chaque entité, qui est assimilée à un moyen de stockage durant le traitement qu'elle effectue sur la pièce, doit donc coopérer avec les entités précédentes et les entités suivantes. Cette coopération, qui apparaît naturelle dans un atelier de type masse où les pièces passent dans un ordre fixé (la production étant linéaire, elle a eu lieu sans blocages *Grûnenberger(1994))*, entraîne des difficultés de mise en oeuvre dans une structure de production de type atelier flexible.

Lorsqu'un ensemble de produits est dans une situation telle que chacun détient au moins une ressource et est en attente, pour poursuivre sa progression, d'une ressource détenue par l'un des autres, il se produit un interblocage. Une telle situation présente la caractéristique suivante, il existe une chaîne circulaire de produits en attente telle que chaque produit détient les ressources requises par le produit suivant de la chaîne. Dans un tel système à ressources partagées, les interblocages constituent l'un des problèmes majeurs à poser lors de la conception préliminaire *D' Souza(1994)*. Dans un FMS incorrectement conçu , le seul remède aux interblocages peut être le dégagement manuel des buffers ou des machines, puis le redémarrage du système à partir de conditions initiales qui peuvent corriger ces interblocages sous des conditions de production normales. Ainsi, le réamorçage du système, va engendrer une baisse de la production et des coûts de main-d'oeuvre. Il apparaît donc nécessaire de maîtriser parfaitement toutes les phases de la conception préliminaire, c'est-à-dire la spécification fonctionnelle, la modélisation et l'évaluation du comportement du système.
Les réseaux de Petri (RdP_S) *Proth(1995)*, se sont révélés être un langage précis pour les échanges avec les concepteurs. En outre, leurs propriétés ont permis de mettre en évidence certains types de

comportement généraux. Nous allons montrer que les réseaux de Petri à Objet *Sibertin-Blanc(1988)*, peuvent servir à modéliser des systèmes de production relativement complexes.

La deuxième section de cet article présente brièvement les réseaux de Petri à Objets (RdPO). La troisième section s'intéresse à la présentation du modèle générique de synchronisation (MGS), et donne le RdPO correspondant. La simulation se fait ensuite en faisant évoluer le marquage. Cette évolution du marquage étant parfaitement définie par les règles que nous allons présenter, elle est automatisable. Un programme a été écrit pour réaliser cette simulation.

2. LES RESEAUX DE PETRI A OBJETS

Les ($RdPO_S$) sont une abréviation des RdP_S . Un RdPO peut être utilisé dans un environnement orienté-objet caractérisé par un ensemble d'objets O (machines, opérateurs, Gammes d'usinage, ...). Les éléments de O sont des instances de classes d'objets (classification des objets). Chacun de ces objets possède un certain nombre d'attributs comme:(le nom de l'instance, la date de livraison pour une pièce, l'opération à faire pour une pièce, la liste des opérations pouvant être faites pour une machine.....).
Dans un RdPO, les jetons sont vus comme des n-uplets d'instances de classes d'objets. Les conditions et les actions associées aux transitions correspondent aux méthodes, elles travaillent sur les attributs des objets substitués aux variables formelles lors du franchissement des transitions.

Définition des réseaux de Petri à Objets *(Sibertin-Blanc (1988))*

Un réseau de Petri à Objets est défini par le 9-uplet:

$$N_0 = < P, T, C_{lass}, V, Pre, Post, A_{tc}, A_{ta}, M_0 >$$

P et T sont des ensembles finis de places et de transitions respectivement

C_{lass} est un ensemble fini de classes d'objets, éventuellement organisé en une hiérarchie et définissant pour chaque classe un ensemble d'attributs,

V est un ensemble de variables typées par C_{lass},

Pre est la fonction place précédente qui à chaque arc d'entrée d'une transition fait correspondre une somme formelle de n-uplets d'éléments de V,

Post est la fonction place suivante qui à chaque arc de sortie d'une transition fait correspondre une somme formelle de n-uplets d'éléments de V,

A_{tc} est une application qui à chaque transition associe une condition de franchissement faisant intervenir les variables formelles associées aux arcs d'entrée et les attributs des classes correspondantes,

A_{ta} est une application qui à chaque transition associe une action faisant intervenir les variables formelles associées aux arcs d'entrée et les attributs des classes correspondantes.
Une action est constituée d'un ensemble d'affectations de la forme: var . att_i Expr, avec

var, une variable de sortie de la transition,

att_i , un attribut défini par la classe typant la variable var,

Expr, une expression du même type que l'attribut att_i et dont les paramètres sont les attributs des variables d'entrées.

M_0 est le marquage initial qui associe à chaque place une somme formelle de n-uplets d'instances d'objets (les objets doivent être représentés par des identificateurs, leur nom par exemple).

Analyse

Les réseaux de Petri à Objets sont définis sous la forme de réseaux de Petri ordinaires (les réseaux sous-jacents) munis d'inscriptions. L'analyse portera d'abord sur ces réseaux .

3. LE MODELE GENERIQUE

Dans cet article, on présente d'abord brièvement la structure du modèle générique. Une description plus détaillée peut être trouvée dans ***Martineau (1996)***. Par la suite, on exposera la méthodologie de spécification du modèle.

3.1.Présentation

Dans un FMS, on définit une place comme une place de stockage physique unitaire au sein de l'atelier (exemples: une des cases d'un entrepôt, le mandrin d'une fraiseuse, la pince d'un robot...). Par extension, on définit un stock comme étant un ensemble de places partageant le même ensemble d'actionneurs (des exemples d'actionneurs : un palettiseur, un moteur de tapis roulant ou de robot. Des exemples de stock : les places sur un tapis roulant, les places d'un entrepôt de stockage, la place d'une pince de robot). L'entité opérative correspond à l'ensemble des places muni de son ou de ses actionneurs (par exemple, les places d'un tapis roulant constituent un stock et, ces même places munies du moteur du tapis, forment l'entité opérative). L'état du stock de l'entité opérative peut donc être représenté par le septuplet [C_{max}(le nombre maximum de places), N_0(le nombre de places occupées), L_0(la liste des places occupées), L_L(la liste des places libres), N_i(le nombre de places indisponibles dans le cas où l'entité est à capacité variable), L_i (la liste des places indisponibles dans le cas où l'entité est capacité variable)].Une entité opérationnelle est constituée de l'entité opérative elle-même, de l'automate chargé de la contrôler et d'un PC sur lequel est installé le programme de commande. Les entités opérationnelles peuvent être soit mobiles (robot, AGV, Convoyeur) ou non mobiles (Tour à commandes Numériques, fraiseuse, raboteuse, entrepôt automatisé, station de contrôle,...).

Elles peuvent être modélisées indépendamment avec seulement sept primitives:

-Vérification de déstockage : Primitive chargée de contrôler que le déstockage est possible et de déterminer la pièce à déstocker.

-Préparation de déstockage: primitive permettant de préparer l'entité pour réaliser le déstockage (exemples : dans le cas de la synchronisation de deux robots, il s'agit, de faire pénétrer le bras du premier robot dans le champ d'action du second. Dans le cas d'un convoyeur non accumulatif, on doit faire avancer le tapis si la pièce se situe hors de la zone d'atteinte du manipulateur qui va la saisir)

-déstockage : Primitive réalisant l'opération de déstockage proprement dite (i.e. délivrer une pièce à l'entité suivante).

-Vérification stockage : Primitive chargée de contrôler que le stockage est possible et de déterminer l'endroit où stocker la pièce.

- *Préparation stockage: Primitive chargée de préparer le stockage.*

On notera que les primitives Préparation déstockage et Préparation stockage ne concernent que certaines liaisons. En effet, seules les entités mobiles peuvent effectuer ce type de traitement.

- Stockage : *primitive chargée d'effectuer le stockage effectif de la pièce sur l'entité (i.e recevoir une pièce de l'entité précédente).*

-Traitement : Primitive représentant la valeur ajoutée que doit fournir l'entité à un produit faisant partie de son domaine. Un traitement peut par exemple être une opération d'usinage pour une machine outil, le déplacement vers la prochaine station pour l'AGV, le trie des produits emmagasinés pour un entrepôt automatisé.....) .

La description de ces entités est plus détaillée en termes de Pré-Conditions / Actions / Post-Actions *Richard(1988).*

La Pré-Condition réunit l'ensemble des conditions d'activation de la primitive de base. Elle correspond à la condition nécessaire et suffisante pour lancer la primitive de base. L'Action réunit les actions élémentaires que la primitive réalise. La Post-Action est l'émission de signaux que la primitive envoie pour garantir l'enchaînement des commandes et la cohérence des informations.

Dans un FMS, la synchronisation s'effectue entre deux entités, puisqu'il s'agit de déstocker de l'entité amont et stocker sur l'entité aval et vice et versa. Par exemple, si un produit sur l'entité A doit être déstocké vers l'entité B (Fig.1), le superviseur du FMS émet une demande de déstockage vers l'entité A. Dès que le signal est reçu et si toutes les autres conditions sont remplies, l'entité A entre dans la primitive "Vérification de Déstockage ". La synchronisation entre A et B se poursuit. Quand le

produit est stocké dans l'entité B, la Post-Action de la primitive Stockage envoie au superviseur le message " Fin Stockage".

3.2 SPECIFICATION DU MODELE GENERIQUE

Chaque entité (machine, robot, convoyeur etc..) du modèle est représentée par une classe d'objets contenant les informations suivantes:

Entité = { NomE: identificateur
Opération: liste des opérations possibles
Type: mobile ou immobile
Nbo: nombre de places occupées
NbI: nombre de places libres
Données: informations concernant l'état interne de l'entité }

Chaque pièce est représentée par une classe d'objets, contenant les informations suivantes :

Pièce = { Nom P: identificateur
Gamme: liste des machine et des opérations pour l'usinage de la pièce }

Puisque la synchronisation s'effectue deux à deux, on doit spécifier le type de la liaison (mobile-immobile, immobile - mobile ou mobile - mobile), à cet effet on introduit une classe le représentant contenant les informations suivantes:

Typeliaison = {Nom L: identificateur }

Un jeton est un n-uplet contenant les informations spécifiques à la synchronisation. Par exemple, une demande de déstockage d'une pièce de l'entité A vers l'entité B sera représentée par un jeton de type:

$$< E_i, E_j , T_{ij} , N_k >$$

E_i : l'entité amont (A)

E_j : l'entité aval (B)

T_{ij} : le type de la liaison

N_k : le numéro de la pièce

Le réseau de Petri de la figure 2 modélise la synchronisation ainsi que les échanges de signaux entre deux entités quelconques d'un FMS, avec prise en compte de tous les types de liaisons que ce soit la liaison mobile-mobile, mobile-immobile, ou immobile-mobile.

Sémantique des places:

P_1 : demande de déstockage (émis par le superviseur)

P_2: fin de la primitive vérification de déstockage

P_3: fin de la primitive préparation de déstockage

P_4: réservation des données relatives aux entités

P_5: réservation de l'entité

P_6: demande de traitement (émis par le superviseur)

P_7: fin de traitement (émis par le superviseur)

P_8: fin de déstockage
P_9: place intermédiaire
P_{10}: nombre de places occupées dans l'entité
P_{11}: nombre de places libres dans l'entité
P_{12}: fin de stockage (émis par le superviseur)
P_{13}: préparation ou lancement de déstockage (selon le type de la liaison)
P_{14}: demande lancement de déstockage

Sémantique des transitions

t_1: vérification de déstockage
t_2: préparation de déstockage
t_3: déstockage
t_4: vérification de stockage
t_5: préparation de stockage
t_6: stockage
t_7: traitement
t_8: transition intermédiaire
t_9: transition intermédiaire
t_{10}: transition intermédiaire [après la fin de stockage, la conduite émet une demande de traitement vers l'entité concernée]
t_{11}: transition intermédiaire [après la fin de traitement, la conduite émet une demande de déstockage vers l'entité concernée]

Le réseau de Petri est un 9-uplet:

$N_0 = < P, T, Class, V, Pré, Post, A_{tc}, A_{ta}, M_0 >$

$P = \{ P_1 , P_2 , P_3 , P_4 , P_5, P_6, P_7, P_8, P_9, P_{10}, P_{11}, P_{12}, P_{13}, P_{14} \}$

$T = \{ t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10} , t_{11} \}$

Class = { entité, pièce, typeliaison }

V ≡ { X, Y, NP, TL } avec X et Y entité (amont et aval respectivement); NP ∈ Pièce et TL ∈ Typeliaison

Pré est une matrice 14 11, les éléments non nuls sont:

Pré [P_1 , t_1] = < X, Y, TL, NP >
Pré [P_7, t_{11}] < Y, NP >
Pré [P_2 , t_4] = < X, Y, TL, NP >
Pré [P_8 , t_6] = < Y >
Pré [P_3 , t_4] = <X, Y, TL, NP >
Pré [P_9 , t_3] = <X, Y, TL, NP >
Pré [P_4, t_1] = < X >
Pré [P_{10} , t_1] = < X >
Pré [P_4, t_4] = < Y >
Pré [P_{11} , t_4] = < Y >
Pré [P_4, t_7] = < Y >

Pré [P_{12}, t_{10}]= < Y, NP >
Pré [P_5 , t_4]= <X, Y >
Pré [P_{13} , t_2]= <X, Y, MM, NP >
Pré [P_5 , t_7]= <Y >
Pré [P_{13} , t_8]= <X, Y, I, NP >
Pré [P_6 , t_7]= < Y, NP >
Pré [P_{14} , t_9]= <X, Y, TL, NP >

Post est une matrice 14 x 11, les éléments non nuls sont
Post [P_2 , t_1] = < X, Y, TL, NP >
Post [P_{11} , t_3] = < X >
Post [P_4 , t_1]= < X >
Post [P_{12} , t_6]= < Y, NP >
Post [P_{13} , t_4]= < X, Y, TL, NP >
Post [P_5,t_6] = < Y >
Post [P_4 , t_4] = < Y >
Post [P_{10} , t_6] = < Y >
Post [P_3 , t_2,] = < X, Y, TL, NP >
Post [P_4 ,t_7] = < Y >
Post [P_{14} , t_5] = < X, Y, TL, NP >
Post [P_5 ,t_7] = < Y >
Post [P_9 , t_8] = < X, Y, I, NP >
Post [P_7 , t_7] = < Y, NP >
Post [P_9 , t_9] = < X, Y, TL, NP >
Post [P_6 , t_{10}] = < Y, NP >
Post [P_8 , t_3] = < X, Y, NP >
Post [P_1 , t_{11}]= < X, Y, TL, NP >

Atc : une application qui associe à chaque transition une condition faisant intervenir les variables formelles associées aux arcs d'entrée et les attributs des classes correspondantes.

Puisqu'on gère un ensemble de synchronisations alors:
* le franchissement de la transition t_1 nécessite que X. NomE = E_i (l'entité amont concernée par la synchronisation est E_i)
- le franchissement de t_4 nécessite que Y.NomE = E_j (entité aval)

Le RdP de la figure 2 modélise deux types de liaisons mobile- immobile ou mobile-mobile donc:
*pour que la transition t_8 soit franchissable, il faut que TL.NomL=I (liaison mobile-immobile ou inversement)
* pour que la transition t_2 soit franchissable, il faut que TL.NomL = mm
Pour les autres transitions il n'y a pas de conditions..

Ata : on associe cette application à la partie modélisant la conduite
* L'action associée au franchissement de t_{10} est Y := Y.Nom E
* L'action associée au franchissement de t_{11} est X:= Y. Nom E

Comme marquage initial on peut avoir par exemple:

$$M_0 = [< E_i, E_j, N_p, T_{ij},>, 0, 0, <E_i,>, 0,0,0,0,0, <E_j,>,0,0,0,0>]^T$$

FONCTIONNEMENT DU MODELE

On suppose initialement que la pièce est disponible sur l'entité amont, prête à être déstockée. Détaillons la charge de chaque primitive et les transferts de messages.

- Le superviseur émet une demande de déstockage vers l'entité amont, ceci est schématisé par la présence d'une marque $< E_i , E_j, T_{ij}, N_k >$ dans la place P_1, d'une marque $< E_i >$ dans la place P_4 et d'une marque $< E_j >$ dans la place P_{10} , ce qui permet à l'entité amont d'entrer dans la primitive vérification de déstockage représentée dans la figure 2 par la transition t_1 , le franchissement de cette dernière sera suivi par l'émission d'une demande de stockage vers l'entité aval ceci est matérialisé par la présence d'une marque $< E_i, E_j, T_{ij}, N_k >$ dans la place P_2 et par la libération des données de l'entité amont, c'est-à-dire par la restitution d'une marque $< E_i >$ dans la place P_4, et la décrémentation d'une unité du nombre de places occupées dans l'entité E_i .

L'entité aval E_j entre dans la primitive vérification de stockage représentée dans la figure par la transition t_4 , cette dernière est tirable dans le cas où il existe au moins une place de libre (soit $N_l > 0$), ceci peut être modélisé par la présence d'une marque $< E_j >$ dans la place P_{11} , la présence d'une marque $< E_j >$ dans la place P_4 signifie que les données de l'entité aval sont réservées , la présence d'une marque $< E_i >$ et d'une marque $< E_j >$ dans la place P_5 signifie que l'entité amont et l'entité aval sont réservées, le franchissement de la transition t_4 est suivi par la décrémentation d'une unité du nombre de places libres.

Dans le cas où les deux entités sont mouvantes, une demande de préparation de déstockage est émise vers l'entité amont, ceci est schématisé par la présence d'un jeton $< E_i , E_j , mm, N_k >$ dans la place P_{13} , ce qui permet de franchir la transition t_2, ce qui sera suivi par une émission d'une demande de préparation de stockage vers l'entité aval, ceci est schématisé par la présence d'un jeton dans la place P_3 .

La primitive préparation de stockage d'une pièce sur l'entité aval, représentée dans la figure par la transition t_5, peut être sensibilisée, après quoi une demande de lancement de déstockage est émise vers l'entité amont , ceci est matérialisé par la présence d'un jeton dans la place P_{14}, ce qui permet de franchir la transition intermédiaire t_9 et restituer la marque $< E_i, E_j, mm, N_k >$ dans la place intermédiaire P_9, ainsi la transition t_3 peut être tirée, le franchissement de cette dernière se traduit par l'incrémentation d'une unité du nombre de places libres dans l'entité amont , ce qui peut être schématisé par la présence d'un jeton dans la place P_{11}, et par l'émission d'un signal fin de déstockage vers l'entité aval ce qui peut être schématisé par la présence d'un jeton $< E_i , E_j, mm, N_k >$ dans la place P_8, ainsi l'entité aval peut entrer dans la primitive stockage représentée dans la figure par la transition t_6 , le franchissement de cette dernière entraîne l'incrémentation d'une unité du nombre de places occupées dans l'entité aval, schématisée par la présence d'une marque $< E_j >$ dans la place P_{10}, la libération des deux entités respectives E_i et E_j donc la restitution des deux marques $< E_i >$ et $< E_j >$ dans la place P_5 et enfin l'émission d'un signal fin de stockage vers la conduite qui est schématisée par la présence d'une marque $< E_j , N_k >$ dans la place P_{12} .

- La présence d'une marque de type $< E_j , N_k >$ dans la place P_6, d'une marque $< E_j >$ dans la place P_5 et d'une marque $< E_j >$ dans la place P_4 signifie que la primitive traitement, représenté dans la figure par la transition t_7 peut être exécutée, le franchissement de cette dernière va engendrer une émission d'un message fin de traitement , matérialisé par la présence d'un jeton $< E_j , N_k >$ dans la place P_7, ainsi, un jeton de type $< E_j >$ est restitué dans la place P_4 et la libération de l'entité va entraîner la

restitution d'un jeton dans la place P_5, par suite, la conduite peut émettre une demande de déstockage (l'entité aval précédente, devient l'entité amont à la prochaine synchronisation)

- Dans le cas ou l'une des deux entités est immobile, on répète le même cheminement des primitives que précédemment , en omettant à chaque fois, l'exécution des primitives, préparation de déstockage, représentée dans la figure par la transition t_2 , et préparation de stockage, représentée dans la figure par la transition t_5 .

4. CONCLUSION

L'approche que nous avons proposée est une approche partielle couvrant la phase de spécification. Notre modèle fournit d'une part une grande puissance de description nécessaire à la maîtrise de la complexité inhérente au modèle générique de synchronisation, et d'autre part, des possibilités de validations permettant la détection de blocages éventuels, il autorise en outre une validation préalable de la commande des FMS_S.

BIBLIOGRAPHIE

D'Souza (1994) K.A . D' S OUZA, and S.K. KHATOR: *A survey of Petri net applications in modeling controls for automated maufacturing systems:* Computers in Industry, Vol.24, 1994, pp.5-16.

Grûnenberger(1994) E. Grûnenberger, *" PC Code: Une méthodologie opérationnelle pour concevoir le système de conduite de cellules de production"*. Thèse de Doctorat de l'université de Tours, Ecole d'Ingénieurs en Informatique pour l'industrie, le 16 décembre 1994.

Ho (1989) V.C.Ho, *" Scanning the issue, Dynamics of Discrete Event Systems "*, Proceedings of the I.E.E.E. , Special Issue on Dynamics of Discrete Event Systems, vol. 77, n° 1, jan.1989.

Martineau(1996) P.Martineau, J- L - Labesse, C. Carmier, F. Cotard, C.Proust, *A generic model for Flexible Manufacturing Systems (FMS),* Workshop on production Planning and control, Mons, Belgium, 9-11 september 1996

Proth(1995) J-M.Proth, X. Xie, *Les réseaux de Petri pour la conception et la gestion des systèmes de production*, Masson 1995

Proust (1989) C.Proust, J-P. Asselin de Beauville, A. Dolla, A.Rouillon: *Une chaine de production en réduction à l'université: outil pédagogique et support de recherche*, Revue d'Automatique et de Productique Appliquées, vol.2, n° 2, pp.43-70, 1989.

Richard P, Grunenberger E (1993) " Squelette de synchronisation: un modèle de contrôle de processus concurrents", Rapport interne n° 131 du LI/E3I/Univ Tours 22 pages 1993.

Sibertin-Blanc(1988) Sibertin-Blanc, *" Le prototypage des applications interactives à l'aide des réseaux de Petri"*, Journées Internationales "Le génie logiciel et ses applications", pp.837-856, Toulouse (France), Décembre 1988.

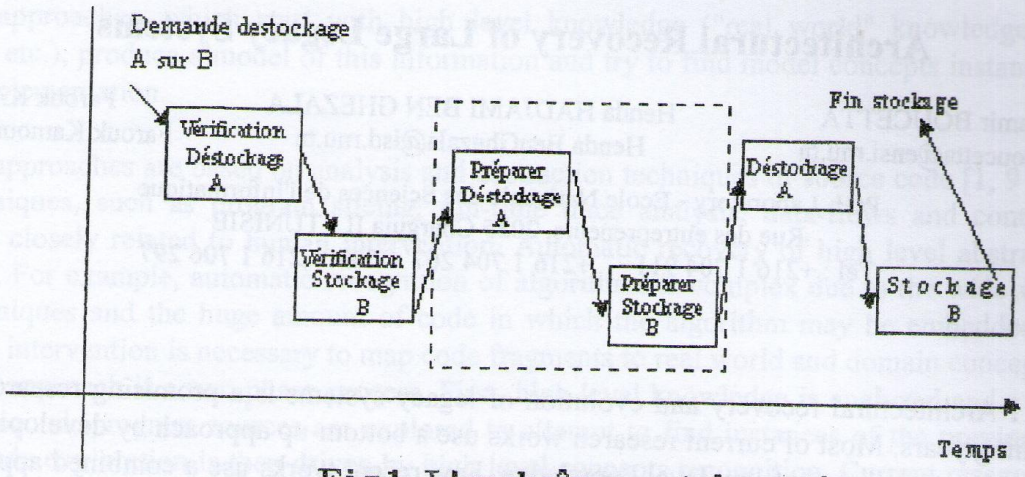

Fig.1. L'enchaînement des actions

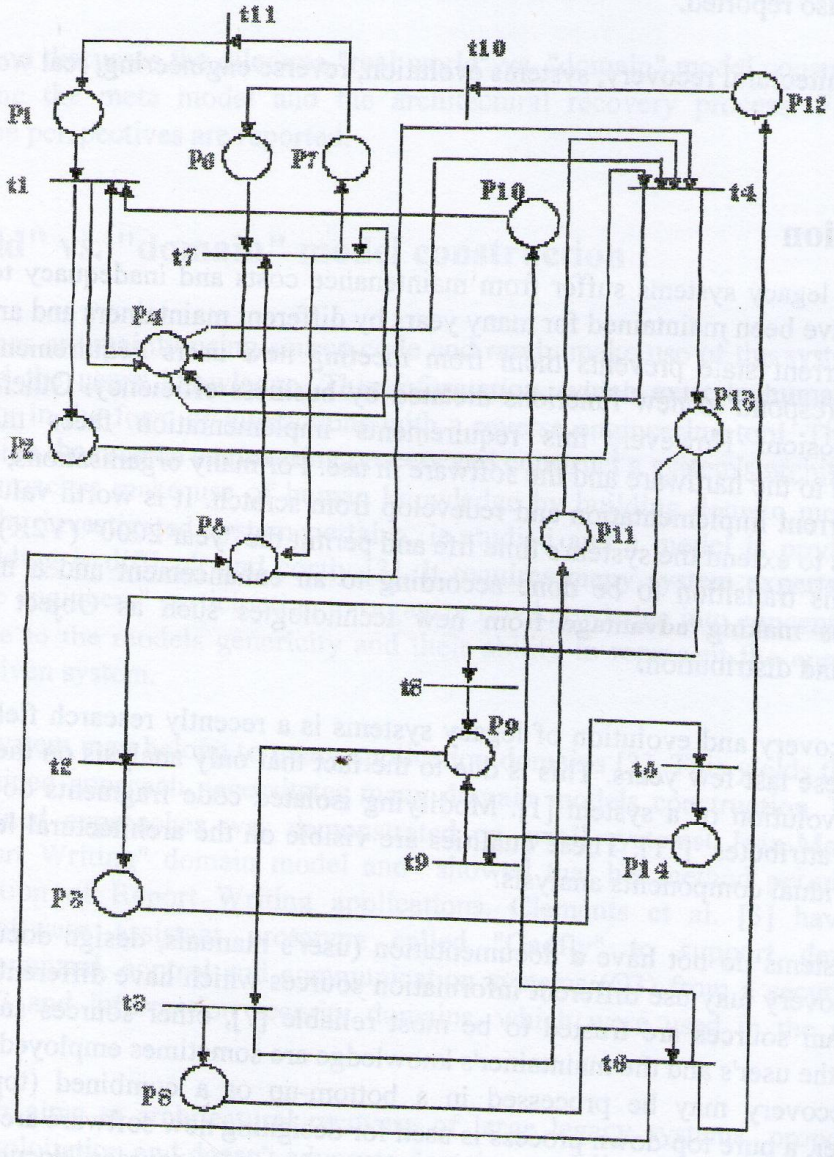

Fig.2. Spécification du MGS