

MODELES DE SIMULATION ORIENTES OBJETS POUR L'EVALUATION DES PERFORMANCES DES SYSTEMES FLEXIBLES DE PRODUCTION AVEC UN CHARIOT FILOGUIDE

Anthony Caumond

Institut Supérieur d'Informatique et de Modélisation et
de leurs Applications (ISIMA),
Campus de Clermont-Ferrand
Les Cézeaux B.P. 125
63173 Aubière Cedex France
Mail: caumond@poste.isima.fr

Nikolay Tchernev

Université Blaise Pascal Clermont-Ferrand II,
Laboratoire d'Informatique, de Modélisation et
d'Optimisation des Systèmes (LIMOS),
Campus Universitaire des Cézeaux,
63177 Aubière Cedex France
Mail : tchernev@isima.fr

Résumé : Cet article traite des phases d'analyse et de spécification pour la réalisation de modèles de simulation à événements discrets pour les systèmes flexibles de production avec une unique ressource de transport. Les modèles de simulation permettent l'évaluation des critères de performance du système (makespan, taux d'utilisation des ressources, politiques de gestion des stocks...) en prenant compte des contraintes de fonctionnement (stocks d'entrée/sortie de capacité limitée, nombre de pièces simultanément autorisées limité...). Un modèle de simulation permet de tester les solutions obtenues avec des méthodes exactes en prenant compte des contraintes qu'il a été impossible de formaliser. Nous détaillons le modèle de connaissance des systèmes flexibles de production avec une seule ressource de transport et nous présentons les principes d'implantation de ce modèle générique. Deux modèles de simulation génériques pour la classe de systèmes (l'un en C++ et l'autre Delphi) ont été auto-validés par des tests réalisés sur les exemples fournis par Ulusoy et Bilge en 1993 et ils ont été validés par un modèle dans un langage de simulation dédié (QNAP2). Les modèles obtenus en C et en Delphi se sont révélés 70 fois plus performants que le modèle QNAP2.

Mots-clés : systèmes flexibles de production, méthodologie de modélisation, évaluation des performances, modèles de simulation, UML.

1 INTRODUCTION

1.1 Les systèmes flexibles de production avec un chariot

Un système flexible de production est selon MacCarthy et Liu (MacCarthy et Liu, 1993) : « un système de production piloté par ordinateur capable de produire différents types de pièces, composé de machines à commande numérique ou à contrôle numérique et d'un système automatisé de stockage connecté par un système automatisé de manutention ». De manière plus formelle, un système flexible de production avec un chariot filoguidé est un atelier automatisé composé de stations et d'un système de manutention utilisant un chariot filoguidé qui assure le transport des pièces entre les stations ainsi que le transport entre les stations et la (ou les) station(s) d'entrée/sortie.

L'atelier se compose de g gammes et chaque gamme définit la liste des stations sur lesquelles une pièce doit être traitée et le temps de traitement de la pièce sur chaque station. Un ensemble de n pièces sont présentes en entrée de l'atelier. Chaque station se compose d'un stock d'entrée, d'un poste d'usage et d'un stock de sortie. Le système de gestion des chariots ne permet pas d'anticiper la satisfaction des demandes de transport des

pièces. Les politiques à prendre en compte sur ce type de systèmes sont :

- les politiques de gestion du chariot lorsque au moins une pièce se trouve dans un stock de sortie d'une station,
- les politiques de gestion du chariot dans le cas où aucune pièce ne se trouve dans un stock de sortie d'une station,
- les politiques de choix d'une pièce à traiter dans le stock d'entrée d'une station,
- les politiques de choix de la prochaine pièce à transporter dans le stock de sortie d'une station.

1.2 Problème d'évaluation des performances

Les approches mathématiques et les approches par simulation sont utilisées pour la planification et l'évaluation des performances des systèmes flexibles de production. L'avantage des méthodes mathématiques est qu'elles fournissent rapidement des résultats mais elles nécessitent généralement de simplifier le fonctionnement du système pour leur mise en œuvre. La simulation peut donner une vue globale du fonctionnement du système et permet de prendre en compte toutes les contraintes de fonctionnement (règles de gestion) et les caractéristiques du système (capacité limitée des stocks des machines...). Il semble qu'une approche prometteuse consiste à évaluer les

performances ou à optimiser le fonctionnement du système avec des méthodes exactes et à tester les résultats par un modèle de simulation qui permet de reproduire un fonctionnement plus réaliste du système. Ensuite, les résultats peuvent être localement optimisés par des algorithmes stochastiques, tels que le recuit simulé ou la descente stochastique (Lacomme et al., 2000), en modifiant les politiques de gestion et/ou l'ordonnancement. En simulation, l'évaluation d'une solution peut passer par des critères variés comme le taux d'occupation du chariot à vide et à charge, le taux d'occupation des machines... De plus, la simulation permet de reproduire la dynamique du système et d'étudier ses états transitoires.

1.3 Les outils de simulation à événements discrets

La réalisation de modèles de simulation peut se faire à l'aide : de langages de programmation généraux (tels que C, C++, ADA...), de langages de simulation (tels que : GPSS, SLAM ou SIMAN), d'environnement de simulation (tels que ARENA, ISI, FACTOR) et d'environnement de modélisation tels que SIGMA et GESICA. Les environnements s'appuient généralement sur de la génération de code dans un langage de simulation (GPSS, SLAM, SIMAN...). Ces langages peuvent être utilisés directement et fournissent un ensemble de primitives de base. Ce sont plus souvent des langages interprétés. Une autre solution consiste à réaliser des modèles de simulation dédiés à l'aide de langages généraux. Ils nécessitent de s'intéresser aux mécanismes de gestion de la simulation, mais ils ont l'avantage d'être facilement accessibles, peu chers, disponibles et très bien connus. Les modèles obtenus ne nécessitent pas d'investissements financiers particuliers et les modèles obtenus sont très performants.

2 Modèle générique orienté-objets d'un système flexible de production

2.1 Description générale d'un système flexible de production avec un chariot filoguidé

Le modèle générique complet est présenté dans (Caumond, 2000). Il comprend deux grands sous-systèmes : le sous système de production et le sous-système de transport.

Les pièces entrent dans le système par la station d'entrée et sortent du système par la station de sortie. Une fois dans le système, les pièces sont transportées d'une station à l'autre par le sous-système de transport. Chaque station se compose d'une machine et de deux stocks. Les pièces passent respectivement par trois entités : le stock d'entrée, la machine et le stock de sortie. Les stocks sont de capacités limitées, de plus une pièce ne peut être déposée dans ce stock que si une place est libre. Chaque station possède indépendamment ses propres caractéristiques : les capacités de son stock d'entrée et de son stock de sortie, les politiques de

gestion des stocks. La station d'entrée (respectivement la station de sortie) n'est qu'une station particulière composée uniquement d'un stock de sortie (respectivement d'entrée) (Figure 1). La politique de gestion du stock d'entrée d'une station permet de déterminer quelle est la prochaine pièce à traiter lorsque le traitement précédent est terminé et que plusieurs pièces sont disponibles dans le stock d'entrée.

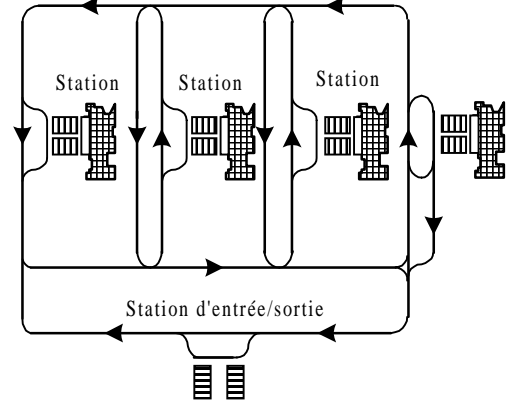


Figure 1. Exemple de système flexible de production

La politique de gestion du stock de sortie permet de déterminer quelle pièce sort en premier du stock lorsque plusieurs pièces sont dans le stock de sortie et que le sous-système de transport vient de prendre une pièce sur cette station. Les règles de gestion sont dynamiques de sorte que les choix effectués sont réalisés pendant que le système évolue dans le temps. L'architecture de ces systèmes et leur mode de fonctionnement interdit toute préemption et toute machine commençant le traitement d'une pièce le termine sans interruption possible.

Le sous-système de transport est composé d'un réseau de transport qui supporte un seul chariot qui ne porte qu'une seule pièce. Le chariot est géré par le superviseur du système qui lui indique sur quelle station il doit se déplacer pour chercher une pièce dans le stock de sortie et sur quelle station il doit déposer cette pièce. Dans le cas où aucune pièce n'est présente dans un stock de sortie le superviseur maintient alors le chariot dans une position d'arrêt (règle de gestion STOP) ou bien lui fait parcourir un chemin pré-calculé dans le réseau (règle de gestion MOVE). La position exacte du chariot est connue par l'intermédiaire de capteurs qui se trouvent au niveau des différentes stations. Lorsque le chariot se déplace dans le réseau entre deux stations, son déplacement ne peut pas être interrompu car en pratique l'ordre complet de transport est chargé dans le système de contrôle embarqué du chariot. Le chariot se déplace toujours entre deux stations suivant le chemin le plus court qui est pré-calculé et disponible en permanence dans le superviseur.

Le parcours des pièces dans le système est induit par leur gamme. Chaque gamme définit la suite des stations sur lesquelles une pièce doit passer et les temps de traitement sur chaque machine.

2.2 Hypothèses

La description faite ci-dessus sous-entend les hypothèses suivantes :

- toutes les pièces sont disponibles dès le début du fonctionnement du système.
- les temps de traitement des pièces sur les machines sont donnés par les gammes et les événements aléatoires ne perturbent pas le fonctionnement des machines. La même remarque peut être faite sur les temps de déplacement du chariot qui sont supposés constants et connus.
- les gammes sont connues à l'avance.
- les moyens de manutentions sur les stations sont supposés indépendants, c'est-à-dire que trois pièces peuvent être en mouvement simultanément sur une même station. Ce cas se produit quand une pièce est chargée, et qu'une pièce passe dans la machine pendant qu'une troisième va dans le stock de sortie. Cette hypothèse peut être levée en décomposant la station plus précisément. Mais ce niveau de détail n'est pas nécessaire dans le cas général.
- les traitements des pièces sur les machines sont non-interruptibles.
- les déplacements du chariot entre deux stations sont non-interruptibles.
- le chariot est géré par le superviseur dynamiquement suivant une règle de gestion telle que FIFO, STT... qui est utilisé chaque fois qu'il y a au moins une pièce dans un stock de sortie d'une station.
- lorsqu'il n'y a pas de pièce dans un stock de sortie, le chariot est géré par le superviseur suivant la règle STOP ou bien la règle MOVE

2.3 Le modèle de classe

Le schéma de la figure 2 donne une vue d'ensemble des principales classes d'objets d'un système flexible avec un chariot. Par rapport à la description du paragraphe précédent nous avons fait explicitement apparaître la

notion de charge du système comme étant l'ensemble des pièces à traiter ainsi que les gammes.

Nous avons mis l'accent sur les relations de type composition afin de montrer les liens avec la présentation du paragraphe précédent. Parmi les nombreuses relations entre les différentes classes nous montrons celles qui existent entre les pièces et les autres classes car ce sont ces relations qui sont les plus importantes puisqu'une pièce constitue un élément de flux dans le système. Le schéma de la figure 2 fait apparaître qu'une pièce se trouve soit dans un chariot soit dans une station. Le mot station désigne la classe station sachant que toute station se compose d'un stock d'entrée, d'un stock de sortie et d'une machine. Toute la dynamique du système réside dans les mouvements des pièces qui vont passer du chariot au stock d'entrée d'une station, puis du stock d'entrée sur la machine et enfin de la machine sur le stock de sortie de la station. Au cours de ses déplacements la pièce change d'état (elle progresse dans sa gamme etc.) et les entités traversées changent, elles aussi, d'état. La dynamique du système pourrait être décrite par un modèle dynamique du système. Nous avons choisi de détailler ici le modèle de classe en justifiant les choix fait au niveau des attributs et des méthodes de chaque classe. Les autres modèles sont détaillés dans (Caumond, 2000).

Dans la classe Chariot l'attribut Position est en fait le numéro de la station devant laquelle le chariot se trouve. Les positions successives du chariot le long du réseau de transport ne sont pas représentées puisqu'il s'agit d'états transitoires et que les déplacements du chariot sont non-interruptibles. Donc le simulateur ne doit gérer que les positions des chariots qui sont devant une station. Le chariot comporte un attribut Vitesse_A_Charge et Vitesse_A_Vide, avec la matrice des distances on peut donc calculer le temps de transport. La méthode Déplacer_Chariot permet de déplacer le chariot de sa position actuelle (donnée par son attribut Position) vers une position de destination.

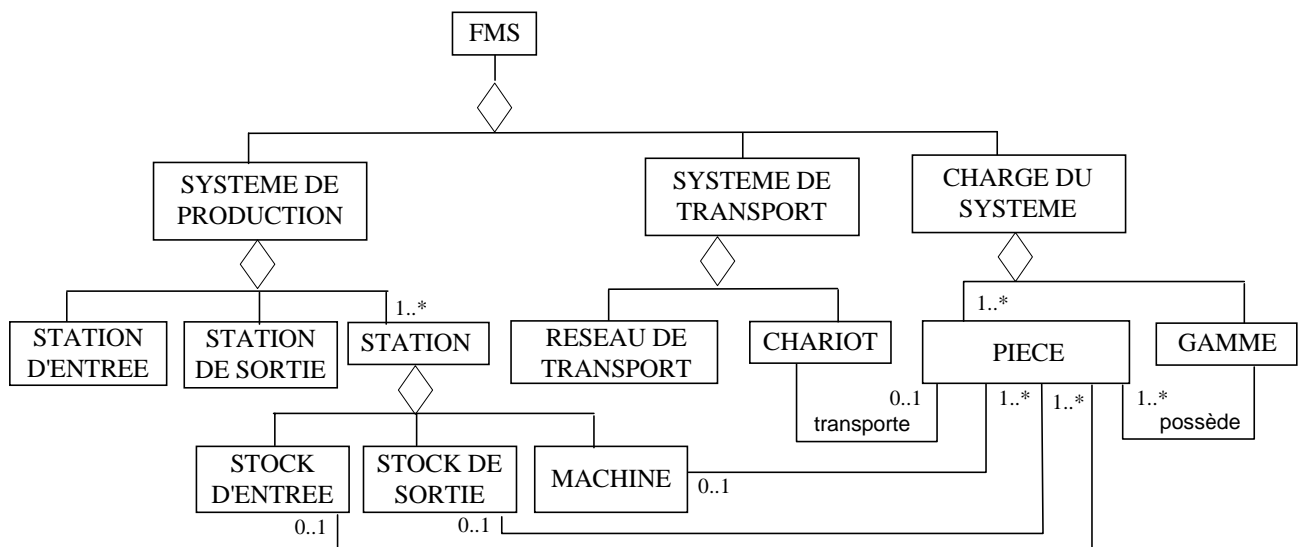


Figure 2 Diagramme de classe



Figure 3 Détails des principales classes du modèle

La méthode Poser_pèce permet de poser une pièce dans un stock d'entrée d'une station et la méthode Prendre_pèce permet de prendre une pièce dans un stock de sortie d'une station.

La classe Superviseur représente le système de gestion du système flexible. La Matrice_des_Distances est une matrice donnant la distance (chemin le plus court) nécessaire au chariot pour aller de la station *i* à la station *j*. Dans un système composé d'un seul chariot il est toujours possible de considérer que le temps de déplacement est connu ou bien calculable avant de démarrer le modèle de simulation. L'attribut Règle_de_gestion_1 désigne la règle à utiliser pour gérer le chariot lorsque au moins une pièce est dans un stock de sortie. Typiquement, cet attribut représentera la règle FIFO (First In First Out), STT (Shortest Travel Time), SPT (Shortest Processing Time) ou encore MOQS (Maximum Output Queuing Size). L'attribut Règle_de_gestion_2 désigne la règle à utiliser lorsqu'il n'y a pas de pièce dans un stock de sortie d'une station, deux possibilités ont été implantées : règle STOP et règle MOVE.

3 Le noyau de synchronisation (Scheduler)

Au schéma précédent, il faut rajouter une classe supplémentaire, la classe Scheduler qui représente le noyau de synchronisation et dont le rôle est de gérer les événements qui vont se produire dans le modèle (Figure 4).

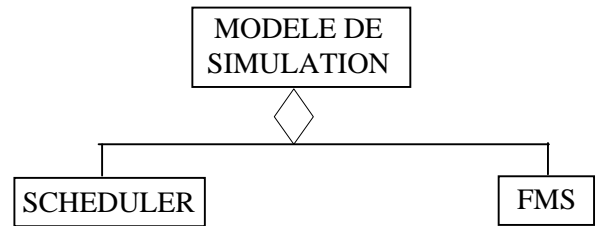


Figure 4. Le modèle objet du modèle de simulation avec le Scheduler

A partir du modèle d'atelier flexible de production précédemment présenté nous pouvons décrire n'importe quel système de cette classe. L'instanciation du modèle générique en un modèle d'action se fait en renseignant les différents attributs et méthodes des différentes classes. Le modèle de simulation obtenu permettra d'évaluer n'importe quel critère de performance du système flexible auquel on s'intéresse. Quel que soit le système flexible dont on simule le fonctionnement, les algorithmes implantés dans le Scheduler restent valables. Pour procéder à l'évaluation, le Scheduler génère tous les événements du modèle à partir de l'état initial et la simulation se termine lorsqu'il n'y a plus d'événements à gérer, ce qui peut avoir deux origines : un blocage (deadlock) ou la fin du traitement de la charge. La difficulté consiste à gérer les événements dus aux interactions entre les entités. On peut faire les remarques suivantes :

- les déplacements du chariot étant non-interruptibles, les changements d'états des stations n'influencent pas directement les décisions de gestion du chariot.
- les changements d'états des stations n'ont pas de liens entre eux d'une station à l'autre. Les changements d'états d'une station ne dépendent pas des changements d'états des autres stations.

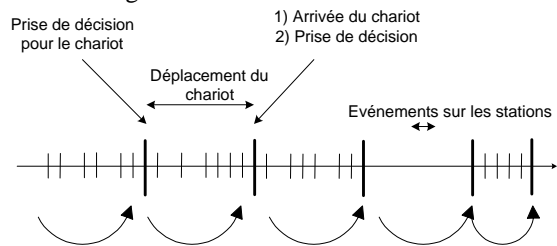


Figure 5. Les événements à gérer

Il est donc possible de distinguer les événements qui sont liés aux mouvements du chariot et ceux liés aux stations (Figure 5). Les événements qui concernent le chariot n'ont lieu qu'à la fin du transport d'une pièce ou bien lorsqu'un événement déclenchant la fin d'une attente chariot se produit (par exemple : arrivée d'une pièce dans le stock de sortie d'une station).

Il n'y a donc que deux sortes d'événements à gérer : la fin d'un transport (i.e. déplacement à vide, chargement, déplacement en charge et déchargement) et la fin d'une situation d'attente du chariot. Ainsi, la gestion des

événements simultanés est plus simple, il n'y a que deux cas à gérer, les autres événements étant gérés de façon rétroactive par le noyau de synchronisation lors de la mise à jour. Une façon efficace d'implanter cette approche consiste à doter chaque objet (autre que les éléments de flux) d'une date courante, le noyau de synchronisation (Scheduler) gérant la date de la simulation. Il faut aussi que chaque objet soit capable de donner la date du prochain événement qui le concerne et qu'il peut prévoir localement. Si cette prévision n'est pas possible en utilisant seulement la connaissance locale (propre à l'objet) du système, une valeur arbitrairement grande (supérieure à toutes les autres dates possibles) peut être toutefois renvoyée par l'objet.

La mise à jour du système se fait jusqu'à la plus petite date sur les objets actifs. Des algorithmes spécifiques ont du être développé pour faire les mises à jour du noyau de synchronisation et les mises à jour des stations. Lorsque la date du prochain événement du chariot est supérieure à la date du prochain événement sur une station et que le chariot est oisif, le prochain événement est l'arrivée de la pièce dans le stock de sortie. Le chariot est oisif si aucune pièce ne peut être traitée par le chariot à l'instant où l'on considère le système.

Le principe des algorithmes implantés est présenté sur figure 6, Figure 7, Figure 8. La classe Modèle de Simulation permet essentiellement d'initialiser le modèle (lecture dans des fichiers) et de restituer les résultats de la simulation. Elle appelle le Scheduler qui va (à partir de l'état initial du système) faire évoluer le modèle de simulation. Le Scheduler gère (entre autre) l'horloge de la simulation qui est mise à jour (compte tenu de ce qui précède) uniquement à la fin d'un transport par le Superviseur dans la procédure Gérer_Système. Normalement le Scheduler ne devrait pas avoir à modifier directement l'horloge de la simulation. Mais l'approche que nous avons choisie nous impose de faire autrement. En effet, le chariot (seul objet du système à faire évoluer l'horloge de la simulation) peut se trouver dans l'état OISIF c'est-à-dire dans un état où il n'est pas possible de lui assigner un déplacement.

La méthode Calculer_Destination du Superviseur détermine sur quelle station le chariot doit être déplacé. Si aucun déplacement n'est possible (il n'y a aucune pièce dans les stocks de sortie) alors le chariot est oisif. Deux cas sont à prendre en compte. Si la règle de gestion est STOP, aucune station de destination ne peut être attribuée et Destination prend la valeur -1. Si la règle est MOVE on calcule une destination en fonction de la position actuelle du chariot et du chemin pré-calculé. Si le chariot est plein, la destination du chariot se déduit de la gamme de la pièce transportée. Si le chariot est vide et s'il existe des pièces dans un stock de sortie, la destination du chariot est calculée en utilisant la règle de gestion numéro 1.

Si aucune station de destination ne peut être attribuée au chariot, il se trouve dans l'état OISIF dans lequel il reste aussi longtemps qu'une pièce n'apparaît pas dans le stock de sortie d'une machine. Donc si le chariot est OISIF et s'il n'est pas possible de lui affecter un déplacement alors il faut mettre à jour la date de la simulation qui fait un « saut » jusqu'à la date d'arrivée de la prochaine pièce dans un stock de sortie (Figure 6, pas 5-6-7-8 de l'algorithme du Scheduler).

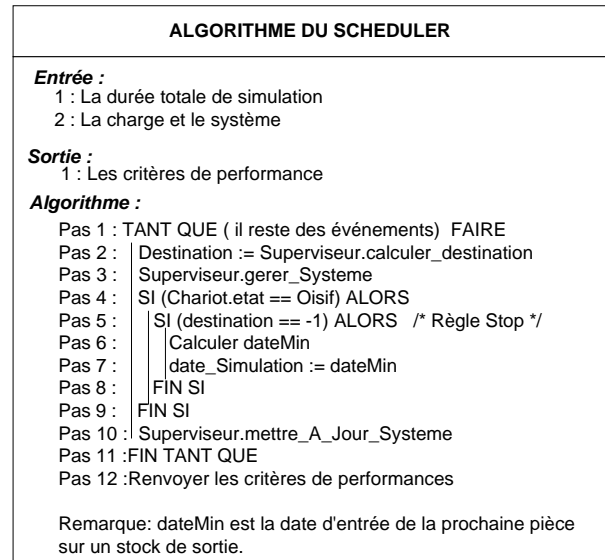


Figure 6. Détails de l'algorithme du Scheduler

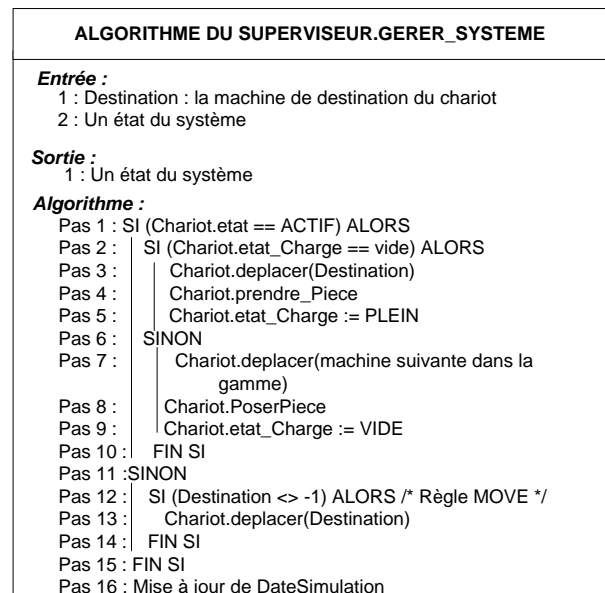


Figure 7 Détail de l'algorithme de gestion des chariots du superviseur

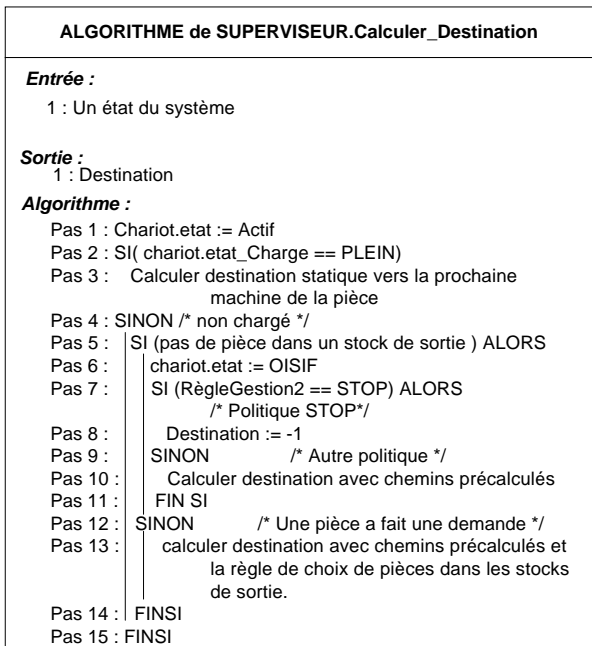


Figure 8. Détails de l'algorithme de gestion des déplacements du Superviseur.

La méthode Gérer-Système gère les changements d'états des entités en fonction des déplacements du chariot. On distingue à ce niveau les déplacements lorsque le chariot est oisif et lorsque le chariot est actif. Dans le cas où le chariot est oisif, si la règle de gestion numéro deux est MOVE, la seule action à réaliser un déplacement du chariot. Notons que dans le cas où la règle est STOP, aucune action n'est à réaliser sur le système par la procédure. Lorsque le chariot est actif il faut mettre à

jour les entités suivant que le chariot charge ou décharge.

4 Étude de cas

Pour l'évaluation de notre modèle de simulation nous avons repris les exemples proposés par Ulusoy et Bilge (Ulusoy et Bilge, 1993). Chacune des topologies (« layout ») proposées par Ulusoy et Bilge se compose d'une station d'entrée/sortie et de quatre stations de traitement des pièces. On suppose que la station d'entrée/sortie a une capacité suffisante pour que l'ensemble des pièces à traiter et celles déjà traitées puissent être stockées. Les gammes utilisées et les temps de déplacement du chariot sont ceux proposés dans (Ulusoy et Bilge, 1993). Un « Job-Set » désigne l'ensemble des données sur les gammes et les temps de déplacement. A titre d'exemple, nous considérons le « layout » numéro 1 avec le « job-set » 1 pour lequel nous supposons que les pièces sont ordonnancées en entrée du système suivant l'ordre : 5 4 1 1 3 3 2 2 où les chiffres représentent les numéros de gamme des différentes pièces.

L'exemple présenté concerne un FMS avec un unique chariot géré par les politiques suivantes :

- la politique du chariot lorsque au moins une pièce se trouve dans un stock de sortie est FIFO.
- la politique du chariot lorsque aucune pièce n'est dans un stock de sortie est STOP.
- la politique utilisée pour choisir une pièce dans le stock d'entrée d'une station est FIFO.
- la politique utilisée pour choisir une pièce dans le stock de sortie d'une station est FIFO.

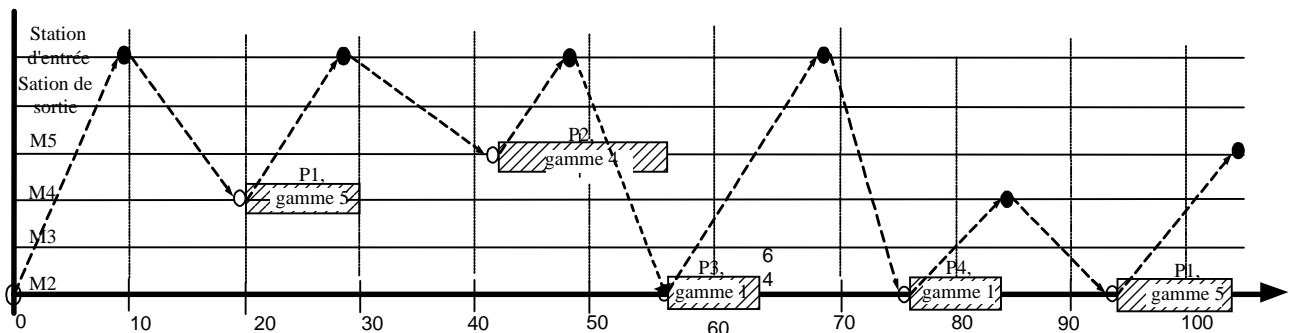


Figure 9 Mouvements du chariot pendant la montée en charge

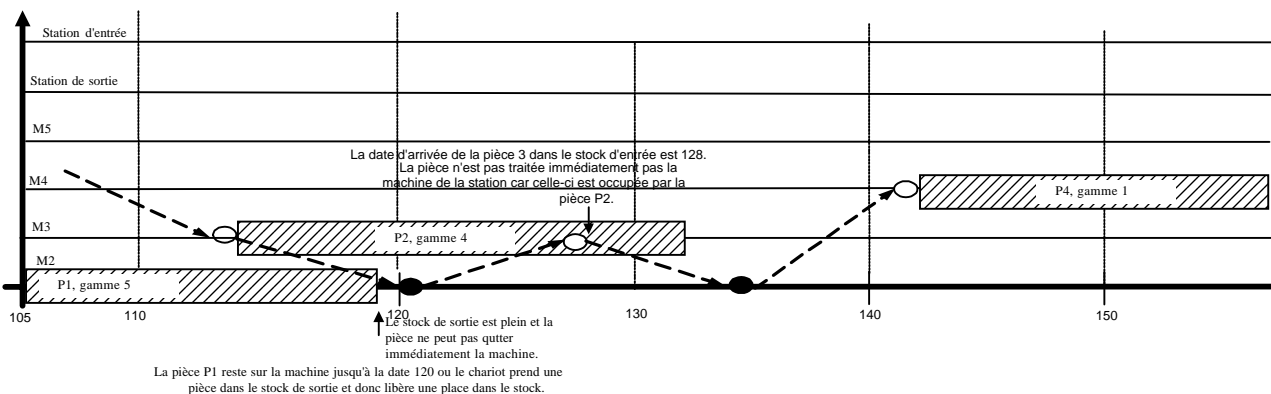


Figure 10 Mouvements du chariot après la montée en charge

Le nombre de pièces simultanément autorisées dans le système est 4. Les temps de chargement et de déchargement ont été fixés à 1 minute.

Tant que c'est possible, le chariot fait entrer des pièces dans le système. Le chariot cesse de faire entrer les pièces dans les cas où : 1) la prochaine pièce va être transportée sur une station dont le stock d'entrée est plein ; 2) le nombre de pièces autorisées dans le système est atteint. Dans l'exemple de la figure 9, le chariot fait entrer 4 pièces et devient oisif puisque le nombre maximal de pièces simultanément autorisées dans le système est 4.

La figure 10 donne un exemple de quelques déplacements effectués par le chariot après la montée en charge, quelques cas intéressants y apparaissent. Par exemple, la pièce P1 de gamme 5 termine son traitement sur la machine 2 à la date 119. Mais cette pièce ne sera dans le stock de sortie de la station qu'à la date 120, c'est la date à laquelle le chariot prend une pièce dans le stock de sortie de la station.

La tableau de la figure 13 donne la descente en charge et la sortie de la dernière pièce. Ce tableau est une trace améliorée du fonctionnement du système. Il met en évidence quelques cas d'attente particuliers.

% d'activité	% temps en déplacement à charge	% temps en déplacement à vide	% temps en chargement/déchargement
100	59	27	14

Figure 11. Critères de performances du chariot

	Taux d'occupation de la machine	Nombre moyen de pièces dans le stock d'entrée	Nombre moyen de pièces dans le stock de sortie
Station 1	23.92	0.000	0.600
Station 2	20.18	0.005	0.457
Station 3	12.62	0.000	0.475
Station 4	13.57	0.000	0.429

Figure 12. Critères de performances des stations.

Les Figure 11 et Figure 12 présente des critères de performances. L'exécution du modèle C++ prend 0.009s sur un PENTIUM II 350Mhz sous Windows 98 pour 50 pièces en entrée. Le modèle QNAP2 V7 nécessite environ 0.6s sur un IBM RISC 6000.

5 Conclusion

Nous avons présenté une partie d'un modèle générique orienté-objet de la classe des systèmes flexibles de production composés d'un seul chariot. A partir de ce modèle générique nous avons également présenté les

date de la simulation	position du chariot	destination	pièce transportée	Gamme	Commentaire
389	2	Sortie	5	3	
401	Sortie				Début déchargement
402	Sortie				Fin déchargement
402	Sortie	2			
408	2				Début chargement
409	2				Fin chargement
409	2	Sortie	6	3	
421	Sortie				Début déchargement
422	Sortie				Fin déchargement
422	Sortie	4			
432	4				Début Chargement
433	4				Fin chargement
433	4	3	7	2	
439	3				Début déchargement
440	3				Fin déchargement
440	3	4			
446	4				Début chargement
447	4				Fin chargement
447	4	3	8	2	
453	3				Début déchargement
454	3				Fin déchargement
454	3				Attente
458	3				Nouvel événement
458	3				Début Chargement
459	3				Fin chargement
459	3	Sortie	7	2	

Figure 13 Descente en charge.

principes généraux mis en œuvre pour réaliser un modèle de simulation dédié dans un langage de programmation général. L'objectif est de fournir un cadre efficace en temps d'exécution pour pouvoir évaluer les performances, optimiser le fonctionnement du système avec des méthodes analytiques et ensuite tester les résultats par un modèle de simulation qui permet de reproduire un fonctionnement plus réaliste du système.

A partir du modèle objets et des algorithmes proposés nous avons développé deux modèles (en C et en DELPHI) qui ont été auto-validés par des tests effectués sur les jeux de données proposés par Ulusoy en 1993. Les résultats montrent que les modèles sont efficaces (environ 70 fois plus rapides que le modèle écrit dans langage général de simulation) et que des critères de performances très variés peuvent être évalués avec une grande précision.

Les perspectives de ce travail s'inscrivent dans le développement d'une interface graphique de spécification d'un système de la classe des systèmes flexibles avec un seul chariot qui permette de générer automatiquement un modèle de simulation. Nous envisageons de nous intéresser à la réalisation de modèles pour des systèmes flexibles à plusieurs chariots. Nous envisageons aussi d'améliorer ce simulateur en intégrant des éléments aléatoires. En effet

les systèmes flexibles sont soumis à des interruptions et à des phénomènes stochastiques tels que les pannes. Pour prendre en compte ces éléments il faut retravailler l'analyse pour définir et intégrer de nouvelles politiques de gestion des pannes. Les temps de transport ou de traitement sont généralement mieux modélisés par une loi de distribution statistique, en intégrant des calculs d'intervalle de confiance, nous pourrions intégrer ces éléments en se basant sur l'approche orienté-objets.

6 Bibliographie

- Caumond A., Modélisation et simulation des ateliers flexibles de production, Rapport de stage de deuxième année, ISIMA, Clermont-Ferrand, 2000.
- MacCarthy B.L. et J. Liu, A New Classification Scheme For Flexible Manufacturing Systems, International Journal of Production Research, vol 31 (2), pp 299-309, 1993.
- Lacomme P., A. Moukrim et N. Tchernev, Résolution conjointe de problèmes d'ordonnancement et de tournées de véhicules : couplage Branch and Bound - simulation à événements discrets. Congrès ROADEF, Ecole des Mines de Nantes, pp. 128-129, 2000.
- Ulusoy G. et Ü. Bilge, Simultaneous Scheduling of Machines and Material Handling System in an FMS, International Journal of Production Research, vol. 31, pp. 2857-2873, 1993.