

Développement d'un logiciel d'ordonnancement d'un système de production

N. Keddari¹, A. Hassam², Z.Sari³

^{1,2,3}Univ. Abou-Bakr Belkaid B.P. 119 Tlemcen, Algérie.

¹*keddarinassima@yahoo.fr*

²*a_hassam@mail.univ-tlemcen.dz*, ³*z_sari@mail.univ-tlemcen.dz*

Résumé- Les problèmes d'ordonnancement sont des problèmes souvent rencontrés dans l'industrie ou l'ingénierie. Dans le système d'exploitation, le noyau doit gérer les priorités et l'organisation des processus qui s'exécutent en même temps, ce qui correspond à un problème d'ordonnancement. De même, nous avons des exemples dans beaucoup d'autres disciplines. La gestion de production et le fonctionnement des ateliers d'une industrie sont régis par des règles d'ordonnancement. Pour cela et vu l'importance de l'ordonnancement pour les systèmes de production, on va essayer de développer un logiciel d'ordonnancement dans le quel on va intégrer une grande majorité des principales règles d'ordonnancement qui existe dans la littérature. Ce logiciel va être réalisé avec un langage de programmation orienté objet tel que C++.

Mots clefs- Règles d'ordonnancement, Jobshop, Système de production, Simulation, Langages de programmation.

Abstract- Scheduling problems are problems often encountered in industry and engineering. In the operating system, the kernel must manage priorities and organization of processes running simultaneously, which corresponds to a scheduling problem. Similarly, we have examples in many other disciplines. Production management and operation of an industry workshops are governed by rules of scheduling. For this and given the importance of scheduling for production systems, we will try to develop scheduling software in which it will integrate a large majority of the key ordering rules that exist in the literature. This software will be done with an object-oriented programming language such as C++ or JAVA.

Key words- Rules of scheduling, Jobshop, Production System, Simulation, Programming Languages.

I. INTRODUCTION

De nos jours, les entreprises doivent faire face à plusieurs changements en raison de la mondialisation et de la globalisation des marchés. Les environnements de plus en plus dynamiques, les courts cycles de vie des produits et l'abondance de l'offre compliquent la tâche aux entreprises, qui agissent désormais dans un contexte hautement concurrentiel. Ceci pousse les entreprises à chercher des méthodes pour tirer leur épingle du jeu en essayant d'offrir une vaste gamme d'avantages et de services aux consommateurs.

Un atelier flexible (Flexible Manufacturing System, *FMS*) consiste en l'agencement de plusieurs machines CNC

(Computer Numerical Control) desservies par un système de transport et de chargement automatisés des pièces et des outils, l'ensemble étant contrôlé et coordonné par un ordinateur central. Un tel système combine donc la flexibilité individuelle offerte par les machines CNC à une flexibilité accrue du système de manutention [1].

Les problèmes d'ordonnancement sont des problèmes souvent rencontrés dans l'industrie ou l'ingénierie. Ces problèmes se basent sur la détermination conjointe des dates d'exécution d'un ensemble d'opérations et des ressources mobilisées dans cette exécution. Ils se retrouvent dans toutes les formes d'organisation de la production et la qualité des solutions proposées conditionne largement la performance, voire la survie, de l'organisation. Les règles de priorité des files d'attente, généralement appelées « règles de priorité », sont parmi les méthodes les plus simples et les plus utilisées pour ordonnancer un atelier de type *job-shop*.

L'approche par simulation est de loin la plus répandue dans les logiciels industriels. Elle repose sur l'utilisation de règles de priorité sur les tâches, appelées aussi « règles d'ordonnancement » (par exemple : priorité aux tâches les plus courtes, aux tâches les plus en retard...). La littérature relative à ces règles est particulièrement riche [2].

Le potentiel de la simulation est très vaste, car elle est applicable à tous les flux de l'industrie et même des services, à tous les niveaux hiérarchiques et à toutes les phases du cycle de vie d'un système de production [3]. Le plus souvent, le logiciel de simulation est utilisé en phase de conception ou d'amélioration des ateliers pour dimensionner les capacités des stocks et files d'attente, pour tester des règles de fonctionnement, pour identifier les goulots d'étranglement, pour mesurer l'influence des perturbations, etc. La simulation de flux peut également être utilisée en phase d'exploitation, en complément d'outils de planification ou d'ordonnancement pour, par exemple, estimer des délais. Elle permet aussi de déterminer les valeurs optimales des paramètres de pilotage à appliquer au système de production avant de lancer l'exécution en couplant au modèle de simulation un algorithme d'optimisation [4].

Les logiciels d'ordonnancement génèrent souvent plusieurs propositions d'ordonnancement pour une situation donnée et fournissent une évaluation de ces propositions selon différents critères. Ceci permet une sélection manuelle du plan jugé globalement le meilleur. L'importance relative accordée aux différents critères doit alors tenir compte des

priorités (tactiques ou stratégiques) de la firme : veut-on mettre l'accent sur le respect des délais plutôt que sur la réduction des encours ? Peut-on renégocier l'une ou l'autre date de livraison ? ...etc.

L'idée ici est de développer un logiciel d'ordonnancement dans le quel on va intégrer la règle d'ordonnancement *FIFO* en utilisant une règle de routage qui sélectionne les routages dans un ordre croissant préétabli en analysant l'absence du routage préalablement choisi dans le système et simuler le passage des OFs à effectuer sur les différents postes de travail dans un atelier de type *job shop*. A chaque instant et sur chaque poste.

Cet outil logiciel BORLAND, basé sur le concept de programmation orientée objet, permet à un développeur, même non expérimenté, de créer assez facilement une interface homme/machine d'aspect « WINDOWS ».

Le programme n'est pas exécuté de façon séquentielle comme dans un environnement classique. Il s'agit de programmation « événementielle », des séquences de programme sont exécutées, suite à des actions de l'utilisateur (clique, touche enfoncée etc....), détectées par WINDOWS.

Cet article s'articule autour de trois sections. Dans la première partie on positionne le problème dans son contexte, en présentant une revue de littérature sur l'environnement du travail. Dans la deuxième de cette partie, on va présenter et justifier l'utilisation d'un outil informatique permettant de simuler le comportement dynamique d'un atelier de type *job shop*. Il s'agit de réaliser un logiciel que l'on peut simuler plusieurs scénarios et comparer les résultats avec le logiciel de simulation *ARENA* ; La dernière section est consacrée aux résultats trouvés et aux interprétations.

II. ETAT DE L'ART

Pour clarifier le concept de flexibilité, la majorité des recherches entreprises jusqu'à présent ont eu pour objectif de comprendre ce que la flexibilité pouvait apporter aux systèmes manufacturiers, de faire le tri entre ce qui est de la flexibilité et ce qui ne l'est pas. A partir de des observations sur le terrain et de l'étude de la littérature, ils se sont intéressés à l'analyse de la flexibilité des systèmes manufacturiers. Elle a été étudiée en trois grandes étapes :

- Définition du concept
- Décomposition du concept en dimensions structurées au sein d'une classification
- Conception d'outils de mesure de la flexibilité permettant de guider l'introduction de la flexibilité dans des entreprises ou d'analyser l'utilisation, par l'entreprise, de la flexibilité dont elle dispose.

Dans les années 1980, la production industrielle a subi une profonde mutation pour s'adapter aux évolutions des marchés. De nouveaux concepts sont apparus tels que le Juste-à-Temps (JIT), OPT, les ateliers flexibles (*FMS*), la logistique, etc. qui ont entraîné la disparition progressive des méthodes tayloriennes. Parallèlement, l'informatique a aussi connu une profonde évolution avec l'apparition des ordinateurs personnels, ce qui a permis un essor considérable des outils de simulation pour l'ingénierie. De

nombreux logiciels ont vu le jour et la plupart existent encore aujourd'hui, sous une version plus évoluée : Witness, Hocus, Genetik, Siman, Promodel, etc [5].

Les règles d'ordonnancement procurent en général des temps de réponse acceptables, il est difficile de choisir une règle en fonction de la satisfaction d'un ensemble d'objectifs donné. Celle-ci varie en effet en fonction du cas traité (nombre de machines, nombre de tâches dans les gammes, durées des tâches,...etc.). Plusieurs études ont été menées pour identifier la meilleure règle d'ordonnancement en fonction de la configuration de l'atelier et des conditions d'exploitation [6], [7], [8]. De telles études se sont, à notre connaissance, cantonnées à des problèmes d'ordonnancement simples et n'ont pas été généralisées dans un contexte industriel.

Dans [9] les auteurs ont démontré l'effet de variation des niveaux de flexibilité de routages (le nombre de routages alternatifs pris par un type de pièce) sur les performances d'un FMS jugées par le *makespan*, ils ont proposé dans [10] une stratégie d'ordonnancement prenant en considération la flexibilité de routages basée sur la logique floue dans un *FMS*.

De nos jours, les simulateurs utilisent une interface graphique sophistiquée permettant facilement de décrire les ressources d'un système productif, les produits faisant l'objet d'opérations de production et leurs gammes. L'interface graphique élimine la nécessité d'un apprentissage préalable d'un langage de programmation qui n'est nécessaire que dans la description d'actions sophistiquées liées à certains événements. La simulation demandée provoque la « projection d'un film » décrivant le fonctionnement dynamique du système (flux de production, circulation de ressources mobiles) ce qui lui confère un fort pouvoir descriptif du comportement d'un système productif piloté par certaines règles et soumis à une certaine demande [11].

On peut citer les travaux de [12] qui ont étudié par simulation l'impact de la flexibilité des machines et de routages en variant différentes conditions de l'atelier sur le temps de cycle moyen des pièces, on trouve aussi les travaux de [13] qui ont étudié l'effet des règles de priorité et des niveaux de flexibilité de routage sur les différentes performances d'un *FMS*.

III. PRESENTATION DU MODELE FMS ETUDIE

Nous allons étudier et développer un simulateur pour un modèle d'atelier *Job Shop* développé par C. Saygin en 2001 dans le quel on va intégrer et comparer ses performances avec le logiciel de simulation *Arena* [14].

La disposition de ces machines dans le système sous l'interface graphique du logiciel C++ est illustrée dans la figure suivante :

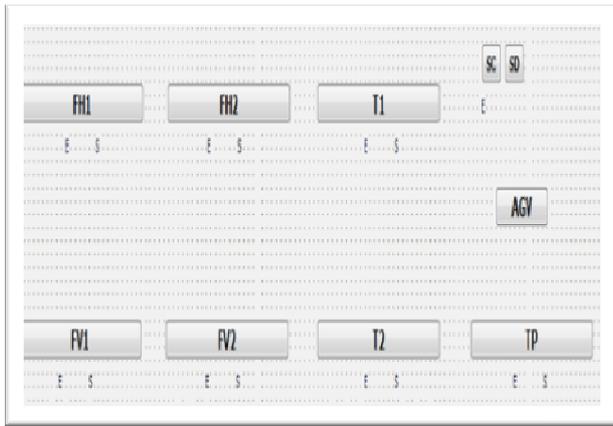


Fig.1 Configuration du modèle FMS étudié

A. Configuration physique du modèle

Le système modélisé comprend :

- Deux fraiseuses verticales (FV1 et FV2) ;
- Deux fraiseuses horizontales (FH1 et FH2) ;
- Deux tours (T1 et T2) ;
- Une toupie (TP) ;
- Une station de chargement (SC) ;
- Une station de déchargement (SD).

Chaque machine comporte une file d'attente d'entrée et une file d'attente de sortie, la station de chargement contient aussi une file d'attente d'entrée.

B. Hypothèses de fonctionnement

Les opérations sur le modèle étudié sont basées sur les suppositions suivantes :

- Chaque machine comporte une file d'attente d'entrée et une file d'attente de sortie, la station de chargement contient aussi une file d'attente d'entrée d'une capacité fixe $Q_{num}=2$.
- Les routages alternatifs de chaque type de pièce sont connus avant le début de la production.
- Le temps de traitement est déterminé et il comprend le temps de changement des outils et le temps d'exécution de la machine.

Chaque machine ne peut traiter qu'une seule pièce à la fois.

TABLEAU I

Les Routages alternatifs et temps de traitement des pièces [15].

Type de pièces et taux d'arrivée	Routages & temps de traitement (min)
A 17%	SC – T1 (30) – FV1 (20) - SD
	SC – T1 (30) – FV2 (20) – SD
	SC – T2 (30) – FV1 (20) – SD
B 17%	SC – T2 (30) – FV2 (20) - SD
	SC – T1 (20) – TP (1) – FV1 (15) – SD
	SC – T1 (20) – TP (1) – FV2 (15) – SD
C 17%	SC – T2 (20) – TP (1) – FV1 (15) – SD
	SC – T2 (20) – TP (1) – FV2 (15) – SD
	SC – T1 (40) – FV1 (25) - SD
D 21%	SC – T1 (40) – FV2 (25) – SD
	SC – T2 (40) – FV1 (25) – SD
	SC – T2 (40) – FV2 (25) – SD

D 21%	SC – T1 (40) – TP (1) – T1 (20) – FH1 (35) – SD
	SC – T1 (40) – TP (1) – T1 (20) – FH2 (35) – SD
	SC – T1 (40) – TP (1) – T2 (20) – FH1 (35) – SD
	SC – T1 (40) – TP (1) – T2 (20) – FH2 (35) – SD
	SC – T2 (40) – TP (1) – T1 (20) – FH1 (35) – SD
	SC – T2 (40) – TP (1) – T1 (20) – FH2 (35) – SD
	SC – T2 (40) – TP (1) – T2 (20) – FH1 (35) – SD
	SC – T2 (40) – TP (1) – T2 (20) – FH2 (35) – SD
	SC – T1 (25) – TP (1) – T1 (35) – FH1 (50) – SD
	SC – T1 (25) – TP (1) – T1 (35) – FH2 (50) – SD
E 20%	SC – T1 (25) – TP (1) – T2 (35) – FH1 (50) – SD
	SC – T1 (25) – TP (1) – T2 (35) – FH2 (50) – SD
	SC – T2 (25) – TP (1) – T1 (35) – FH1 (50) – SD
	SC – T2 (25) – TP (1) – T1 (35) – FH2 (50) – SD
	SC – T2 (25) – TP (1) – T2 (35) – FH1 (50) – SD
	SC – T2 (25) – TP (1) – T2 (35) – FH2 (50) – SD
F 8%	SC – FH1 (40) – SD
	SC – FH2 (40) – SD

III. ANALYSE ET INTERPRETATION DES RESULTATS

Ce papier a pour but de présenter une synthèse des résultats obtenus sur le modèle d'atelier *Job shop* défini précédemment. Pour ce faire, une simulation simultanée du système en question à l'aide du simulateur conçue par nos soins et du logiciel de simulation *Arena*, a été nécessaire afin d'aboutir à une approche comparative de la fiabilité du simulateur.

Nous allons étudier le comportement du système selon Le taux de création des pièces pour 5000 et 10000 pièces en fixant préalablement la taille des files d'attente d'entrée et de sortie des stations au niveau 2 (cas de système saturé), avec la règle de routage définie précédemment, et la règle classique de priorité *FIFO*, en se basant sur les critères de performances suivants :

A. Le temps de cycle

Le paragraphe suivant permet de mettre en évidence les résultats relatifs au temps de cycle.

TABLEAU II

Le taux de production pour une file d'attente égale à 2 :

Taux de création	1/25	1/20	1/15	1/10	1/5
<i>5000 Pièces</i>					
Arena	166.8	199.32	200.8	198.36	200.8
	6		1		9
Simulateur	122.6	109.59	126.4	201.22	376.5
	3		2		7
<i>10000 Pièces</i>					
Arena	170.7	198.24	201.7	196.88	198.0
	8		7		6
Simulateur	104.4	123.64	134.1	201.62	364.3
	1		4		1

La lecture des deux figures permet de dévoiler que le temps de cycle selon le simulateur est plus long en cas de saturation de système (5 et 10 minutes), tandis qu'un résultat plus favorable est constaté dans les autres cas.

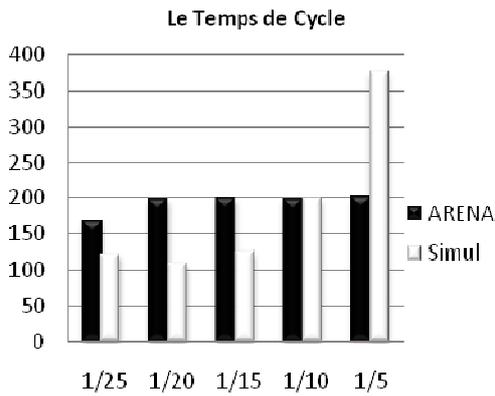


Fig.1 Temps de cycle pour 5000 pièces

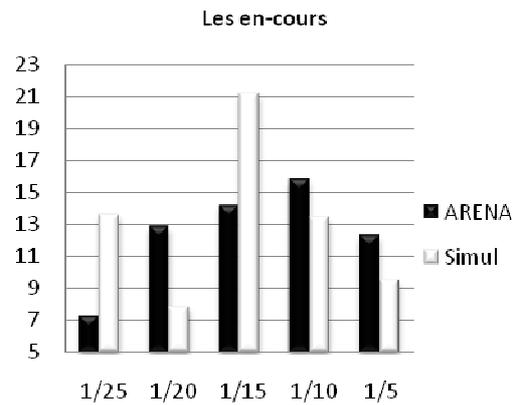


Fig.3 Les en-cours pour 5000 pièces

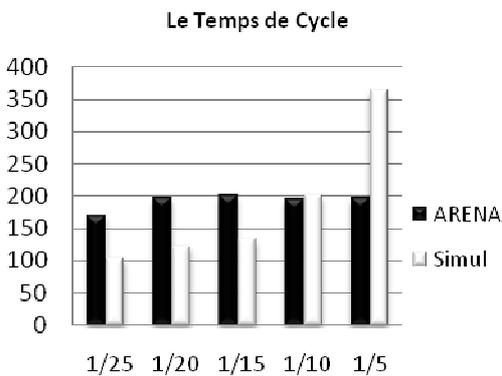


Fig.2 Temps de cycle pour 10000 pièces

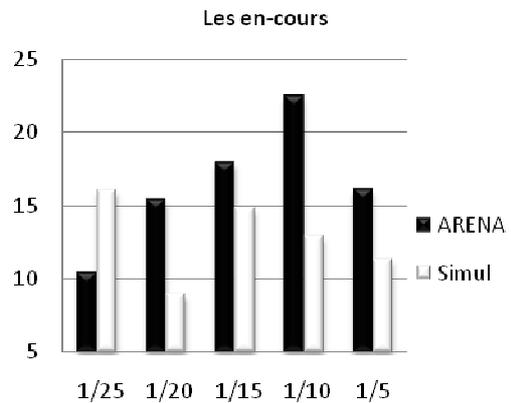


Fig.4 Les en-cours pour 10000 pièces

B. Les en-cours

Dans cette partie nous allons aborder les résultats relatifs aux en-cours.

TABLEAU III

Les en-cours pour une file d'attente égale à 2 :

Taux de création	1/25	1/20	1/15	1/10	1/5
<i>5000 Pièces</i>					
Arena	7.33	12.90	14.21	15.86	12.35
Simulateur	13.66	7.88	21.24	13.46	9.54
<i>10000 Pièces</i>					
Arena	10.47	15.37	17.91	22.44	16.10
Simulateur	16.04	8.95	14.8	12.87	11.35

Dans la majorité des cas nous distinguons un niveau inférieur des pièces en attente de transformation dans le processus de production.

C. Le taux de production

Le taux de sortie des pièces est calculé en divisant le nombre de pièces sorties du système sur le nombre de pièces créées, afin de faire une normalisation.

TABLEAU IV

Le taux de production pour une file d'attente égale à 2 :

Taux de création	1/25	1/20	1/15	1/10	1/5
<i>5000 Pièces</i>					
Arena	98.85	86.18	69.59	60.00	52.17
Simulateur	83.70	90.63	71.88	75.12	69.02
<i>10000 Pièces</i>					
Arena	99.27	85.85	69.51	59.46	52.18
Simulateur	82.98	88.77	79.24	75.69	64.20

Un taux de production appréciable est observé dans les quatre paliers (5,10 ,15 et 20 min). Le palier 25 minutes présente un faible taux de production pour les deux variantes (5000 et 10000).

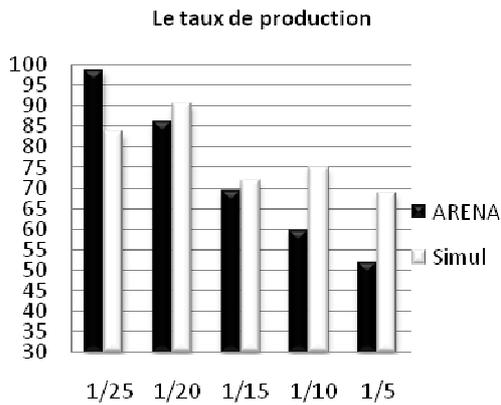


Fig.5 Le taux de production pour 5000 pièces

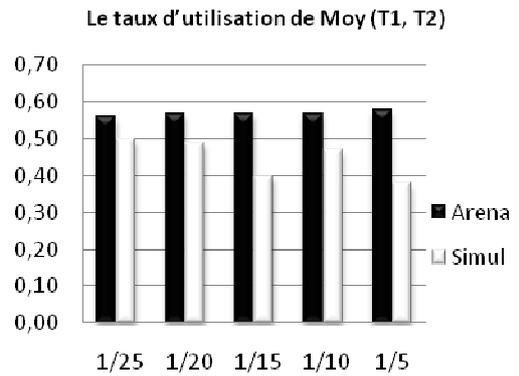


Fig.7 Le taux d'utilisation de Moy (T1, T2) pour 5000 pièces

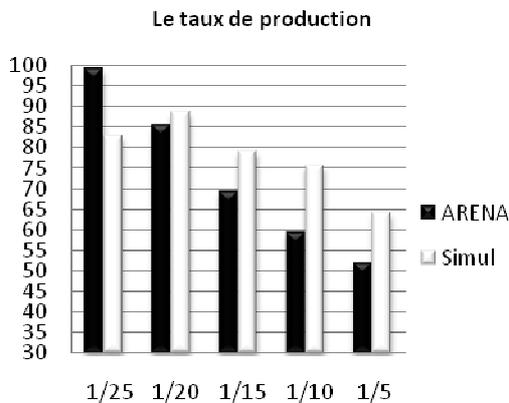


Fig.6 Le taux de production pour 10000 pièces

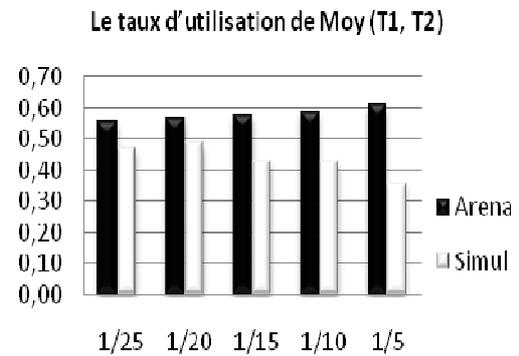


Fig.8 Le taux d'utilisation de Moy (T1, T2) pour 10000 pièces

D. Taux d'utilisation de Moy (T1, T2)

Dans le but d'explorer l'influence de taux d'utilisation des machines, nous avons choisi d'étudier les résultats des machines les plus sollicitées (T1, T2).

TABLEAU IV

Le taux d'utilisation des machines T1 et T2 pour une file d'attente égale à 2 :

Taux de création	1/25	1/20	1/15	1/10	1/5
<i>5000 Pièces</i>					
Arena	0.56	0.57	0.57	0.57	0.58
Simulateur	0.50	0.49	0.40	0.47	0.38
<i>10000 Pièces</i>					
Arena	0.56	0.57	0.58	0.59	0.61
Simulateur	0.47	0.49	0.43	0.43	0.36

Le taux d'utilisation des machines les plus sollicitées selon le *simulateur* est inférieur à celui obtenu par *ARENA* ce résultat montre le degré faible de compétitivité du simulateur conçu.

IV. CONCLUSION

L'analyse des différents résultats a permis de mettre en relief l'importance de la conception du simulateur et sa fiabilité selon les différentes normes de performances. Cette analyse nous a permis également de déterminer les limites du simulateur.

Dans ce travail, nous avons développé une application de simulation (un *simulateur*) afin d'aboutir à une approche comparative avec le logiciel *ARENA*.

Nous avons examiné l'impact d'une comparaison de ces deux logiciels en se basant sur les quatre critères de performance afin d'évaluer l'approche implémentée, et de valider l'application qui en découle.

La simulation de part son approche de modélisation a pour objectif l'expérimentation du comportement dynamique de système à l'aide de modèles informatiques.

La *POO* nous offre une réponse complète quant à la simulation et à l'optimisation. Elle nous permet de disposer d'un puissant outil d'aide à la décision induisant à l'étude des différents scénarios. L'objectif étant d'obtenir un modèle de simulation analogue à celui d'*Arena*

Ce travail nous a permis de constater que les résultats obtenus sont accompagnés par des améliorations sensibles pour certains critères.

Nous avons constaté également, que l'application du *simulateur* quant à la résolution du problème de l'ordonnancement de *Job Shop* donne des résultats qui ne peuvent induire obligatoirement à une conclusion de supériorité.

REFERENCES

- [1] T.Parker & T. Lettes, (1991). "Accounting standing in the way of flexible computer-integrated manufacturing". Management Accounting (January): 34-38.
- [2] Sabuncuoglu, (1998). "A study of scheduling rules of flexible manufacturing systems : a simulation approach". International Journal of Production Research, 36 (2), pp. 527-546.
- [3] C.Berchet, (2000). "Modélisation pour la simulation d'un système d'aide au pilotage industriel", 2000, Thèse, Institut National Polytechnique de Grenoble..
- [4] F.Fontanili, (1999). "Intégration d'outils de simulation et d'optimisation pour le pilotage d'une ligne d'assemblage multi produit à transfert asynchrone " Thèse de doctorat, Université Paris XIII, 1999.
- [5] S.Mirdamadi (2009). "Modélisation du processus de pilotage d'un atelier en temps réel à l'aide de la simulation en ligne couplée à l'exécution" Thèse de doctorat, Université de Toulouse,2009.
- [6] Panwalke & Iskander. (1977) "A survey of scheduling rules», Operations Research" vol. 25, p. 45-61.
- [7] B.Grabot & L.Geneste. (1994). "Dispatching rules in scheduling: a fuzzy approach". International Journal of Production Research, 32 (4), pp. 903-915.
- [8] H.Pierreval & N.Mebarki, (1997). "Dynamic selection of dispatching rules for manufacturing system scheduling". International Journal of Production Research, vol. 35, n°6, pp 1575-1591.
- [9] R. Caprihan & S. Wadhwa, "Impact of routing flexibility on the performance of an FMS-A simulation study", International Journal of Flexible Manufacturing Systems 9(3), 1997, pp.273-298.
- [10] R. Caprihan, & A. Kumar, K.E. Stecke, "A fuzzy dispatching strategy for due date scheduling of FMSs with information delays", International Journal of Flexible Manufacturing Systems 18(1), 2006, pp. 29-53.
- [11] M.Blaha & J.Rumbaugh (2005), Modélisation et Conception Orientées Objet avec UML 2, Paris, Pearson Education, 2005.
- [12] H. Tsubone & M. Horikawa, "Comparison between Machine flexibility and routing Flexibility", The international journal of flexible manufacturing systems, 11, 1999, pp. 83-101.
- [13] F. Mahmoodi & C.T. Mosier, "The effects of scheduling rules and routing flexibility on the performance of a random flexible manufacturing system", International journal of flexible manufacturing systems, 1999, pp. 271-289.
- [14] Saygin, C., F. F. Chen and J. Singh, 2001. " Real-Time Manipulation of Alternative Routeings in Flexible Manufacturing Systems : A Simulation Study ", International Journal of Advanced Technology. 19, p. 755-763.
- [15] C. Saygin & S.E. Kilic, (1995). "Dissimilarity Maximization Method for Real-time Routing of Parts in Random Flexible Manufacturing Systems", The International Journal of Flexible Manufacturing Systems, 16, 1995, pp. 169-182.