

ELICITATION DE CONNAISSANCES DE PILOTAGE TEMPS REEL VIA SIMULATION INTERACTIVE VISUELLE

A.L. Huyet, H. Pierreval

LIMOS-IFMA

Campus des Cézeaux – BP 265

63170 Aubière cedex

Anne-Lise.Huyet@ifma.fr, Henri.Pierreval@ifma.fr

RESUME : *La gestion d'un atelier et particulièrement le pilotage de la production est d'une importance cruciale pour une entreprise, mais cela se révèle bien souvent un exercice délicat. Afin d'aider à la prise de décision dans ce contexte, les décideurs ont besoin de bien comprendre le comportement des systèmes dans certaines situations. Ceci suppose d'avoir des connaissances sur le fonctionnement dynamique, les interactions entre différents phénomènes, ... Parfois des experts ont ces connaissances mais, si elles existent, elles ne sont pas toujours évidentes à formaliser et à transmettre à d'autres personnes. L'usage de la simulation interactive visuelle peut s'avérer une approche intéressante dans ce contexte dans la mesure où elle va permettre de prendre en compte une modification du modèle à tout moment. Dans cet article, nous proposons une méthodologie basée sur des simulations visuelles interactives auxquelles on connecte des systèmes d'apprentissage automatique. L'idée est ici de récolter et d'exploiter les données d'une simulation interactive afin d'éliciter les connaissances sur le pilotage du système étudié. La simulation interactive visuelle va permettre au responsable de l'atelier de s'entraîner au pilotage dans différents contextes et le système d'apprentissage en déduira alors des règles d'utilisation du système par rapport au contexte qui peuvent être utilisées par la suite avec profit. Un exemple basé sur un Flow Shop simplifié sera présenté afin d'illustrer l'intérêt de cette méthode pour l'aide à la décision sur le choix des règles de priorité.*

MOTS-CLES : *Simulation Interactive Visuelle, Apprentissage automatique, Pilotage des Systèmes de Production.*

1 INTRODUCTION

Le pilotage des systèmes complexes, comme les ateliers de production, les hôpitaux, ports de commerce, etc., nécessite souvent que des décisions soient prises en temps réel. Ces décisions concernent l'affectation de ressources (outils, machines, personnel), ou des choix concernant les tâches à réaliser. Elles sont connues pour être particulièrement difficiles à prendre car l'évolution dynamique du système est difficile à prévoir, les conséquences d'aléas sont difficiles à évaluer et des problèmes combinatoires importants peuvent en découler. Pour des systèmes complexes, tels les ateliers de production, de nombreux chercheurs ont travaillé sur la prise en compte de ces décisions dans le cadre de systèmes d'aide au pilotage. Plusieurs approches ont été proposées. Certaines d'entre elles, notamment celles qui reposent sur des principes d'intelligence artificielle (systèmes « experts », systèmes multi-agents, systèmes immunitaires artificiels), tentent d'intégrer des connaissances sur la manière dont des pilotes du système décident en temps réel. Malheureusement, si des méthodes d'aide à l'acquisition de connaissances existent (comme KADS (Schreiber *et al.*, 1993) par exemple), ces méthodes s'avèrent extrêmement limitées lorsqu'il s'agit d'éliciter des connaissances de pilotage. En effet, la

très grande variété des situations et la complexité du système et des problèmes rencontrés constituent des obstacles majeurs aux méthodes d'aide à l'acquisition de connaissances.

L'approche que nous présentons dans la suite consiste à mettre l'expert (le pilote) dans des situations de pilotage variées, à observer sa façon de décider et à formaliser les connaissances mises en œuvre sous une forme exploitable informatiquement.

Pour que les situations de pilotage auxquelles va être confronté le décideur puissent être suffisamment variées et reproductibles (pour observer les conséquences de ses décisions et les améliorer si nécessaire), l'approche proposée suggère d'utiliser des simulations interactives visuelles (SIV, traduction de l'anglais « visual interactive simulation » : VIS). Nous proposons alors l'usage de méthodes d'apprentissage automatique, capables de recueillir des connaissances sur la logique des décisions prises lors des simulations et sur les seuils numériques qui les déclenchent. L'usage de SIV a déjà fait l'objet de publications diverses (Neau, 2004) et l'apprentissage a déjà été utilisé en conjonction avec des simulations classiques (Pierreval et Ralambondrainy, 1990 ; Pierreval, 1994 ; Huyet et Paris, 2004 ; Huyet, 2006). Cependant, l'approche proposée, mettant le décideur en situation et modélisant ses réactions, n'a pas – à notre connaissance – fait l'objet de publications.

La présentation de notre approche est organisée comme suit. D'abord, nous rappelons les principes des SIV et leur usage dans notre contexte. Nous nous intéressons ensuite, de façon globale, aux méthodes d'apprentissage automatique et notamment aux algorithmes reconnus comme C4.5. Nous indiquons comment ces deux approches peuvent collaborer pour produire un arbre de décision caractérisant le comportement de l'expert en situation. Pour illustrer le potentiel de cette approche, nous l'appliquons à un système de production de type « flow shop » simplifié, qui a déjà fait l'objet de plusieurs publications. Nos conclusions et les perspectives de recherches qu'offre ce travail sont finalement présentées.

2 APPROCHE PROPOSÉE

2.1 Simulation Interactive Visuelle

Le terme de « Visual Interactive Simulation » semble avoir été utilisé pour la première fois par Hurrion en 1978 au cours d'un travail utilisant la Simulation Visuelle Interactive (SIV) pour modéliser des problèmes de planning de production dans les ateliers. La SIV est l'utilisation d'un modèle de simulation où l'utilisateur peut suspendre l'exécution du modèle, modifier un ou plusieurs paramètres, y compris la structure du modèle, puis reprendre son exécution (Hurrion, 1991). Il est alors possible de voir l'impact de ces modifications et d'en tenir compte dans les prochaines prises de décision. Outre les aspects graphiques de représentation nécessaires à la SIV, l'intérêt essentiel de ce type d'approche repose sur sa capacité à interagir durant le fonctionnement du modèle (Robinson, 1994). Il ne s'agit pas seulement de simulations animées. Les objectifs vont alors de supporter la décision (Bell *et al.*, 1990 ; Chau, 1993 ; Ceric, 1997), à l'aide à la compréhension des systèmes (Bell, 1989) en passant par l'outil de communication (Porter, 1991). Pour atteindre ces différents objectifs, la SIV est parfois couplée à d'autres approches sur le modèle de ce que l'on trouve avec la simulation. En 1990, Hurrion utilise un réseau de neurones afin de représenter les résultats d'un modèle de simulation. Le réseau de neurones est implanté comme une interaction dans le modèle de simulation interactive visuelle. Tous les résultats obtenus de la simulation sont transmis au réseau de neurones. Après une période convenable d'entraînement, la qualité des résultats obtenus en provenance du réseau de neurones correspond à celle obtenue en lançant le modèle de simulation original. A l'aide d'un exemple, il montre que la combinaison d'un tel système peut augmenter la qualité de prise de décision d'un modèle de type SIV et surtout, qu'il crée ainsi un métamodèle de simulation performant. Déjà en 1991, ce même auteur présente un autre couplage du SIV avec, cette fois-ci, des systèmes experts (Hurrion, 1991). Mettant l'accent sur le fait que le modélisation interactive visuelle est une technique passive qui est généralement utilisée par des personnes

expérimentées, il propose d'inclure des connaissances expertes au modèle interactif de simulation (VIM). La nature intrinsèque du VIM permet à l'utilisateur d'initier une interaction avec le modèle mais ces interactions entre le modèle et l'utilisateur restent de forme passive et sont seulement des conditions possibles préprogrammées. Dans cet article, une description est faite sur des méthodes permettant d'ajouter de l'expertise au VIM et les faire ainsi devenir plus participatifs avec l'utilisateur. En conclusion, l'auteur propose une approche pour l'inclusion de composants intelligents au VIM, permettant ainsi de changer la passivité de la méthode et faciliter la recherche de solutions originales.

Par ailleurs, un autre couplage de SIV avec l'intelligence artificielle est développé dans une étude de cas sur le design de systèmes industriels flexibles (Vujosevic, 1994). L'application de techniques d'intelligence artificielle permet, dans ce cas, d'implanter un ensemble de connaissances sources.

Néanmoins, certains auteurs ont mentionné la difficulté qui résulte de l'aspect aléatoire de la SIV (notamment Bell, 1989). Il en ressort l'importance de l'utilisation d'une SIV pour la modélisation du comportement transitoire d'un système ainsi que la prise en compte de ces aspects dès la modélisation notamment pour s'assurer que tous les indicateurs pertinents pour la simulation apparaissent bien dans la partie interactive avec l'utilisateur.

2.2 Apprentissage automatique

Pour notre méthodologie, il est nécessaire de construire un système d'apprentissage qui va nous permettre d'induire des décisions menant à un pilotage performant du système. Ceci revient à un processus de classification dont le but sera alors d'identifier la classe d'une solution (ici, relative à un choix de règles de priorité) à partir de certains caractères (ici, des états représentatifs du système tel que le nombre d'entités dans les files d'attente) et de hiérarchiser les paramètres en fonction de leur criticité sur la performance du système. Il existe de nombreuses méthodes d'ECD ou d'apprentissage (Michalski *et al.*, 1984). Parmi elles, nous pouvons citer les réseaux de neurones (Bishop, 1995), (Mitchell, 1997) ou les Genetic Based Machine Learning (Mitchell, 1997), (Bonnarini, 1997) qui sont utilisés dans diverses approches mais qui restent des boîtes noires, ce qui est rédhibitoire avec les objectifs pédagogiques et explicatifs de notre méthodologie. L'une des méthodes où ce problème n'apparaît pas est celle des graphes d'induction (Zighed and Rakotomalala, 2000). Leurs procédures de construction sont compréhensibles par les différentes personnes concernées par le problème et leur représentation graphique apporte un atout évident dans les principales approches de construction de graphes d'induction, nous pouvons citer CART (Breiman *et al.*, 1984), ID3 (Quinlan, 1986) ou C4.5 (Quinlan, 1993). Pour implémenter notre module

d'apprentissage, nous avons choisi l'algorithme C4.5 développé par Quinlan (1993), l'efficacité de celui-ci étant reconnue. Le but de cette approche est, à partir d'un ensemble de données, d'organiser les paramètres sous forme d'arbre dans lequel plus le paramètre est pertinent, plus il est proche de la racine de l'arbre. La pertinence d'un paramètre, dans ce cas, dépend de sa capacité à séparer les données de l'ensemble initial en sous-ensembles ne contenant que des données de même classe. Le paramètre générant plusieurs sous-ensembles à partir d'un ensemble est appelé « séparateur ». L'ensemble initial est séparé en sous-ensembles et ainsi de suite jusqu'à ce qu'ils ne contiennent plus que des données appartenant à la même classe ou que les données d'autres classes soient en nombre insuffisant pour être représentatives. La structure étant celle d'un arbre, le vocabulaire en est inspiré, ainsi, nous parlons de la racine pour le paramètre le plus important et de feuilles pour les ensembles purs. Le tableau 1 et la figure 1 présentent un exemple de données associées à un arbre de décision.

N°	Ensoleillement	Température (°F)	Humidité (%)	Vent	Jouer
1	Soleil	75	70	Oui	Oui
2	Soleil	80	90	Oui	Non
3	Soleil	85	85	Non	Non
4	Soleil	72	95	Non	Non
5	Soleil	69	70	Non	Oui
6	Couvert	72	90	Oui	Oui
7	Couvert	83	78	Non	Oui
8	Couvert	64	65	Oui	Oui
9	Couvert	81	75	Non	Oui
10	Pluie	71	80	Oui	Non
11	Pluie	65	70	Oui	Non
12	Pluie	75	80	Non	Oui
13	Pluie	68	80	Non	Oui
14	Pluie	70	96	Non	Oui

Tableau 1 : exemple de données sur lesquelles on souhaite apprendre

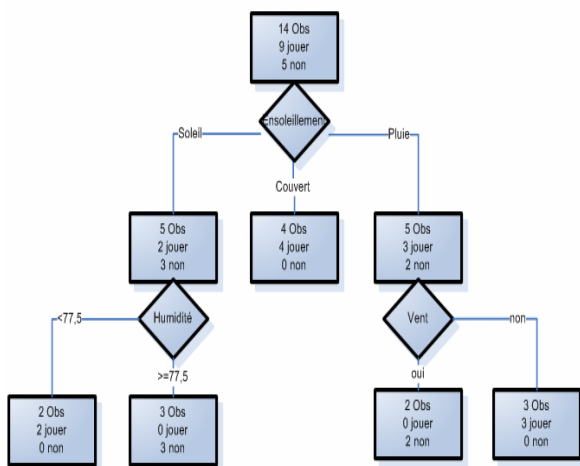


Figure 1 : Arbre de décision construit à partir des données présentées dans le tableau 1

Néanmoins, cette structure peut devenir inexploitable car illisible lorsque de trop nombreux paramètres interviennent. Ainsi, lorsque la structure d'arbre devient trop complexe pour présenter au mieux les résultats, il est possible d'extraire, à partir de l'arbre construit, des règles représentant chaque chemin de la racine à chacune des feuilles. La figure 2 présente les règles extraites à partir de l'arbre de la figure 1. Ce sont des règles obtenues à partir d'arbre qui seront présentées dans l'application de cet article.

S'il fait soleil et que l'humidité est < 77.5%, on joue
S'il fait soleil et que l'humidité est >= 77.5%, on ne joue pas.
Si le temps est couvert, on joue
S'il pleut et qu'il y a du vent, on ne joue pas.
S'il pleut et qu'il n'y a pas de vent, on joue.

Figure 2 : Règles de décision extraites à partir de l'arbre de la figure 1

Différentes approches d'apprentissage automatique ont déjà été utilisées pour analyser des résultats de simulation (Pierreval et Ralambondrainy, 1990 ; Pierreval, 1994 ; Huyet et Paris, 2004 ; Huyet, 2006). Cependant, dans ces travaux, on cherchait à apprendre comment se comporte le modèle et non pas, comment se comporte un utilisateur du modèle en fonction de l'état de celui-ci. L'approche proposée dans cet article traite l'élicitation de connaissances en pilotage ce qui implique une notion de temps réel absente dans les travaux déjà existants qui combinent apprentissage et simulation. C'est l'utilisation de SIV qui rend cette prise en compte du temps réel possible.

2.3 Méthodologie proposée : élicitation de connaissances par apprentissage automatique à partir de simulation visuelle interactive

L'objectif principal est ici d'éliciter les connaissances tacites d'un expert/utilisateur du système, c'est-à-dire les rendre explicites. Ces connaissances peuvent ensuite être transmises à d'autres utilisateurs moins expérimentés et leur permet, à leur tour, de piloter le système efficacement. Cette formalisation automatisée des connaissances empiriques d'un expert permettra également une analyse des résultats obtenus pouvant être suivie d'une amélioration des pratiques existantes. Le principe repose sur une acquisition temps réel des actions de l'expert en fonction du contexte. Pour cela, tout au long de la SIV, on garde une trace des décisions prises et de l'état du système, au travers des variables qui le caractérise, à ce moment. On réalise ainsi un mouchard qui mémorise les actions et les modifications réalisées par l'expert ainsi que les indicateurs représentant l'état du système au même moment. Le comportement de l'expert est donc espionné et mémorisé. Ces données sont ensuite exploitées par apprentissage automatique. Cette approche permet ainsi d'analyser et de comprendre non plus un modèle comme c'est le cas dans des approches existantes de couplage apprentissage-simulation mais l'utilisateur de ce modèle ainsi que ces décisions en fonction de l'état

que ces décisions en fonction de l'état du système. Dans cette application, nous avons choisi d'utiliser les arbres et règles de décision dans un souci de facilité de communication et d'interprétation. La figure 3 présente le principe de l'approche proposée.

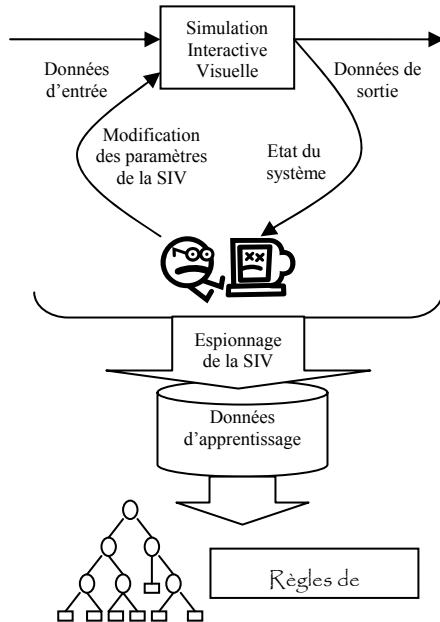


Figure 3 : Principe de l'approche proposée

Les résultats obtenus en termes d'arbre ou de règles d'induction vont donc rendre explicites les connaissances tacites d'un expert. On obtiendra ainsi des résultats du type : selon le niveau de tel indicateur dans le système, il faudra modifier telle entrée de telle façon. A partir de cela, un autre utilisateur non ou moins expert du système aura la possibilité, en suivant les règles obtenues à partir de la SIV de l'expert, de piloter le système d'une façon efficace à son tour. Par ailleurs, l'analyse des résultats obtenus peut également amener à repenser certains aspects du pilotage qui, même s'ils sont réalisés par un expert, peuvent se révéler peu pertinents dans la globalité du process. La représentation explicite des connaissances peut mettre en valeur certaines incohérences ou inefficacités dans la méthode de l'expert.

3 APPLICATION

3.1 Description du système étudié

Le modèle étudié ici est le flow shop simplifié de Barrett et Barman (1986). Ce modèle, largement utilisé dans la littérature, est constitué de 2 postes d'usinage (workcenter 1 ou WC1 et workcenter 2 ou WC2) réalisant l'un après l'autre l'usinage des pièces. Chaque poste est composé de deux machines identiques. Sur un

poste, si les deux machines sont occupées, les pièces attendent dans une file d'attente unique, propre à ce poste d'usinage.

Au niveau du workcenter 2, 10% des pièces usinées doivent subir une retouche sur l'une des deux machines du poste. Ces pièces à retoucher sont prioritaires par rapport aux pièces en attente d'usinage.

L'arrivée des pièces est définie par une loi exponentielle et les temps de process sur les machines sont distribués exponentiellement également.

Enfin, les machines sont mises hors service toutes les 50 pièces pour permettre le changement des outils de coupe.

Les dates de livraison des pièces sont définies lors de l'arrivée de celles-ci dans l'atelier par la formule :

$$D_i = A_i + \alpha \sum p_{ij}$$

Avec

- D_i , la date de livraison promise au client de la pièce I
- A_i , le temps d'arrivée dans l'atelier de la pièce
- α , le facteur de retard (ici nous prendrons 3)
- p_{ij} , le temps de process d'une entité i sur une machine j .

Pour chaque simulation, il est possible de choisir la charge de l'atelier. L'utilisateur a le choix entre une charge lourde (91%), moyenne (86%) et faible (81%).

Le temps de process des pièces peut également être modifié en changeant la dispersion des commandes. Deux options se présentent : forte ou faible. La déviation standard varie donc entre 0,3 et 0,6. Le temps de process est donc calculé de la manière suivante :

$$\text{Temps de process} = \text{Espérance du temps de process} + N(0, \sigma)$$

Avec N = valeur sélectionnée par la distribution de loi normale de moyenne 0 et de déviation standard de 0.3 ou 0.6 en fonction du choix de dispersion.

Enfin, les hypothèses sont les suivantes :

- Les temps de process de chaque pièce sont définis à l'arrivée des pièces et demeurent fixes durant la simulation.
- L'usinage d'une pièce ne peut être interrompu.
- La capacité d'usinage de chaque machine est d'une seule pièce.
- Pas de panne.
- Le temps de transport des pièces entre les workcenters vaut 0,3.
- L'atelier travaille 40 heures par semaine et 50 semaines par an.

La figure 4 présente le schéma de l'atelier flow shop étudié.

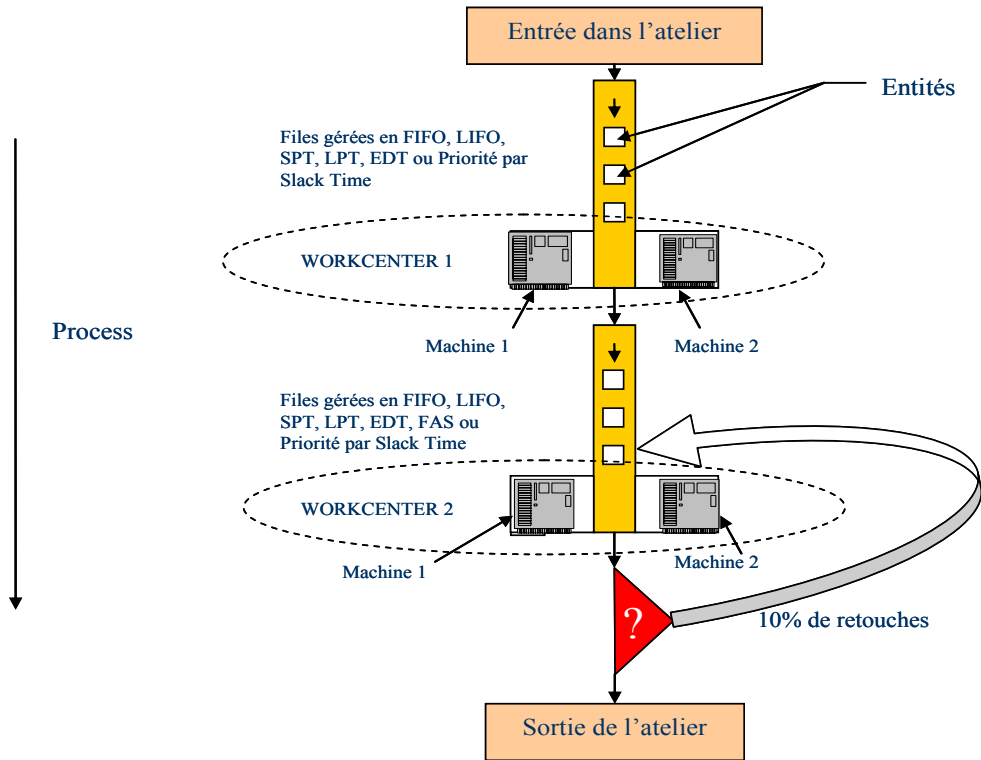


Figure 4 : Schéma du Flow Shop de Barrett et Barman (1986)

L'objectif de cette simulation est de minimiser la moyenne des retards obtenus pour les pièces usinées dans cet atelier, en choisissant dynamiquement des règles de priorité optimales pour les files d'attente. Les règles de priorité disponibles sont :

- FIFO,
- LIFO,
- SPT = Shortest Processing Time,
- LPT = Longest processing Time,
- EDT = Earliest Due Time,
- FAS = First Arrived at Shop (équivalent à FIFO en ce qui concerne la file 1),
- Slack Time = temps séparant la date de livraison de la fin de l'usinage.

Pour cela, l'expert dispose d'une fenêtre interactive lui permettant, à la fois, de suivre le déroulement de la simulation en terme d'accumulation des pièces dans les files d'attente, d'évolution des retards...et, d'agir sur le

choix des règles de priorité suivies par le workcenter 1 et par le workcenter 2. La figure 5 présente une capture d'écran de cette fenêtre.

Afin de permettre à l'expert de prendre les bonnes décisions au bon moment, le modèle de SIV doit lui permettre de connaître précisément les différents états du système identifiés comme intéressants pour la prise de décision :

- le nombre de pièces dans la file 1
- le nombre de pièces dans la file 2
- le nombre de pièces en retard dans la file 1
- le nombre de pièces en retard dans la file 2
- le nombre de pièces presque en retard dans la file 1
- le nombre de pièces presque en retard dans la file 2
- le nombre de pièces à l'heure dans la file 1
- le nombre de pièces à l'heure dans la file 2
- le retard moyen calculé

On retrouve une représentation graphique de ces indicateurs dans la fenêtre interactive en figure 5.



Figure 5 : Fenêtre permettant la Simulation Visuelle Interactive du Flow Shop étudié

Cette interface graphique est la troisième qui a été créée. C'est avec elle que les meilleurs résultats ont été obtenus par rapport à la prise de décision de l'utilisateur et au regard des indicateurs de performance. Elle permet de visualiser, en temps réel, le retard moyen des pièces (« Mean Tardiness ») en fonction des règles suivies dans les deux files d'attente (« WC1 Dispatching Rules » et « WC2 Dispatching Rules »).

3.2 Résultats

Dans cette application, 8000 heures de simulation interactive visuelle à l'état stationnaire du système (obtenu après 7000h de simulation) ont servi de données d'entrée, simulation interactive visuelle menée par un expert du système. Nous avons besoin de 8 000 heures de simulations après avoir atteint le régime permanent.

Dans leur étude statique, pour les mêmes réglages en termes de variation de temps de process et de charge de l'atelier, Barrett et Barman (1986) ont montré que la meilleure combinaison de règles de priorité pour minimiser le retard moyen était SPT-EDT, le retard moyen était alors de l'ordre de 8 heures.

Dans des conditions équivalentes, la SIV réalisée par notre expert génère un retard moyen de 3,35 heures. A partir du moucharid réalisé pour ce modèle, une base de 800 données a été constituée. Cette base est ensuite transmise en entrée de la méthode d'apprentissage choisie, l'algorithme du C4.5 de Quinlan (1993). Avec

des feuilles d'au moins 18 données et un taux de confiance de 0.5, on obtient un arbre de décision de 13 nœuds et 7 feuilles soit 7 règles d'induction présentées en figure 6. Ces résultats ont été obtenus sur Tanagra, logiciel libre d'apprentissage automatique développé par Rakotomalala (2005).

Rule 1

If $Q2_minimum_expected_processing_time \geq 0,3275$

And $number_in_Q2 \geq 12$

Then $WC1_dispatching_rule\ WC2_dispatching_rule = SPT\ SPT$

Rule 2

If $Q2_maximum_remaining_time < -1,4134$

And $number_in_Q2 \leq 11$

Then $WC1_dispatching_rule\ WC2_dispatching_rule = EDT\ SPT$

Rule 3

If $Q1_minimum_expected_processing_time < 0,1821$

And $Q1_maximum_expected_processing_time < 9,1965$

And $number_in_Q1 \geq 12$

And $Q2_maximum_remaining_time \geq -1,4134$

And $number_in_Q2 \leq 11$

Then $WC1_dispatching_rule\ WC2_dispatching_rule = LPT\ SPT$

Rule 4

If $Q1_minimum_expected_processing_time \geq 0,1821$

And $Q1_maximum_expected_processing_time < 9,1965$

And $number_in_Q1 \geq 12$

And $Q2_maximum_remaining_time \geq -1,4134$

And $number_in_Q2 \leq 11$

Then $WC1_dispatching_rule\ WC2_dispatching_rule = SPT\ SPT$

Rule 5

If $Q1_maximum_expected_processing_time \geq 9,1965$

And $number_in_Q1 \geq 12$

And $Q2_maximum_remaining_time \geq -1,4134$

```

And number_in_Q2 <= 11
Then WC1_dispatching_rule WC2_dispatching_rule = SPT SPT
# Rule 6
If number_in_Q1 <= 11
And Q2_maximum_remaining_time >= -1,4134
And number_in_Q2 <= 11
Then WC1_dispatching_rule WC2_dispatching_rule = SPT SPT
# Rule 7
If Q2_minimum_expected_processing_time < 0,3275
And number_in_Q2 >= 12
Then WC1_dispatching_rule WC2_dispatching_rule = SPT EDT

```

Figure 6 : Règles d'induction extraites de l'arbre obtenu

L'analyse de ces résultats montre que l'expert prend en compte le nombre de pièces dans la file Q2 (devant le workcenter 2) puis selon le temps restant avant d'être en retard ou les temps de process min et max ainsi que le nombre de pièces dans la file Q1, il change les règles de priorité. En détaillant par exemple la règle 3, on obtient le principe suivant : s'il y a moins de 12 pièces en attente en Q2 et que les pièces dans Q2 disposent d'encore plus de 1.4 heures avant leur date de livraison, on regarde le nombre de pièces dans la file Q1. Si ce nombre est d'au moins 12, on regarde le maximum des temps de process des pièces de Q1. S'il est inférieur à 9,2 heures, on regarde le minimum de ces temps de process et s'il est inférieur à 0.18 heures, on choisit LPT pour règle de priorité de WC1 et SPT pour WC2. Ce qui revient à faire passer rapidement une pièce de Q1 qui aurait un temps de process relativement court lorsqu'il y a beaucoup de pièces dans Q1 et pas trop dans Q2. On espère ainsi désengorger WC1.

Les résultats obtenus sur cette application simplifiée corroborent bien le raisonnement empirique mené ; mais, outre sa formalisation automatique, ils apportent une évaluation des seuils à prendre en compte ce qui n'est pas souvent explicite dans le raisonnement de l'expert.

3.3 Extension possible de ces expérimentations

Dans les expérimentations effectuées, les décisions ont été prises lorsque l'atelier fonctionne en état stable. Les connaissances ainsi obtenues concernent donc ce fonctionnement normal du système. Or, il serait possible de mettre l'utilisateur dans des situations prédéfinies et ainsi éliciter des connaissances ciblées par rapport à ces situations. En effet, la simulation aurait pu concerner un contexte spécifique comme l'occurrence d'une panne particulière, afin de susciter des décisions de type réaction aux perturbations, par exemple. En ce sens, la définition de plans d'expériences destinés à définir les situations dans lesquelles orienter l'utilisateur pour susciter des décisions est un aspect intéressant à associer à cette approche.

4 CONCLUSION ET PERSPECTIVES

L'exemple simple que nous avons présenté montre que l'approche proposée est capable de générer des connaissances sur le comportement du décideur mis en situation. De telles connaissances, qui concernent des réactions en temps réel, auraient été particulièrement difficiles à obtenir avec une autre approche.

La possibilité de refaire plusieurs simulations pour tester plusieurs manières de piloter permet à l'utilisateur du système d'améliorer sa façon de prendre des décisions. La SIV affichant les performances en termes de retard des pièces, l'utilisateur, comme dans un jeu vidéo, peut d'abord améliorer ses performances, afin que l'extraction des connaissances ne se fasse que lorsqu'il est satisfait de son « score » et qu'il trouve son comportement suffisamment représentatif et efficace.

En terme de perspectives de recherche, différents axes peuvent être dégagés. L'algorithme d'apprentissage utilisé peut être amélioré, notamment par la prise en compte de données floues dans l'expression des connaissances acquises par exemple.

S'agissant des connaissances élicitées, dans notre exemple, toutes les connaissances sont apprises via simulation. Or, si certaines ne sont pas explicitement connues par l'expert, celui-ci en possède d'autres qui pourraient utilement compléter les connaissances apprises via la SIV, ou pourraient permettre de les enrichir.

De plus, même si l'exemple traité est issu du domaine des systèmes de production, l'approche proposée est applicable pour des systèmes variés pour lesquels l'ergonomie doit être adéquate. En effet, l'interface de la SIV a ici fait l'objet d'un développement spécifique pour cette application. D'autres interfaces et d'autres interactions entre l'utilisateur et le modèle sont possibles comme l'usage d'autres périphériques d'entrée, selon la nature des décisions à tester et le domaine d'application. La méthodologie proposée est, quand à elle, directement transposable ; par ailleurs, nous envisageons la mise en œuvre de cette approche pour l'étude de l'aide au pilotage d'un système d'assemblage flexible réel.

REMERCIEMENTS

Nous adressons nos remerciements à J. Champenois, T. Dück, S. Marco, M. Maudet, C. Oerthel et A. Quéau qui nous ont considérablement aidé dans le développement et la mise au point de la SIV de l'application présentée.

REFERENCES

Barrett R.T. and Barman S., 1986, A SLAM II simulation study of a simplified flow shop, *Simulation*, 47(5), pp. 181-189.

- Bell P.C., 1989, Stochastic Visual Interactive Simulation Models, *Journal of the Operational Research Society*, 40(7), pp 615-624.
- Bell P.C., Arshad A.T. and Kirkpatrick P.F., 1990, Visual interactive simulation modeling in a decision support role, *Computers & Operations Research*, 17(5), pp 447-456.
- Bishop C., 1995, *Neural Networks for Pattern Recognition*, Oxford University Press.
- Bonnarini A., 1997, Evolutionary learning of fuzzy rules: competition and cooperation, *Fuzzy Modelling: Paradigms and Practice*. Kluwer Academic Publishers, Norwell, MA.
- Breiman L., Friedman J.H., Olshen R.A. and Stone C.J., 1984, *Classification and Regression Trees*. California: Wadsworth international.
- Ceric V., 1997, Visual interactive modeling and simulation as a decision support in railway transport logistic operations, *Mathematics and Computers in Simulation*, 44, pp 251-261.
- Chau P.Y.K., 1993, Decision support using traditional simulation and visual interactive simulation, *Information and Decision Technologies*, 19, pp 63-76.
- Hurriion R.D., 1978, An Investigation of Visual Interactive Simulation Methods Using the Job Shop Scheduling Problem, *Journal of the Operational Research Society*, 29(11), pp. 1085-1093.
- Hurriion R.D., 1991, Intelligent Visual Interactive Modelling, *European Journal of Operational Research*, 54, pp 349-356.
- Hurriion R.D., 1992, Using a Neural Network to enhance the decision making quality of a visual interactive simulation model, *Journal of the Operational Research Society*, 43(4), pp 333-341.
- Huyet A.-L., 2006: Optimization and analysis aid via data-mining for simulated production systems, *European Journal of Operational Research*, 173(3), pp. 827-838.
- Huyet A.-L. and Paris J.-L., 2004, Synergy between Evolutionary Optimization and Induction Graphs Learning for Simulated Manufacturing Systems, *International Journal of Production Research*, 42(20), pp.4295-4313.
- Michalski R.S., Carbonell J. and Mitchell T., 1984, *Machine Learning*, Tioga Press, Palto Alto., Calif., USA.
- Mitchell T., 1997, *Machine Learning*, Mc-Graw-Hill.
- Neau J., 2004, *Etude de la Simulation Interactive Visuelle dans la phase de croissance du cycle de vie*, Rapport de Master OSIL – Centrale Paris.
- Pierreval H., 1992, Rule-based simulation metamodels. *European Journal of Operational Research*, 61, pp. 6-17.
- Pierreval H. and Ralambondrainy H., 1990, A simulation and learning technique for generating knowledge about manufacturing systems behavior, *Journal of Operational Research Society*, 41(6), pp. 461-474.
- Porter K., 1991, Visual interactive simulation as a communication tool – A case study. *European Journal of Operational Research*, 54, pp 287-292.
- Quinlan J.R., 1986, Induction of decision trees. *Machine Learning*, 1, pp 81-106.
- Quinlan J.R., 1993, *C4.5: Programs for Machine Learning*, Morgan Kaufmann.
- Rakotomalala R., 2005, TANAGRA : un logiciel gratuit pour l'enseignement et la recherche, *Actes de EGC'2005*, RNTI-E-3, vol. 2, pp.697-702.
- Robinson S.L., 1994, An Introduction to Visual Interactive Simulation in Business. *International Journal of Information Management*, 14, pp 13-23.
- Schreiber G., Wielinga B. & Breuker J., 1993, *A principled approach to knowledge-based system development*, Academic Press.
- Vujosevic R., 1994, Visual interactive simulation and artificial intelligence in design of flexible manufacturing systems. *International Journal of Production Research*, 32(8), pp 1955-1971.
- Zighed D.A. and Rakotomalala R., 2000, *Graphes d'induction*, Hermes.