

## PRISE DE DECISION ET SIMULATION EN LIGNE : APPLICATION A UNE LIGNE D'ASSEMBLAGE

Serguei IASSINOVSKI\*, Khalid KOUISS\*\*, Christophe FAGNART\*\*\*

---

Résumé. - Cet article présente un projet commun entre la société SIMPARTNERS et l'AIP-PRIMECA Auvergne pour la réalisation d'un prototype de prise de décision et de simulation en ligne sur une ligne d'assemblage réelle. Cet article décrit l'objectif du projet ainsi que l'architecture et les principes qui seront mis en œuvre pour la réalisation du démonstrateur. En introduction, nous situons la prise de décision en ligne par rapport aux outils de gestions de l'entreprise (ERP, CRM, SCM, MES, . . .). Ensuite, nous présentons un bref état de l'art sur la simulation en ligne avant de présenter la ligne d'assemblage de l'AIP-PRIMECA et le modèle de simulation réalisé. Nous détaillons également la partie « prise de décision » et enfin, nous concluons et évoquons les perspectives pour ce projet.

Mots-clés : Simulation en ligne, Pilotage, Prise de décision, Atelier flexible.

### 1. Introduction

Au cours de ces dernières années, de nombreuses entreprises ont investi massivement dans des systèmes informatiques pour les aider à mieux gérer leurs opérations. Certaines applications sont aujourd'hui bien connues et largement répandues. On peut citer entre autre les solutions ERP (Enterprise Resource Planning), CRM (Customer Relationship Management) ou encore SCM (Supply Chain Management). Mais ces applications s'intéressent à des niveaux de

---

\* Facultés Universitaires Catholiques de Mons 151 chaussée de Binche B7000 Mons, Chargé de recherche, iassinov@fucam.ac.be.

\*\* LIMOS-IFMA Campus de Clermont-Ferrand/Les Cézeaux - BP 265 - 63175 Aubière Cedex. France, Maître de Conférences, khalid.kouiss@ifma.fr.

\*\*\* Simpartners SA 2, rue Descartes B7000 Mons, Administrateur délégué, fagnart@simpartners.com.

gestion supérieures. D'un point de vue opérationnel, en ce qui concerne la gestion d'atelier, les éditeurs commencent à développer des solutions MES (Manufacturing Execution System). Ces outils regroupent un ensemble de fonctions pour la plupart différentes de celles des systèmes déjà nommés (ERP, SCM, CRM) et les recouvrements de fonctions qui existent ne concernent généralement pas le même horizon temporel. Ces systèmes apportent des bénéfices spécifiques et sont focalisés sur les activités de production. Un système MES fournit l'information nécessaire à l'optimisation des systèmes de production du lancement de l'ordre de fabrication au produit fini. En utilisant des données précises et actualisées, un système MES permet de guider, d'initialiser, de répondre et de rapporter les événements traduisant l'activité de l'atelier aux moments où ils se produisent. La rapidité de la réponse à un changement de situation et la focalisation sur la réduction des activités non productives permettent d'améliorer l'efficacité des processus de production.

La prise de décisions en ligne est au cœur des fonctionnalités des solutions MES notamment pour l'ordonnancement, la gestion de la maintenance, la gestion des ressources, etc. Un outil efficace de prise de décisions doit employer différentes méthodes scientifiques (simulation, systèmes experts, optimisation) et doit reposer sur un modèle du système piloté. La société SIMPARTNERS SA mène actuellement un projet de développement d'un tel logiciel appelé System Dynamic Builder (SDBuilder®). C'est dans le cadre de ce travail qu'un projet en commun avec le centre AIP-PRIMECA Auvergne a été envisagé pour tester à l'aide d'un démonstrateur le concept et les fonctionnalités de SDBuilder® sur un système réel. Le démonstrateur est implémenté sur la ligne d'assemblage de l'AIP-PRIMECA .

Dans le cadre de ce projet, un modèle de simulation de la ligne d'assemblage et différents modèles de prise de décision à différents niveaux seront développés. Dans une première phase, les modèles de prise de décision seront connectés au modèle de simulation de la ligne ce qui permettra d'étudier différentes idées de pilotage et de trouver les méthodes les plus appropriées. Ensuite, l'application de prise de décision en ligne sera connectée aux automates au travers des moyens de communication présents sur la ligne. La problématique de simulation en ligne d'une unité de production a déjà été évoquée par Kouiss et Najid (Kouiss et Najid, 2004).

Cet article présente les grandes lignes de ce projet et les mécanismes présents au sein de SDBuilder® pour la prise de décision et la simulation en ligne.

L'article est structuré de la manière suivante. Après un bref état de l'art nous présenterons la ligne d'assemblage de l'AIP-PRIMECA, nous analyserons les décisions à prendre aux différents niveaux ainsi que l'architecture du démonstrateur. Nous présenterons ensuite le modèle de simulation de la ligne. Enfin, nous terminerons par les conclusions et les perspectives.

## 2. Etat de l'art

La simulation est un outil bien connu du monde industriel pour la conception, la re-conception ou le dimensionnement des systèmes de production. Mais elle peut être utilisée également pour l'aide à la décision au cours de la production. En effet, la simulation (Varady, 1999) permet une modélisation fidèle de l'outil de production et peut prendre en compte sa complexité réelle et son caractère dynamique. Elle peut donc apporter un réel avantage pour un pilotage efficace et une optimisation des performances.

Cependant, bien que l'utilisation de la simulation pour aider au pilotage soit attirante au regard du concept, elle reste souvent difficile à mettre en place à cause de la difficulté de collecter les données du terrain de manière cohérente et automatique (Davis, 1999). Il est difficile de garder un modèle et ses données suffisamment proches de la réalité pour que les résultats soient exploitables. Alors, bien souvent, les décideurs se contentent de prévisions faites à partir de simples tableurs pour piloter leurs systèmes de production. De plus, les logiciels de simulation eux-mêmes ne facilitent pas toujours la prise en compte de données réelles. La simulation en ligne présente donc deux problèmes que sont, d'une part, la collecte efficace et automatique des données du terrain et, d'autre part, leur intégration dans le modèle de simulation.

L'idée de rapprocher la simulation - et plus largement la modélisation - avec les données récoltées sur le terrain, a déjà été étudiée dans plusieurs travaux de recherche. Manivannan et Banks (Manivannan, 1992) ont étudié la liaison entre un modèle de simulation et des données du terrain pour obtenir des résultats exploitables à court terme. Ils ont proposé deux applications possibles. Dans la première ils présentent l'intérêt d'un tel système pour l'aide à la décision à travers l'étude de différents scénarii lors de l'occurrence d'une panne dans le système. Dans la deuxième application, ils ont proposé d'utiliser cet outil pour détecter des baisses de qualité et ainsi effectuer une maintenance préventive plus efficace. De plus, les auteurs se sont intéressés aux difficultés rencontrées lors de la mise en œuvre et lors de l'exploitation du système et en ont retiré quelques facteurs-clés de réussite : le rafraîchissement des données collectées (fréquence), la façon de les collecter (automatique, manuelle), la stabilité du système modélisé et la fréquence des simulations.

Drake (Drake, 1996) insiste également sur les difficultés de la mise en place de la simulation en ligne. Il donne un schéma général de mise en place de cette simulation vis à vis des autres systèmes de l'entreprise et il donne un exemple d'application de ce schéma à l'aide du langage de simulation SIMAN. L'article se focalise sur le pilotage et l'analyse de scénarii à court terme.

Kouiss et Pierreval (Kouiss, 1999) ont aussi travaillé sur ce concept : mettre en relation des données du terrain et la simulation. Le but de leur application est de permettre à l'utilisateur

d'étudier différents ordonnancements lors d'un dysfonctionnement, afin d'éviter une trop grande perte de productivité. Tout événement pouvant être préjudiciable à la productivité est considéré comme un dysfonctionnement : des pannes, des arrêts pour maintenance machine, des ruptures de stocks, etc. Le levier d'action utilisé est l'ordonnancement, élément déterminant pour les performances du système.

Dans (Smith et al., 1996), les auteurs se sont particulièrement attachés à la proximité du modèle avec le système, et ils ont proposé une nouvelle voie de développement. En effet, pour eux, l'intérêt du concept est de pouvoir estimer une date de fin de production. L'utilisateur possède un plan de charge de son système de production et, grâce à des simulations, va pouvoir estimer la date de fin de traitement de ce plan de charge. Pour réaliser cela, et donc pour avoir des résultats fiables, il faut évidemment que le modèle soit le plus proche possible de la réalité. Ainsi, les auteurs ont établi des échanges pertinents et fréquents entre le modèle et le système. Ils proposent notamment de définir, à partir des données récoltées, les temps de traitement sur chaque poste et ce à chaque nouvelle simulation, ce qui permet de prendre en compte le mieux possible l'état du système.

Gonzalez et Davis (Gonzales, 1996) ont proposé différentes orientations pour réaliser une modélisation qui prend en compte le fonctionnement réel de l'outil de production. Les données auxquelles ils se sont le plus intéressés sont : les temps machines, les temps de transport, les temps de convoyage, les pannes (mode dégradé), la maintenance préventive (mode dégradé), le plan de charge du système et les horaires de travail.

Notre travail s'inscrit dans la continuité des travaux cités ci-dessus. Nous voulons démontrer que la simulation couplée à des modules d'optimisation peut constituer un composant important du MES (Manufacturing Execution System).

### **3. La ligne d'Assemblage du CTT**

#### **3.1 Ligne d'assemblage**

La ligne d'assemblage de l'AIP-PRIMECA Auvergne représente, dans un but pédagogique, un système de production complet possédant toutes les caractéristiques des systèmes réels utilisés en entreprise.

La ligne d'assemblage présentée à la figure 1 est composée de différents postes et de deux convoyeurs permettant aux palettes de se déplacer de poste en poste.

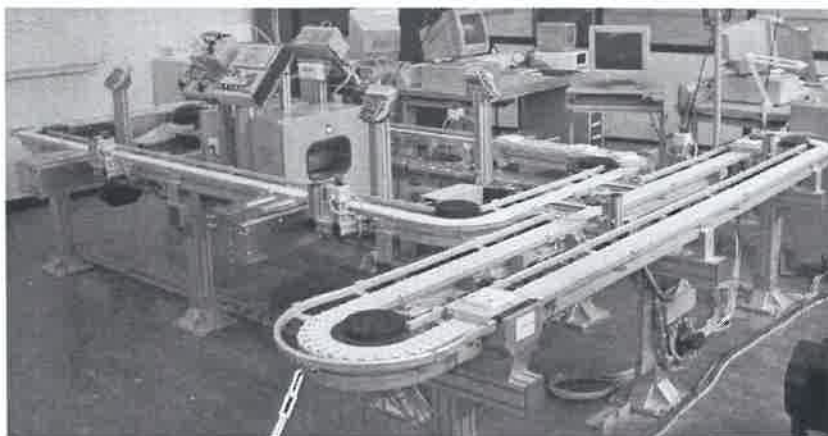


Figure 1 : Ligne d'assemblage.

Le premier convoyeur réalise la boucle de chargement-déchargement. Le second, relié au premier par deux postes de transfert, fait circuler les palettes de poste en poste pour simuler une production. La Figure 2 présente la structure de la ligne d'assemblage et les différents postes de travail.

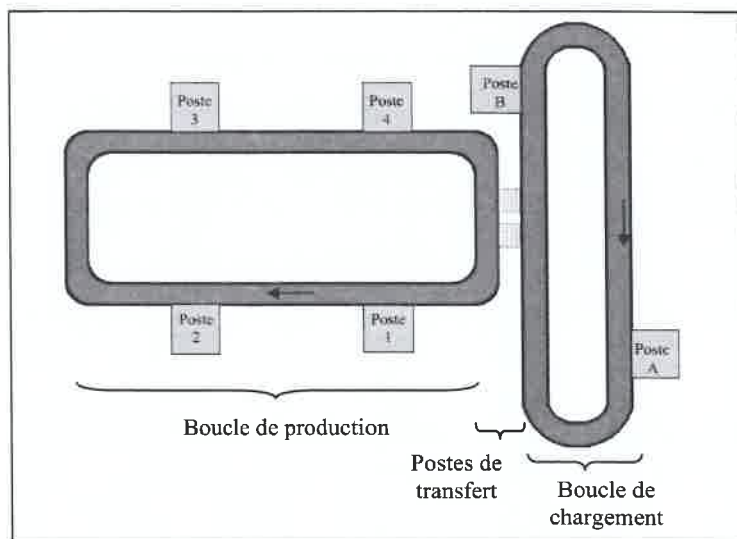


Figure 2 : Structure de la ligne d'assemblage.

Une particularité de cette ligne est l'utilisation de plusieurs couches réseaux et protocoles de communication. La première couche est dite niveau terrain, c'est un réseau de type AS-i. Il est utilisé pour accéder aux actionneurs et capteurs. La seconde est dite niveau cellule et est constituée d'un réseau Ethernet. Elle véhicule les informations entre les automates et les fonctions supérieures (supervision, simulation, ordonnancement) par l'intermédiaire du protocole TCP/IP.

### 3.2 Les pièces fabriquées :

- Il existe 4 types de pièce : A, B, C et D ;
- Les pièces se déplacent sur des palettes ;
- Le nombre de palettes est constant (10 palettes) ;
- Les gammes de fabrication des pièces peuvent posséder des routes alternatives, cependant dans un premier temps elles seront simples (les chiffres désignent les postes) :
  - ⇒ Type A : 1 - 2 - 1 - 4
  - ⇒ Type B : 2 - 4 - 1 - 4
  - ⇒ Type C : 2 - 3 - 1
  - ⇒ Type D : 1 - 2 - 1

### 3.3 Les différents postes :

- Les postes 1, 2, 3 et 4 sont des postes opérationnels ;
- Le poste A est celui de chargement des pièces sur les palettes ;
- Le poste B est celui de déchargement ;
- Il existe 2 postes dits de transfert :
  - ⇒ Un poste 'aller', qui transfère toutes les pièces s'y présentant de la boucle de chargement vers la boucle de production
  - ⇒ Un poste 'retour' qui transfère les pièces de la boucle de production ayant terminées leur gamme de fabrication vers la boucle de chargement

Lorsqu'un poste est en cours de traitement d'une pièce (travail opératoire hors chargement ou déchargement sur le poste), il peut laisser passer la pièce suivante si celle-ci n'a pas besoin du dit poste.

Les durées des opérations sont théoriquement connues mais varient généralement en fonction des compétences et/ou de l'état des opérateurs.

### 3.4 Les points de décision et d'optimisation:

Le premier niveau de décision concerne la séquence de chargement des différents types de pièces. En fonction des commandes respectives des pièces A, B, C et D et des gammes de fabrications, il existe une séquence optimale de chargement des pièces. Cette séquence pourrait être remise en cause régulièrement en fonction des arrivées des ordres de fabrication et des durées opératoires récemment observées. Cette remise en question pourrait être réalisée à l'aide d'une méta-heuristique utilisant la simulation (S.I.Yasinovsky, 1996) (V.V.Yemelyanov et al., 1996).

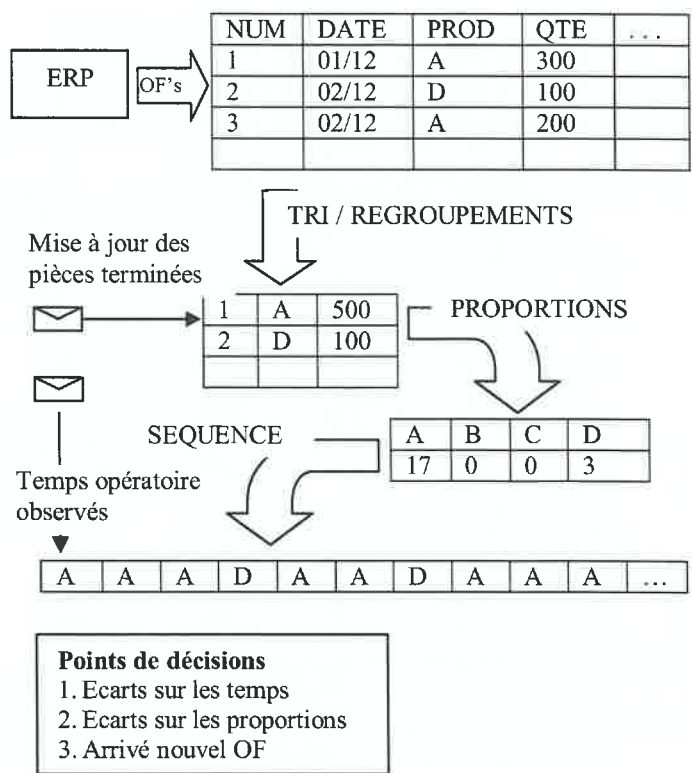


Figure 3 : Exemple de structure du modèle de décision sur les séquences de lancement.

La figure 3 illustre un exemple du modèle de décision qui détermine les séquences. Ce modèle communique avec l'ERP pour recevoir les ordres de fabrication, avec les automates de la ligne pour mettre à jour les pièces réalisées et recevoir une information sur les temps opératoires réels. Il envoie vers le poste de chargement la séquence des pièces à charger. Le but ici n'est pas de discuter de l'algorithme en lui-même qui n'est certainement pas le meilleur, mais bien d'illustrer le principe qui peut être mis en œuvre pour la prise de décision en ligne.

Le second niveau concerne la gestion des files d'attente devant les postes opérationnels et les postes de transfert. Etant donné que les temps d'opérations varient, la séquence déterminée au préalable peut se révéler désastreuse si des aléas surviennent lorsque le système est en fonctionnement. Imaginons une file d'attente d'un poste donné comme suit A, B, B, B, B, B, ... Si seule la pièce A attend le poste pour une opération longue et que les pièces B ne doivent pas passer sur le poste en question, ne serait-il pas judicieux de laisser passer la pièce A et de lui faire faire un tour complet afin de libérer les pièces B ? Cette gestion des files d'attente doit se faire en continu sur la base de l'état réel du système.

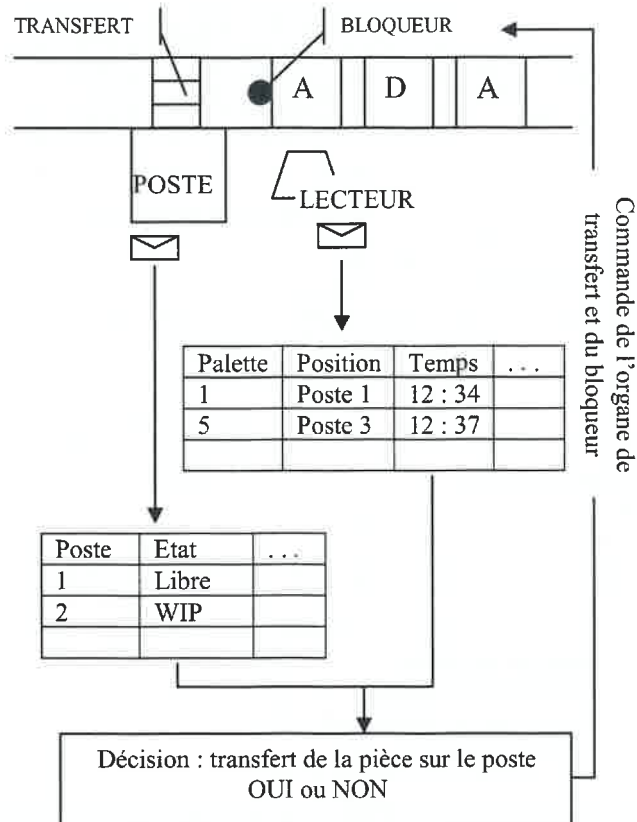


Figure 4 : Modèle de gestion des files d'attente des postes.

Ce modèle reçoit des messages qui actualisent l'emplacement des palettes sur les convoyeurs et l'état des postes. Au moment de la lecture du code à barres sur une palette, le modèle décide de bloquer ou de laisser passer cette palette. En retour, le modèle renvoie les messages à destination des automates de gestion des bloqueurs et des organes de transfert.

#### 4. Communication entre applications et automates

L'objectif final du projet est d'obtenir un modèle décisionnel qui dialogue avec la ligne d'assemblage réelle afin d'observer et de prendre des décisions, mais intègre aussi des données distantes.

La ligne d'assemblage possède un serveur OPC capable de communiquer avec les automates en temps réel et de servir ces données aux clients OPC. Cela signifie qu'il fournit à des applicatifs clients un ensemble de services d'accès à des variables d'automatisme sans que le client se préoccupe du type d'automate et de son protocole de communication.

Le serveur OPC assure l'interface entre un ensemble d'automates et un ou plusieurs applicatifs clients dans lesquels nous souhaitons consulter et/ou modifier les valeurs de



certaines données de ces automates. La figure 5 représente la structure du réseau de terrain et le serveur OPC.

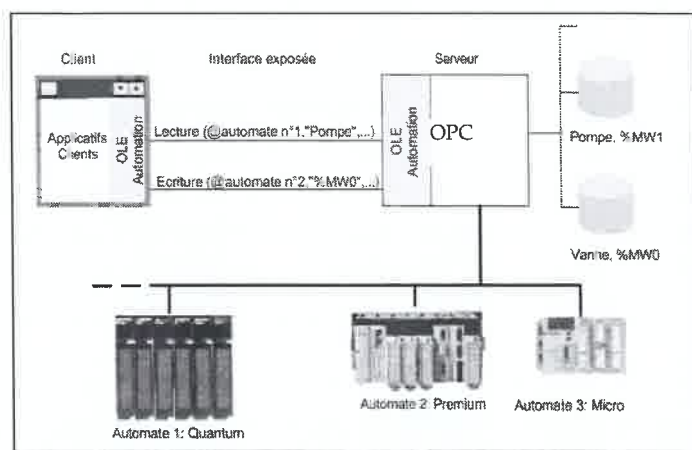


Figure 5 : Architecture OPC au sein d'un système complexe.

Le serveur OPC de la ligne d'assemblage est le produit OPC Factory Server de Schneider. Il est configuré et opérationnel pour communiquer avec les différents capteurs et actionneurs de la ligne d'assemblage. Le tableau 1 présente un exemple des variables automates accessible au travers du serveur OPC.

<i>Variables internes</i>	<i>Symboles</i>	<i>Description</i>
%MW0		Capteurs du Poste A
	%MW0:X0	Presence_pie ce Capteur Butée de présence
	%MW0:X1	Attente_pie ce Capteur Butée d'attente
	%MW0:X2	FinCourse_H aut Butée de présence levée
	%MW0:X3	FinCourse_Ba s Butée de présence baissée

Tableau 1 :Exemple de variables automates accessibles au travers du serveur OPC.

Pour communiquer avec d'autres applications, le moteur de simulation SDBuilder® utilise l'API Windows de messagerie pour envoyer entre applications des messages de type 'COPYDATA'. Les messages sont gérés par une application client/serveur de message qui se charge du cheminement des messages vers le destinataire final, local ou distant. Pour les applications distantes, le serveur utilise les communications par socket.

La plate-forme SDBuilder® possède également un client OPC paramétrable qui permet de se connecter aux serveurs OPC, d'acquérir et de stocker les données, faire le traitement des données et envoyer des messages suivant des règles définies. Ce client est également capable d'écrire dans les variables des automates pour piloter le système physique.

## 5. La structure du démonstrateur

Le démonstrateur présenté à la figure 6 est composé des éléments suivants :

- Les deux applications (modèles décisionnels SDBuilder®) chargées des prises de décision aux deux niveaux. Le modèle « séquence », qui détermine la meilleure séquence de lançements de pièces en se basant sur les ordres de fabrication venus de l'ERP, et le modèle « bloqueurs », qui décide pour chaque palette qui arrive devant un poste de la laisser passer ou de l'arrêter. Cette dernière décision dépend de l'état des postes, des prévisions sur les durées opératoires, de l'emplacement des autres palettes, des types de pièces qui se trouvent sur les palettes, etc.
- Le client OPC est configuré de sorte à envoyer via des messages tout changement important de l'état de la ligne, de façon à maintenir l'état du modèle en phase avec le système piloté. Il a également en charge l'envoi au serveur OPC des commandes de pilotage déterminées par le modèle « bloqueurs » vers les postes et les bloqueurs.

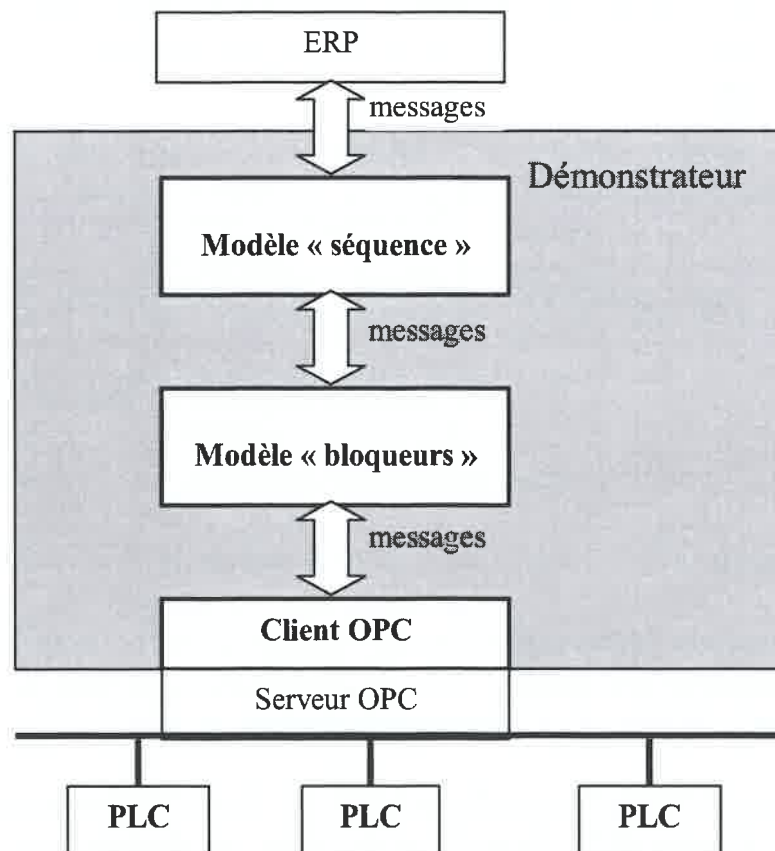


Figure 6 : Structure du démonstrateur.

## 6. Représentation d'un système en SDBuilder

Le programme SDBuilder est un logiciel de simulation à événements discrets basé sur des règles de production. Il est fortement inspiré des principes du logiciel RAO (Artiba et al., 1998). La souplesse du langage sous-jacent permet également de faire de l'optimisation, soit par méthode exacte, pour des problèmes combinatoires avec une recherche dans un graphe d'état (Nilsson Nils J., 1980), soit à l'aide de méta-heuristique, combinée éventuellement avec de la simulation. On peut, par exemple, programmer un algorithme génétique ou un algorithme de descente, dont la fonction d'évaluation est un modèle de simulation. Le principe de description des processus à l'aide des règles de production autorise également la programmation d'un système expert. Au sein d'une seule plateforme, on retrouve les fonctions de simulation, de supervision, d'optimisation et un système expert qui peuvent être combinés entre elles pour piloter un système réel. SDBuilder® peut être connecté aux automates et contrôleurs d'un système réel, au travers du protocole OPC, ou à d'autres logiciels, au travers d'une couche d'échange de messages.

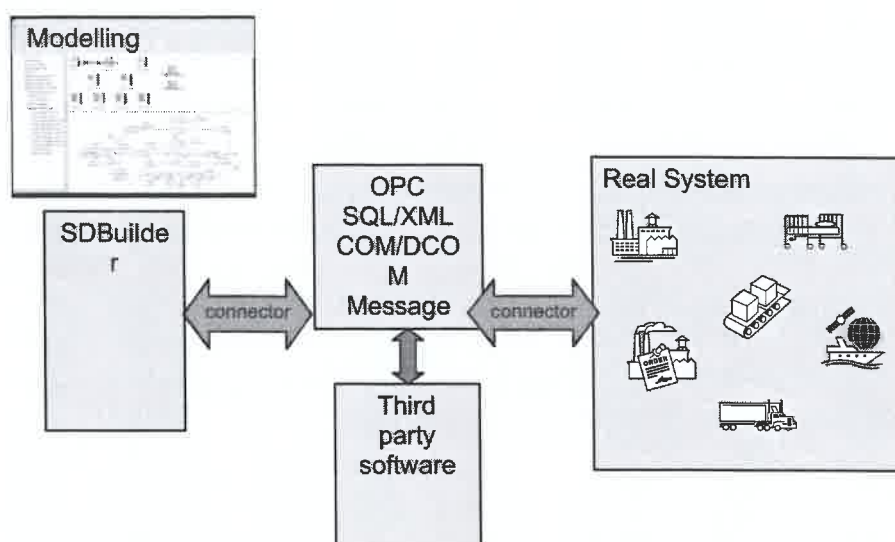


Figure 7 : Structure de la plateforme SDBuilder®.

Un système discret complexe est représenté dans le langage SDB (utilisé par SDBuilder®) comme un ensemble d'objets qui interagissent entre eux. Ces objets sont soit permanents (palettes, convoyeurs...), soit temporaires (pièces, ordres...). Ils sont caractérisés par des paramètres, le tout représentant de façon abstraite les éléments réels du système et leurs états (figure 8).

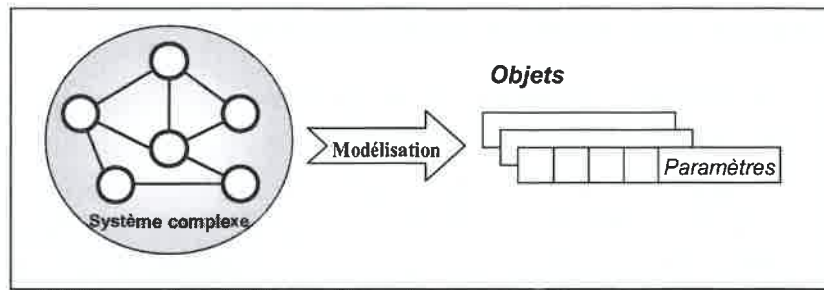


Figure 8 : Représentation des éléments d'un système discret complexe.

Les objets évoluent à travers leurs paramètres au fur et à mesure des événements. Un événement est un changement d'état d'un ou plusieurs objets en fonction de pré conditions précises. Il existe plusieurs types d'événements : les débuts et les fins d'actions, les événements irréguliers, les déclenchements des règles d'évolution et les déclenchements des règles de réaction aux messages entrants (Figure 9).

Le principe de SDBuilder® est le suivant : lors de la réception d'un message, une règle associée à ce type de messages est activée. Si la pré-condition associée à cette règle est vérifiée (en fonction de valeurs de paramètres de message et de l'état du modèle), la règle est déclenchée, et elle change l'état du modèle. Ce changement d'état, à son tour fait que d'autres règles peuvent être activées. L'enchaînement des règles réalise la logique de prise de décision. Lors de l'exécution d'une règle, l'état du système est modifié, mais des messages peuvent également être envoyés vers d'autres modèles et/ou applications, notamment les commandes aux automates.

Les événements irréguliers sont des événements qui apparaissent de façon aléatoire pour simuler par exemple des pannes.

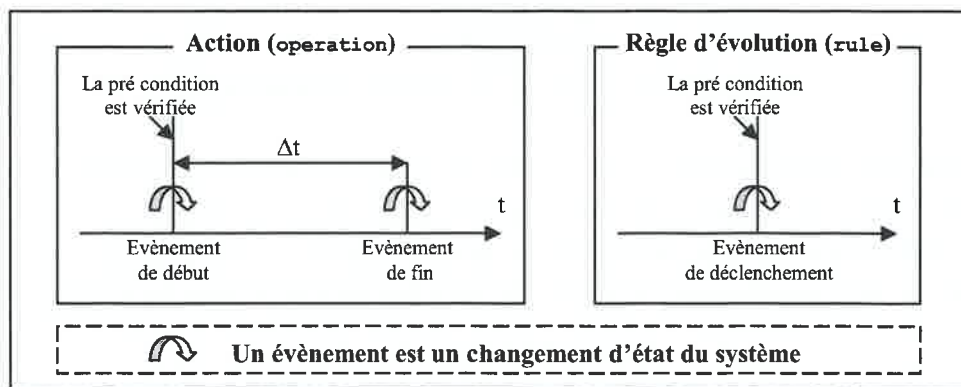


Figure 9 : Les événements en SDBuilder.

## 7. Le modèle de simulation

Dans le cadre de ce projet, la première phase consiste à réaliser un modèle de simulation de la ligne d'assemblage. Ce modèle servira à valider les modèles décisionnels qui seront développés par la suite avant de connecter ceux-ci sur le système réel. Ce modèle sera également utile dans le cas d'utilisation de méta-heuristique pour la prise de décision. C'est, en effet, ce même modèle du processus piloté qui sera utilisé pour évaluer l'impact de différents scénarii. Avec SDBuilder®, il faut décrire, d'une part, les objets qui composent le système et, d'autre part, les interactions entre ceux-ci (les règles de fonctionnement de notre système).

### 7.1 Les types d'objets

Pour modéliser au mieux le système discret complexe que constitue la ligne d'assemblage étudiée nous avons déterminé les types d'objets suivants :

- *une\_piece* : cet objet, de type temporaire, modélise les pièces présentes dans la simulation. Un objet de ce type est créé lors de l'arrivée d'une pièce dans le modèle et est détruit lorsque cette pièce est, suivant le modèle, déchargée.  
Il n'y a pas d'initialisation de ce type d'objets, ils sont créés lors du déroulement de la simulation.
- *une\_palette* : élément clé de la simulation, cet objet modélise les palettes de la ligne. Toutes les palettes présentes dans la simulation sont initialisées de telle sorte à être placées en attente de chargement.
- *un\_atelier* : cet objet contient les paramètres globaux de la ligne.
- *un\_convoyeur* : objet nécessaire à la gestion des convoyeurs et donc à la progression des palettes, il permet de définir les différents convoyeurs du système.
- *un\_poste* : cet objet modélise un poste de travail. Il est « occupé » lorsqu'une palette y est chargée ou déchargée et il est dans l'état « wip » si le poste est en cours d'opération.
- *un\_bloqueur* : ce type d'objet modélise les actionneurs bloquants de la ligne d'assemblage. La particularité de l'utilisation d'un objet « un\_bloqueur » est de permettre, dans le cas d'une connexion avec le système réel, de modéliser directement les actionneurs de la ligne réelle. L'emplacement des bloqueurs sur la ligne est présenté sur la figure 10.
- *un\_element* : ce type d'objet modélise un maillon de la séquence d'arrivée des pièces.
- *une\_operation* : ce type nous permettra de définir pour chaque type de pièce sa gamme de fabrication.

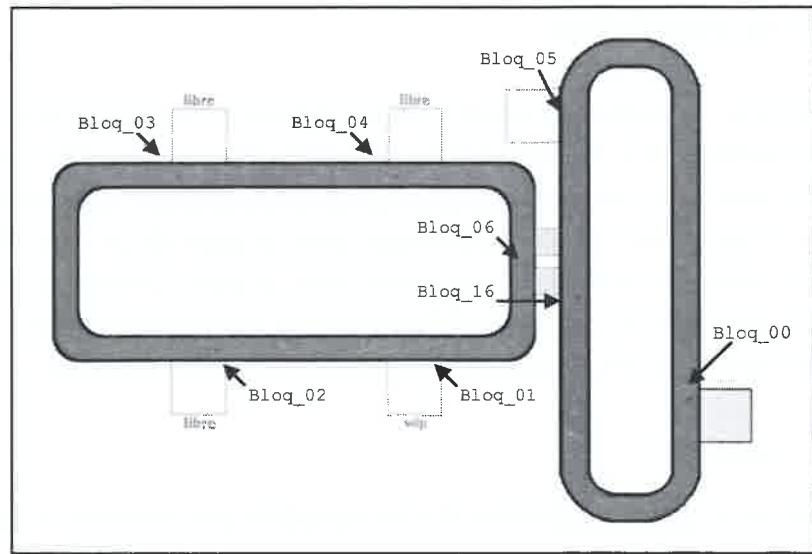


Figure 10 : Emplacement des bloqueurs.

## 7.2 Les règles de fonctionnement

La simulation du modèle est une succession d'évènements changeant l'état de ses objets. Les patterns sont les définitions des différents évènements au travers des actions et des règles d'évolution.

- Gestion des convoyeurs : cette action décrit un pas d'avancement d'une palette sur un convoyeur non bloqué. Les nouvelles coordonnées de la palette sont recalculées suivant sa position courante et la géométrie de la ligne.
- Règles de circulation : nous avons traité le problème de progression des palettes, il est nécessaire, maintenant, de s'occuper de leur trajectoire et de leur positionnement. Pour cela nous avons traité le problème simplement, en utilisant uniquement les coordonnées pour modifier les paramètres de déplacement des palettes. Ces règles sont au nombre de huit, une pour chaque angle de convoyeur.
- Gestion des collisions : les palettes peuvent désormais circuler virtuellement sur les convoyeurs. Cependant, si une palette est bloquée, il faut que les suivantes s'accumulent derrière. Pour ce faire nous avons découpé les convoyeurs en zone droite, puis nous comparons chaque palette entre elles pour replacer celles qui se chevauchent.
- Chargement des pièces : avant de pouvoir charger une pièce sur une palette, il est nécessaire de bloquer une palette devant le poste de chargement, le poste 0. Ce blocage se fait dès qu'il y a une palette vide devant le poste.  
Le chargement de la pièce est géré par une action particulière qui se déclenche si l'utilisateur clique sur le poste de chargement lors de la simulation. Si à ce moment

là, il y a une palette vide devant ce poste et si le poste est libre, une pièce est chargé sur la palette. Le type de la pièce est donné par la séquence de lancement.

- Déchargement des pièces : si une palette se présente devant le poste de déchargement (poste 5) et si le poste est libre, la pièce est enlevée de la palette. A la fin de cette action, l'objet représentant la pièce est supprimé.
- Gestion des bloqueurs : les bloqueurs sont autonomes, leur état est fonction des postes et des palettes proches. Ce choix a été fait pour pouvoir les gérer facilement, une fois le modèle connecté à la ligne réel. En effet, ils seront directement liés aux actionneurs bloquants de la ligne.

Tous les postes gèrent eux-mêmes les bloqueurs immédiatement avant. Au niveau des bloqueurs 16 et 06, la gestion est plus particulière, car ils sont certes bloquants lorsque le poste de transfert est occupé mais ils vérifient aussi qu'il existe une place disponible sur le convoyeur de réception.

Pour le blocage des palettes contre un bloqueur, nous utilisons le même principe que la gestion des collisions. C'est en comparant les coordonnées des palettes avec les coordonnées dites « réelles » du bloqueur en question que la palette est bloquée ou non.

- Gestion des postes : chaque poste est géré par quatre règles  
`Presence_Piece_EntreeN_Pat`: détection d'une pièce au niveau du bloqueur, si le poste N est libre et que la pièce doit y subir une opération alors la palette y est envoyée.

`Presence_Piecebis_EntreeNbis_Pat`: détection d'une pièce au niveau du bloqueur, si le poste est en cours d'opération et que la pièce ne doit pas y subir une opération alors nous laissons passer la palette.

`Traitement_PosteN_Pat`: une fois la pièce positionnée sur le poste N, son état devient 'wip', c'est-à-dire 'en cours d'opération'. Une fois le délai d'opération passé la palette est renvoyée vers le convoyeur et le poste reprend son état « occupe ».

`Presence_Piece_SortieN_Pat`: détection d'une pièce en sortie du poste N, la palette reprend son cheminement normal sur le convoyeur de production et le poste est libéré.

### 7.3 La fenêtre d'affichage

Fonction importante du démonstrateur, la fenêtre d'affichage est l'interface directe avec l'utilisateur. La description de l'écran d'animation permet d'obtenir l'interface graphique de la simulation. L'écran d'animation affiche les changements d'état de la ligne et permet à l'utilisateur de charger les pièces et de bloquer/débloquer les postes (figure 11).

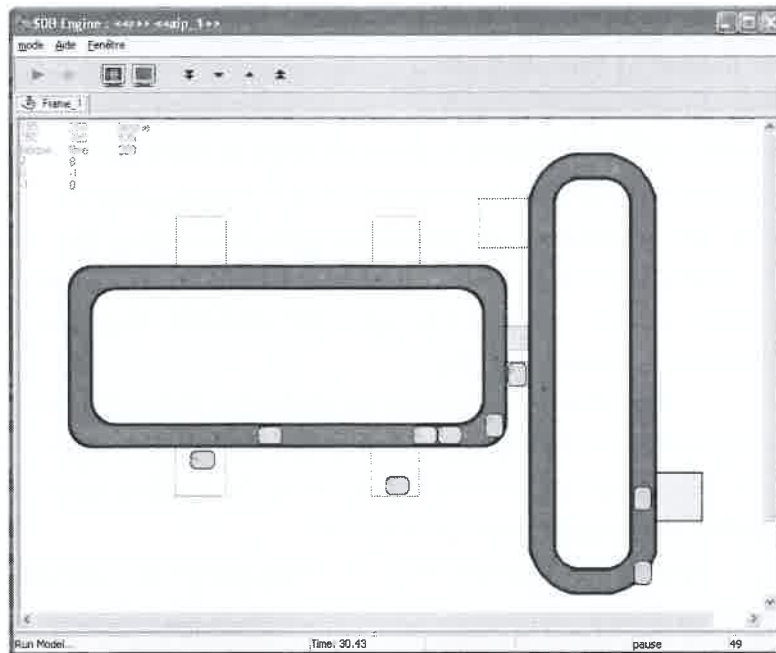


Figure 11 : Ecran d'animation du modèle.

## 8. Conclusions et perspectives

Dans cet article, nous avons présenté un projet développé pour démontrer l'efficacité et les possibilités offertes par le couplage de la simulation et de l'optimisation avec une ligne de production. Nous avons présenté l'architecture et les principes qui ont été mis en œuvre dans ce démonstrateur afin que le modèle de simulation soit connecté au système physique réel.

Le modèle de simulation réalisé reproduit fidèlement l'évolution des grandeurs physiques du système réel. On retrouve entre autre : la position des bloqueurs, la constitution des files d'attente, les informations fournies par les lecteurs de code à barres et l'état des postes. L'objectif du projet n'était pas l'optimisation en soi, mais bien de démontrer la faisabilité du couplage entre la simulation, l'optimisation et un système réel de production. Les essais réalisés à ce jour sont très prometteurs et montrent qu'il est possible en utilisant des protocoles industriels éprouvés tel que OPC, de coupler la simulation avec une ligne de production réelle et de faire en sorte que la simulation devienne un outil de prise de décision et de pilotage en temps réel. Les temps de prise de décision mesurés sont tout à fait compatibles avec la dynamique de cette ligne d'assemblage. D'autres travaux de recherche pourront être menés maintenant, portant, entre autres, sur les algorithmes de prise de décision plus sophistiqués qui peuvent être mis en œuvre afin d'améliorer la qualité du pilotage.



## 9. Bibliographie

- A. Artiba, V.V. Emelyanov, S.I. Iassinovski., 1998. Introduction to Intelligent Simulation: The RAO language. *Kluwer Academic Publishers*, 1998.
- Drake G. and Smith J., 1996. Simulation System for Real-Time Planning, Scheduling, and Control *Proceedings of the 1996 Winter Simulation Conference*, pp 1083-1090, 1996.
- Davis M., 1999. Simulation : Technologies in the New Millenium *Proceedings of the 1999 Winter Simulation Conference*, pp 141-147, 1999.
- Gonzalez F. and Davis W., 1998. Initializing On-Line Simulation from the State of a Distributed System *Proceedings of the 1998 Winter Simulation Conference*, pp 507-513, 1998.
- Kouiss K. and Pierreval H., 1999. Implementing an On-Line Simulation in a Flexible Manufacturing System *Proceedings of the ESS'99 Conference, Nuremberg, Allemagne, 26-28 octobre*, pp 484-488, 1999.
- Kouiss K. and N. Najid, 2004. Un couplage MES-simulation pour le pilotage d'un système de production. *Proceedings of MOSIM'04, Nantes, September 1-3, 2004*, p. 585-592
- Manivannan S. and Banks J., 1992. Design of a Knowledge-Based On-Line Simulation System to Control a Manufacturing Shop Floor *IIE Transactions*, vol.24, n°3, pp 72-83, 1992.
- Smith J., Peters B., Curry J., Drake G. and LaJimodièr C., 1996. Advanced Tutorial – Simulation-Based Scheduling and Control *Proceedings of the 1996 Winter Simulation Conference*, pp 194-198, 1996.
- Varady M., 1999. Dynamic Simulation Based Scheduling *IPA Slovakia 1*, 1999.
- Nilsson Nils J., 1980. Principles of Artificial Intelligence. *Tioga Publishing*, 1980.
- Yasinovsky S., 1996. Hybrid System for Job-Shop Scheduling. *Proc. of CESA'96 IMACS Multiconference on Computational Engineering in Systems Application*. Lille - France, July 9-12, 1996. p.915-920.
- Yemelyanov V., Yasinovskii S., Kryuchkov M., 1996. A dynamic optimal method for cutting out material using a genetic algorithm. *Proc. of First International Conference on Evolutionary Computation and Its Applications (EvCA'96) June 24-27, 1996, Moscow, Russia* p.212-221.