



**HAL**  
open science

# TRAITEMENT DES DONNÉES DES SYSTÈMES DE PRODUCTION POUR LES SYSTÈMES D'AIDE À LA DÉCISION: PROPOSITIONS D'APPROCHES DE CONCEPTION ALTERNATIVES.

Mickaël Adam, Olivier Cardin, Pierre Castagna, Pascal Berruet

► **To cite this version:**

Mickaël Adam, Olivier Cardin, Pierre Castagna, Pascal Berruet. TRAITEMENT DES DONNÉES DES SYSTÈMES DE PRODUCTION POUR LES SYSTÈMES D'AIDE À LA DÉCISION: PROPOSITIONS D'APPROCHES DE CONCEPTION ALTERNATIVES.. 9th International Conference on Modeling, Optimization & SIMulation, Jun 2012, Bordeaux, France. hal-00728683

**HAL Id: hal-00728683**

**<https://hal.science/hal-00728683v1>**

Submitted on 30 Aug 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## TRAITEMENT DES DONNÉES DES SYSTÈMES DE PRODUCTION POUR LES SYSTÈMES D'AIDE À LA DÉCISION: PROPOSITIONS D'APPROCHES DE CONCEPTION ALTERNATIVES

M. ADAM, O. CARDIN, P. CASTAGNA

LUNAM Université, IUT de Nantes - Université de  
Nantes, IRCCyN UMR CNRS 6597 (Institut de  
Recherche en Communications et Cybernétique de  
Nantes), 2 avenue du Pr Jean Rouxel - 44475 Carquefou  
- France  
mickael.adam/olivier.cardin/pierre.castagna@univ-  
nantes.fr

P. BERRUET

Lab-STICC, UMR CNRS 3192 (Laboratoire en sciences  
et technologies de l'information, de la communication et  
de la connaissance), Université Européenne de Bretagne  
- UBS, Centre de Recherche - BP 92116 - F-56321  
Lorient Cedex - France  
pascal.berruet@univ-ubs.fr

**RÉSUMÉ :** Avec l'augmentation des taux de production et de la flexibilité, la complexité des systèmes de production atteint un point où l'opérateur en charge du pilotage du système n'est pas en mesure de prévoir efficacement l'impact de ses décisions sur les performances globales du système. Ainsi, de plus en plus de Systèmes d'Aide à la Décision (SAD) sont développés, tant dans la littérature que dans des applications industrielles. Les SAD ont un point commun: l'initialisation de leur fonctionnalité de prévision est basée sur des données provenant du système de production. Considérant la variabilité des données d'entrées et de sorties, un traitement des données est nécessaire pour adapter celles venant du système de production. Le but de cet article est de présenter plusieurs approches de conception permettant à l'intégrateur d'un nouveau système de commande d'en faciliter l'implémentation, ceci avec l'idée d'automatiser et de systématiser au maximum les phases de conception grâce à l'ingénierie dirigée par les modèles. L'introduction permettra ainsi de contextualiser la problématique. Des solutions permettant de résoudre les problèmes soulevés seront proposées dans un second temps, et enfin une proposition d'automatisation de certaines de ces solutions afin d'en faciliter l'implémentation.

**MOTS-CLÉS:** Système d'Aide à la Décision, Traitement des Données, Systèmes de Production, Ingénierie Dirigée par les Modèles, Transformation de Modèle, Observateur.

### 1 INTRODUCTION

Avec l'augmentation des taux de production et de la flexibilité, la complexité des systèmes de production atteint un point où l'opérateur en charge du pilotage du système n'est pas en mesure de prévoir efficacement l'impact de ses décisions sur les performances globales du système.

Ainsi, de plus en plus de Systèmes d'Aide à la Décision (SAD) sont développés, tant dans la littérature que dans des applications industrielles. Chaque SAD a ses propres performances et hypothèses, mais ils ont tous un point commun: l'initialisation de leur fonctionnalité de prévision est basée sur des données provenant du système de production. En outre, cette fonction est fondamentale, étant donné qu'elle a un impact direct sur la justesse des prévisions.

Étant donné que les données disponibles du système ne sont généralement pas directement utilisables pour l'initialisation du SAD, un traitement des données est nécessaire pour les adapter (figure 1). La difficulté majeure réside dans le fait que les entrées de données du système et celles requises par le SAD peuvent être très différentes et donc difficile à lier.

Enfin, le problème de temps de réponse est très important quand il s'agit de SAD à court terme. En effet, comme les prévisions sont réalisées sur un horizon court, le traitement des données se doit d'être rapide pour que l'opérateur puisse prendre sa décision aussi rapidement que possible.

#### Données Système

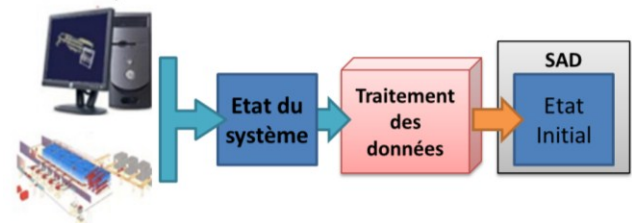


Figure 1 : Initialisation du Système d'Aide à la Décision

Dans le chapitre 2 est réalisé un état de l'art sur les SAD et les systèmes de production découplés. Afin de coupler ces deux éléments, les différents travaux relevés montrent que ce couplage est effectué à la main. Le chapitre 3 détaillera les verrous scientifiques empêchant l'automatisation de ce couplage. La contribution théorique, présentée chapitre 4, repose sur la proposition de solutions permettant de lever ces verrous. Finalement, le

chapitre 5 présentera une nouvelle approche pour la mise en application des solutions présentées les plus génériques et les plus complexes à mettre en œuvre. Cette approche est principalement basée sur l'Ingénierie Dirigée par les Modèles (IDM).

## 2 SYSTÈME DE PRODUCTION ET SAD

Quand une perturbation a lieu dans un système de production, il est généralement nécessaire de prendre une décision. De nombreux systèmes d'Aide à la Décision peuvent être trouvés dans la littérature. Cette section introduit quelques-uns des plus utilisés et détaille le traitement de données nécessaire pour les alimenter.

Certains des SAD utilisent la simulation en ligne. Les performances de la simulation en ligne, basée sur des décisions à court terme calculées par un simulateur à événements discrets, sont connues depuis plusieurs décennies. Elle a récemment été expérimentée sur des systèmes complexes via l'utilisation d'observateurs (Cardin *et al.*, 2009). Sur des problèmes spécifiques (Mahdavi *et al.*, 2010a), (Mahdavi *et al.*, 2010b) utilisent pour l'initialisation une liste des événements passés et une liste d'événements futurs probables calculés. Dans (Chong *et al.*, 2003), quand une perturbation a lieu, plusieurs nouveaux ordonnancements sont calculés. Un simulateur à événements discrets les évalue avant de choisir le plus adéquat pour la décision.

Un autre genre de SAD s'appuie sur le champ de l'intelligence artificielle via la logique floue (Mok, 2009) ou les réseaux de neurones. Les applications sont larges, allant de la définition du contrôle de la tolérance aux défaillances de capteurs (Magdy, 2009) à la prédiction de l'usure de fraise d'usinage par mesure de l'effort de pénétration et aux mesures de couples issus de la coupe (Yang *et al.*, 2009). Ces systèmes ont besoin d'une phase d'apprentissage, généralement obtenu par une observation des états du système lors d'une mise en production (Pierreval, 1992).

Cette étude montre que les jeux de données provenant du système de production et alimentant les différents SAD rencontrés dans la littérature sont très variables. En outre, la variabilité des données disponibles et de la précision requise (allant du taux de charge d'une file d'attente au pied d'une machine à l'identification exacte des produits présents dans cette file, par exemple) est potentiellement infinie.

## 3 LE PROBLÈME DE DISPONIBILITÉ DES DONNÉES POUR L'INITIALISATION DES SAD.

Cette section délimite la problématique. Pour l'illustrer, un cas d'étude simple est présenté, basé sur un système de convoyeur avec une station de travail en dérivation.

### 3.1 Introduction

Comme montrée sur la figure 2, la structure physique de la station de travail est constituée de 2 convoyeurs, 3 capteurs et 1 butée. Quand un produit se trouve au niveau du capteur 1, et si la station de travail 1 est libre, alors la butée 1 laisse le produit passer et la divergence fait rentrer le produit dans la station.

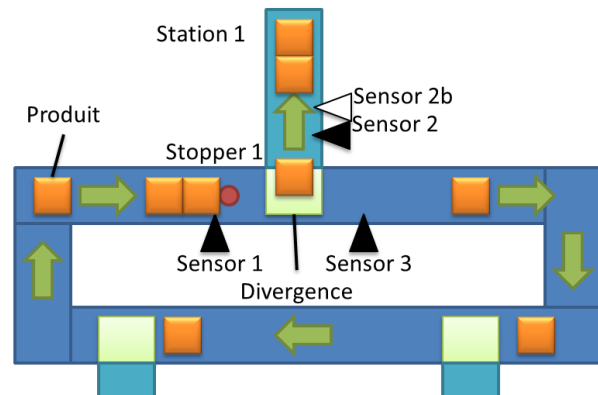


Figure 2 : L'exemple de la station de travail

### 3.2 Les données d'un système de production

Un système de production est constitué de beaucoup de données, de nature et de localisation variables. Parmi les données les plus courantes, nous pouvons citer les données venant des capteurs, de la commande au sens large ou des outils de pilotage de la production, comme l'état de l'exécution des ordres de fabrication par exemple. Il est théoriquement nécessaire pour disposer des valeurs de ces variables de définir, au temps  $t$ , l'ensemble  $S_{system}(t)$ , correspondant à l'état du système.

Au sein de cet ensemble, le sous-ensemble des variables observables  $S_{obs}(t)$  contient toutes les valeurs des variables qui peuvent être lues au temps  $t$ . Ces données peuvent être localisées dans les organes de commande (et généralement facilement récupérable via des protocoles comme OPC, MODBUS, etc...) ou dans la base de données du MES par exemple.

Le sous-ensemble des variables non observable  $S_{unobs}(t)$  contient toutes les variables auxquelles une valeur ne peut être lue au temps  $t$ . Certaines de ces variables sont dans ce sous-ensemble à tout instant, alors que d'autres peuvent migrer d'un ensemble à l'autre au cours du temps. L'exemple le plus classique est la localisation d'un produit sur un convoyeur disposant d'une commande minimaliste. Sa position exacte est connue quand il est en face d'un capteur mais aucune donnée ne donne directement l'exacte position du produit entre deux capteurs.

L'état du système peut ainsi être défini comme l'union des deux sous-ensembles précédemment définie :

$$S_{system}(t) = S_{obs}(t) \cup S_{unobs}(t) \quad (1)$$

Avec:  $S_{obs}(t) \cap S_{unobs}(t) = \emptyset \quad (2)$

### 3.3 Représentation concurrente de données au sein de la commande.

Parmi toutes les données disponibles dans la commande du système, concentrons-nous sur l'implémentation des ressources partagées pour la station de travail 1. Différentes implémentations classiques peuvent en effet être considérées.

La figure 3 montre trois façons différentes de coder l'autorisation pour un produit d'entrer dans la station. Ces trois exemples sont tous écrits en SFC utilisant la même norme IEC 61131-3, afin de faciliter la lecture, mais le lecteur doit garder à l'esprit que le langage peut potentiellement être également différent.

La première façon de coder est d'évaluer le délai d'activation du capteur d'entrée. Quand un seuil critique est atteint, la station est considérée comme pleine. La seconde façon, plus classique, utilise un compteur, représentant le nombre de produits présents dans la station. L'incrément du compteur seule est représentée, alors que le décrémentation est considéré comme localisé au niveau du programme de sortie de la station. La troisième façon utilise un second capteur (2b), localisé juste après le capteur (2). Quand les deux capteurs sont actifs au même moment, alors la station est considérée comme pleine.

$S_{obs}(t)$  diffère clairement entre ces trois exemples, successivement contenant les variables tel que « sensor1 », « sensor2 », « sensor2b », « stopper1 » ou « nbProd ».

Si le SAD requiert la donnée sur l'état de la file d'attente de la station (« pleine » / « pas pleine » / « vide »), une variable (« nbProd »), ou deux variables (« sensor2 » et « sensor2b ») ou le temps d'activation d'une étape du grafcet doivent être récupérés. Cependant, il est nécessaire que ces données soient adaptées aux besoins du SAD.

Ainsi, bien que ces trois façons de coder aient le même comportement macroscopique, elles n'autorisent pas l'implémentation du même SAD directement relié à la commande

### 3.4 Les prérequis du SAD et le besoin d'adaptation des données.

Cette section prend l'exemple d'un SAD qui est utilisé quand un produit doit entrer dans la station et que la file d'attente en entrée de poste est pleine. Ce SAD fournit une décision à savoir s'il est préférable d'attendre que la station devienne libre – mais en bloquant les autres produits sur le convoyeur – ou rester sur le convoyeur et faire un tour de plus avant de rentrer dans la station – avec un haut niveau de risque pour qu'un autre produit rentre entre temps et occupe la station. Cette sorte de décision est délicate, étant donné que rentrent en jeu un grand nombre de paramètres.

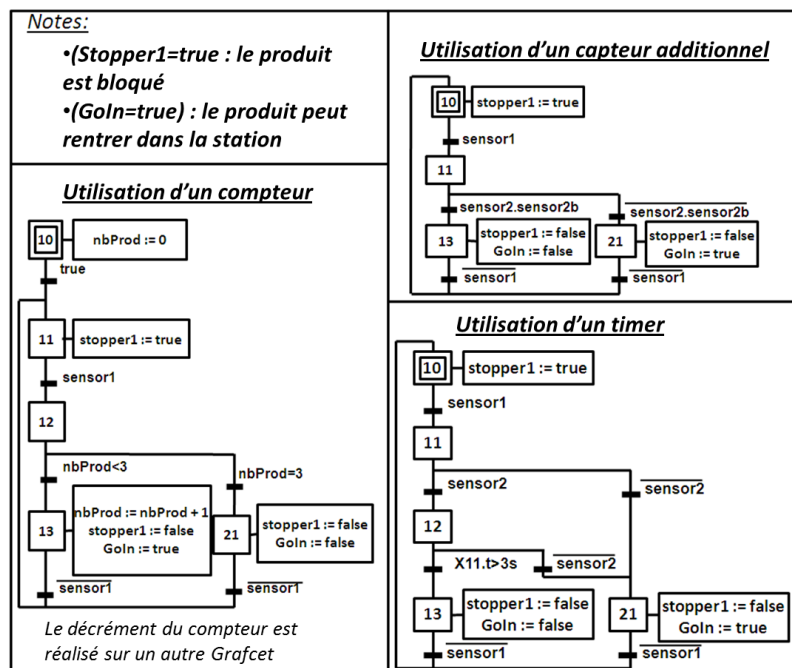


Figure 3 : Trois façons de coder l'entrée dans la station de travail

Pour pouvoir fournir l'information la plus précise possible, le SAD a besoin d'un ensemble de données venant du système pour son initialisation : cet ensemble est noté  $S_{needed}(t)$ . Cet ensemble dépend de plusieurs paramètres, comme la nature du SAD, le temps et les objectifs de la question demandée au SAD. Évidemment cet ensemble ne correspond pas à l'ensemble de toutes les données composant l'état du système :

$$S_{system}(t) = S_{needed}(t) \cup S_{not_needed}(t) \quad (3)$$

Considérant le SAD de cet exemple, il est évident que la décision doit être à court terme. Ainsi, une caractéristique importante de l'adaptation de donnée entre  $S_{system}(t)$  et  $S_{needed}(t)$  est le temps de réponse. En effet, un temps de réponse important fournit un biais dans la prévision, qui doit être réduit à sa valeur minimale. Si  $S_{needed}(t) \cap S_{unobs}(t) \neq \emptyset$ , alors une reconstruction, potentiellement consommatrice de temps, est nécessaire.

### 3.5 Problématique

La dernière partie met en évidence les trois principaux problèmes auxquels on se heurte lors de l'implémentation d'un SAD associé à une commande existante. Premièrement, la liste et la nature des variables nécessaires pour l'initialisation du SAD sont généralement complètement différentes d'un SAD à un autre. Deuxièmement, la correspondance avec les données disponibles de l'état du système est difficile à établir. Finalement, le temps de réponse pour l'initialisation du SAD est une caractéristique assez importante qui doit être prise en compte lors de son implémentation.

Pour faire face à ces problèmes, la prochaine section introduit plusieurs alternatives concernant le traitement des données pour les SAD.

## 4 TRAITEMENT DES DONNÉES POUR L'INITIALISATION DES SAD

Dans cette section, des architectures variées, et le traitement associé sont présentés avec l'objectif d'initialiser le SAD couplé au système de production. En effet, selon le contenu de  $S_{obs}(t)$  et les besoins de  $S_{needed}(t)$ ,

l'architecture peut être différente. L'illustration est basée sur l'exemple présenté dans la figure 2.

### 4.1 Première solution : transfert direct de donnée

La solution la plus facile (figure 4) consiste à initialiser directement le SAD avec les états des variables contenues dans la commande du système  $S_{obs}(t)$ . La difficulté ici est d'avoir une commande et un SAD utilisant les mêmes variables.

En effet, le SAD ne peut avoir accès aux variables  $S_{unobs}(t)$  et ne peut non plus les reconstruire, n'ayant aucune information sur l'évolution de ces variables dans le temps  $t$ . La prévision du SAD en est alors moins précise, à moins que la commande fournisse un grand nombre de données, ce qui est généralement coûteux à implémenter.

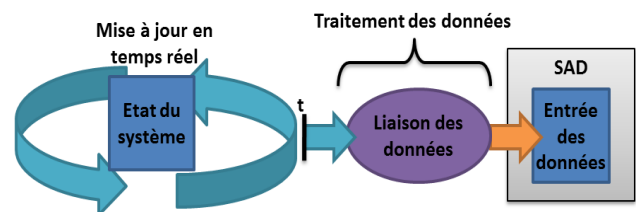


Figure 4 : Transfert direct des données

Comme montrée sur la figure 4, une liaison des données est nécessaire. Cette association n'est réalisée qu'une seule fois, et permet d'établir les relations entre les variables du système et les variables du SAD. Cela répond au problème exposé dans la section 3.3.

### 4.2 Deuxième solution: l'utilisation d'un historique d'événements

Pour connaître les données incluses dans  $S_{unobs}(t)$ , une solution consiste à utiliser un historique des événements passés du système. Cet historique est stocké tout au long de la production, et utilisé lorsque le SAD a besoin d'être initialisé (figure 5).

L'exemple des produits sur un convoyeur est révélateur. Quand le SAD est initialisé, les dernières occurrences des capteurs d'entrée et de sortie sont traitées. Associé à la vitesse des convoyeurs, la position des produits peut

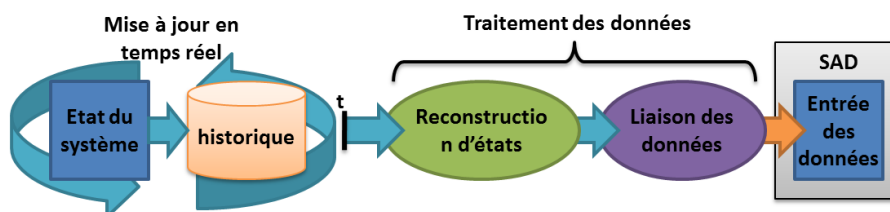


Figure 5 : Utilisation d'un historique d'états

être reconstruite par une simple régression. Le contenu de la file des produits dans un convoyeur à accumulation peut alors être retrouvé à partir de la même donnée.

Évidemment, la liaison des données est toujours nécessaire quand a lieu le paramétrage de la connexion entre le système et le SAD. Cependant, le traitement des données est aussi composé d'une reconstruction des états, qui peut être un processus long. Cela implique que cette solution n'est pas toujours compatible avec les applications pour les SAD à très court terme. Un autre problème demeure dans la quantité de données devant être stockées pendant la durée de vie du système.

### 4.3 Troisième solution: Utilisation d'un reconstruc-tueur d'état générique

La troisième solution présentée repose sur la même idée, mais essaie d'écarter le problème de la consommation de temps et du stockage des données.

Cette solution (Cardin, 2007) autorise la construction d'un  $S_{system}(t)$  du système aussi complet que possible grâce à un observateur, aussi appelé reconstruc-tueur d'états. L'état de l'observateur peut donc être considéré, à n'importe quel moment, le plus proche possible de  $S_{system}(t)$ .

L'initialisation du SAD (figure 6) est ainsi similaire à la première solution, basée sur un transfert direct des données entre l'état de l'observateur et du SAD, avec un très court temps de traitement des données.

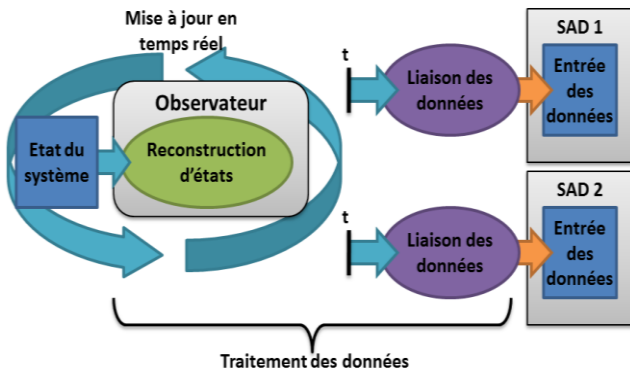


Figure 6 : Utilisation d'un observateur générique

Cette solution est très efficace, mais de grands problèmes demeurent encore :

1. La conception de l'observateur est une tâche difficile, car les mécanismes de synchronisation entre l'observateur et la commande sont difficiles à établir ;
2. La liaison des données est toujours présente, étant donné qu'il est nécessaire d'établir les relations entre les variables à chaque fois qu'un nouveau SAD doit être implanté.

### 4.4 Quatrième solution: Observateur orienté SAD

Afin de remédier à cette dernière question, l'idée est d'intégrer la liaison des données directement dans la phase de conception de l'observateur. L'objectif est de passer de l'état de l'observateur considéré, à tout instant  $t$ , comme étant le plus proche possible de l'image de  $S_{system}(t)$  comme c'était le cas auparavant, à l'état de l'observateur considéré, à tout instant  $t$ , comme l'image la plus proche possible de  $S_{needed}(t)$ . Par ailleurs, cela devrait diminuer la complexité de la conception de l'observateur, comme le nombre de variables doit être diminué.

La méthode consiste ainsi à intégrer le modèle de la structure de données du SAD dans la structure de données de l'observateur considéré (figure 7).

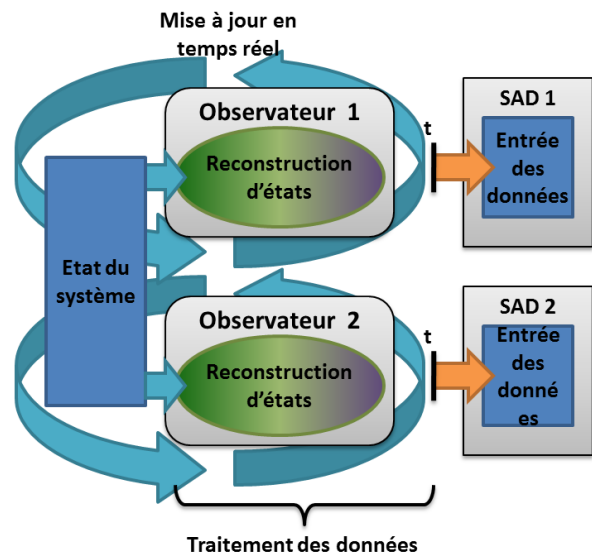


Figure 7 : Utilisation d'un observateur orienté SAD

Les principaux problèmes de cette solution sont :

1. Un observateur supplémentaire doit être conçu à chaque fois qu'un nouveau SAD doit être implémenté sur le système, étant donné que les observateurs sont dédiés à un SAD spécifique;
2. La conception de l'observateur, bien que moins consommateur de temps parce que plus courte, est encore plus complexe étant donné que la structure des données du SAD doit être intégrée.

### 4.5 Conclusion sur ces solutions au traitement des données entre un système de production et le SAD associé

Comme mentionnés ci-dessus, plusieurs moyens sont possibles pour traiter le problème identifié, des plus simples aux plus efficaces. La principale différence se trouve dans la position de la reconstruction d'états et de la liaison des données. Pour évaluer la pertinence de

Solution :	Reconstruction d'état	Compatible temps réel	Généricité de l'approche	Complexité de conception
Transfert direct	Non	Oui	Non	Très basse
Transfert direct avec historisation	Oui	Non	Non	Basse
Observateur générique	Oui	Oui	Non	Élevée
Observateur orienté SAD	Oui	Oui	Oui	Très Élevée

Tableau 1 : Comparaison des solutions

mise en œuvre d'une solution sur un cas réel, quatre facteurs peuvent être comparés:

- Le besoin du choix d'un SAD avec des variables appartenant à  $S_{unobs}(t)$ , et donc le besoin de reconstruction d'états;
- Le respect des exigences temps réel du SAD à court terme;
- La généricité de l'approche;
- La complexité de conception de la solution.

Le tableau 1 classe, pour chacune des solutions proposées, le niveau de ces facteurs.

Le niveau de ces facteurs est très dépendant de l'utilisation destinée au SAD. Cependant, il souligne que les solutions utilisant un observateur sont intéressantes. Son désavantage majeur reste la complexité de conception, qui fait que son implémentation est quasi impossible à large échelle. La prochaine section présente des approches de conception proposée pour faciliter la conception des 2 observateurs; générique et orienté SAD.

## 5 APPROCHES DE CONCEPTION

Cette section introduit une approche générique de conception, déclinée en deux versions suivant si l'on veut

construire un observateur générique ou un observateur orienté SAD.

Ces approches sont basées sur l'Ingénierie Dirigée par les Modèles (IDM) et ont l'originalité de systématiser une conception globale, incluant la commande et le re-construteur d'états. Cela facilite le travail du concepteur, car le plus d'étapes possible sont automatisées; cela en outre permet une réduction des erreurs.

### 5.1 Concevoir un observateur générique

La première approche a été présentée dans (Adam *et al.*, 2011). Le but des auteurs est de créer un flow de conception, présenté figure 8, automatisant la génération du code de commande et de l'observateur.

Ce flow permet à l'utilisateur final de générer un observateur personnalisable, avec les liens avec la commande déjà établie, seulement basée sur une description UML du système transitaire, comme décrit dans (Lallican *et al.*, 2007).

L'utilisation du même formalisme que (Lallican *et al.*, 2007) est intéressant, car il permet la réutilisation de ses résultats sur la génération de code. Cette génération

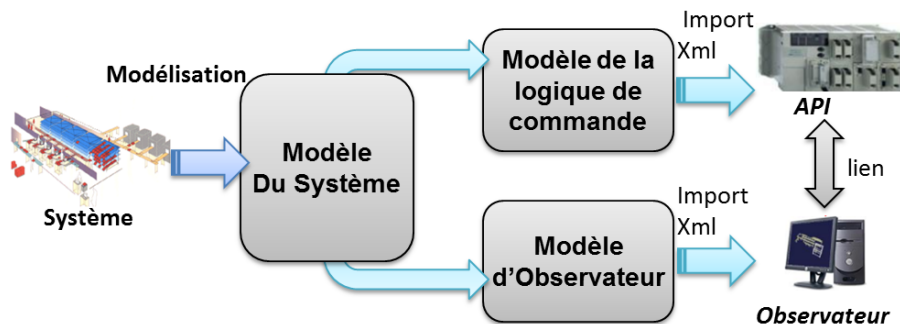


Figure 8 : Flow de conception de l'observateur générique

parallèle est le point clef pour un établissement aisé des liens entre la commande et l'observateur.

Considérant l'exemple de la figure 2, le concepteur doit dans un premier temps modéliser le système transiatique (figure 9). Ce modèle contient toutes les relations entre les composants, leurs positions sur le convoyeur, la tâche qu'ils réalisent et tous les détails sur l'implémentation qui peut être utilisée (comme la technologie des API et le langage qui est souhaité par exemple).

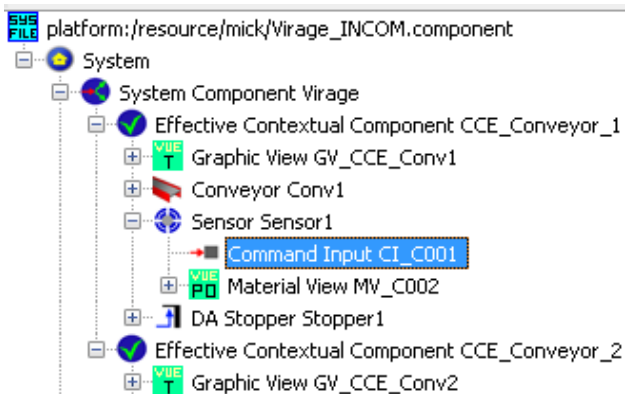


Figure 9 : Modèle base composant de la station de travail

La figure 10 montre le modèle de la commande, contenant tous les composants de bases et les détails sur les cartes d'entrée sortie.

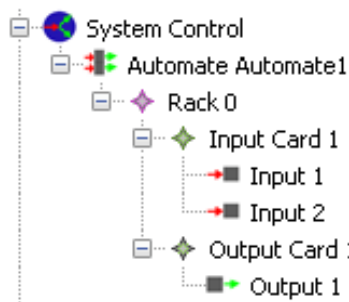


Figure 10 : Modèle hardware de la commande

Le gros avantage de cette solution, imaginée au départ pour un intégrateur d'installation industrielle, est de systématiser, automatiser et accélérer l'implémentation par le concepteur sur le site du client. En outre ce concepteur a seulement à apprendre à modéliser le type de système qu'il implémente, sans besoin de connaissance sur la génération de code.

Cependant, une longue part du travail est réalisée hors ligne, lors d'une phase précédente, par un spécialiste de la génération de code. Ce spécialiste, appelé modélisateur, a en charge la conception des outils utilisés par le concepteur (comme ceux montrés figure 9 et 10) et les transformations de modèles pour la génération du code.

Les composants qu'il crée doivent être génériques pour pouvoir être réutilisés. Cette réutilisabilité est cruciale : en effet cette approche donne de meilleurs résultats à

chaque fois que les outils développés pour une application sont réutilisés dans une nouvelle implémentation.

L'utilisation de cette approche de génération parallèle du code de commande et de l'observateur est très utile pour établir les liens nécessaires entre l'observateur et la commande nécessaire à la synchronisation. Ainsi, le problème de liaisons des données est réduit, étant donné que les primitives de la commande sont toujours codées de la même manière. Cependant, la liaison des données n'est pas totalement résolue, il reste en effet à établir le lien avec le SAD.

De plus, un autre désavantage tient à la genericité de l'observateur : comme il tente d'être synchronisé avec l'état entier du système, le modèle de l'observateur, bien que personnalisable, reste très complexe, et donc difficile à appréhender.

## 5.2 Concevoir un observateur orienté SAD

Cette section tente de montrer comment le flow présenté par (Adam *et al.*, 2011) peut être modifié afin de faire face à deux des inconvénients qui ont été montrés dans la section précédente. L'idée est de modéliser le SAD (dont le résultat est finalement proche du contenu du sous-ensemble  $S_{needed}(t)$ ) dans les premières phases de la modélisation, de sorte que ses caractéristiques peuvent être utilisées lors de la génération basée sur la transformation de modèle (figure 10).

Les composants définis par le modélisateur sont instanciés par l'utilisateur. Plusieurs vues sont attachées à chaque composant. Chaque vue contient les paramètres du système, groupé selon leurs caractéristiques (Lallican *et al.*, 2007). Pour inclure les données du SAD, une nouvelle vue doit être créée. Cette vue est remplie avec les données de  $S_{system} \cap S_{needed}$  nécessaire à l'initialisation du SAD. Cet ensemble de données peut être utilisé pour orienter l'observateur qui ainsi n'observe plus que les données nécessaires et donc diminue par rapport à la forme générique.

Évidemment, le contenu de la nouvelle vue devient dépendant du SAD, cette vue devant être enrichie avec la description de chaque nouveau SAD voulant être implémenté.

## 5.3 Conclusion sur les approches de conception

Si la seconde approche est globalement plus efficace, la difficulté pour le concepteur est généralement la modélisation des données nécessaire au SAD. Cependant, cela est relativement proche de la modélisation du système qu'il réalise également.

Le modélisateur fait face à une grande difficulté. En effet, l'observateur, qui est un modèle de simulation, est classiquement généré par l'utilisation de templates. La



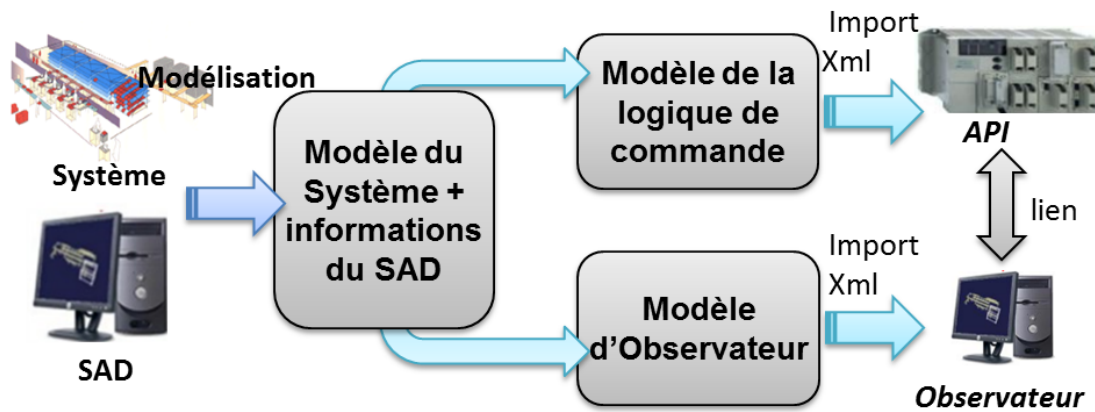


Figure 11 : Génération de l'observateur et de la commande

génération basée sur les composants rend cohérente l'utilisation des templates de simulation, étant donné que presque chaque composant peut être identifié comme un template spécifique. La définition de template est la tâche la plus difficile de l'approche de l'observateur générique.

Le cas de l'observateur orienté SAD nécessite souvent des templates dédiés, étant donné que les fonctions d'observation de chaque template ne sont pas forcément compatibles avec les templates correspondants avec d'autres SAD. Cela force le modélisateur à reconstruire autant de bibliothèques de templates que de vue SAD modélisée. Cependant, le template est généralement plus simple, étant donné que ses fonctions sont moins complexes grâce à l'approche composant.

## 6 CONCLUSION ET TRAVAUX FUTURS

Le but de ce papier est de formaliser et de suggérer des solutions pour le traitement des données entre un système de production et un SAD couplé. Les solutions proposées ont été conçues pour considérer plusieurs implémentations de systèmes relativement proches. Le type de SAD étudié, ont été délimité par des critères de besoin dans leur implémentation, avec une proposition de solution par chaque type de SAD rencontré.

Certaines de ces solutions (basé sur l'utilisation d'un observateur) apportent de réelles solution aux problèmes rencontrés mais souffre d'une complexité de conception accrue. Des procédures de conception ont alors été mise en place pour aider le travail du concepteur. Les procédures ont été autant que possible conçues pour la simplicité et la réutilisabilité. Bien sûr, la conception d'une solution simple pour le concepteur en charge de l'implémentation nécessite une charge plus élevée pour le modélisateur dans la préparation des procédures.

La procédure pour générer un observateur générique a été présentée, avec ses avantages et ses défauts. La procédure de génération d'un observateur dédié a quant à elle été suggérée. La difficulté majeure pour ce type d'observateur réside dans le fait de devoir modéliser les

informations du SAD dans le modèle générique de départ.

Ce travail est destiné à être appliqué avec un intégrateur industriel. L'évaluation de la performance de ces solutions sera effectuée avec les concepteurs et modélisateurs réels afin de repérer les inconvénients de ces approches.

## RÉFÉRENCES

- Adam, M., O. Cardin, P. Berruet and P. Castagna, 2011. Proposal of an Approach to Automate the Generation of a Transitive System's Observer and Decision Support using MDE. *IFAC World Congress*.
- Cardin, O., 2007. Contribution of online simulation to production activity control decision support - application to a flexible manufacture system. Phd Thesis, Université de Nantes, Nantes.
- Cardin, O. and P. Castagna, 2009. Using online simulation in Holonic manufacturing systems. *Engineering Applications of Artificial Intelligence*, 22 (7), pp.1025-1033.
- Chong C.S., A.I. Sivakumar and R. Gay, 2003. Simulation-based scheduling for dynamic discrete manufacturing. *Winter Simulation Conference*.
- Lallican, J.L., P. Berruet, A. Rossi and J-L. Philippe, 2007. A component-based approach for conveying systems control design, *IFAC ICINCO*, 9 (12), pp. 329-336.
- Magdy, M. Abdelhameed and H. Darabi, 2009. Neural network based design of fault-tolerant controllers for automated sequential manufacturing systems, *Mechatronics*, 19, pp. 705-714.
- Mahdavi, I. and B. Shirazi, 2010a. A Review of Simulation-based Intelligent Decision Support System Architecture for the Adaptive Control of

Flexible Manufacturing Systems. *Journal of Intelligence*, 3 (4), pp. 201-219.

- Mahdavi, I., B. Shirazi and M. Solimanpur, 2010b. Development of a simulation-based decision support system for controlling stochastic flexible job shop manufacturing systems. *Simulation Modelling Practice and Theory* 18, pp. 768–786.
- Mok, P.Y., 2009. A decision support system for the production control of a semiconductor packaging assembly line. *Expert Systems with Applications* 36, pp. 4423-4430.
- Pierreval H., 1992. Training a neural network by simulation for dispatching problems. *Proceedings of the Third Rensselaer International Conference on Computer Integrated Engineering*, pp. 332-336.
- Yang, X., H. Kumehara and W. Zhang, 2009. Back Propagation Wavelet Neural Network Based Prediction of Drill Wear from Thrust Force and Cutting Torque Signals. *Computer and information Science*, 2 (3), pp. 75-86.