



**HAL**  
open science

## Proposition d'un environnement d'évaluation pour la mise en oeuvre d'un pilotage par le produit

Rémi Pannequin, André Thomas, Gérard Morel

► **To cite this version:**

Rémi Pannequin, André Thomas, Gérard Morel. Proposition d'un environnement d'évaluation pour la mise en oeuvre d'un pilotage par le produit. *Journal Européen des Systèmes Automatisés (JESA)*, 2009, 43 (4-5), pp.487-511. 10.3166/jesa.43.487-511 . hal-00389097

**HAL Id: hal-00389097**

**<https://hal.science/hal-00389097>**

Submitted on 28 May 2009

**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.

---

# Proposition d'un environnement d'évaluation pour la mise en œuvre d'un pilotage par le produit

**Rémi Pannequin — André Thomas — Gérard Morel**

*Centre de Recherche en Automatique de Nancy (CRAN)*

*CNRS UMR 7039, Nancy Université*

*{Remi.Pannequin, Andre.Thomas, Gerard.Morel}@cran.uhp-nancy.fr*

---

*RÉSUMÉ. Les exigences croissantes en matière de maîtrise de la variété des produits, dans un environnement de plus en plus incertain, ont conduit à formuler les approches de contrôle par le produit. Cependant, le développement tant scientifique qu'industriel de ce concept nécessite des outils d'évaluation permettant l'expérimentation, la validation et la réalisation de démonstrations à destination des industriels et ce, en tenant compte de la dimension et de la complexité de ces contextes. Cet article présente une méthodologie d'évaluation accompagnant le projet de pilotage par le produit au cours de ses phases successives, et spécifie un environnement permettant sa mise en œuvre. Enfin, quelques applications permettant de la valider seront présentées.*

*ABSTRACT. Increasing requirements on the management of product variety, in a more and more volatile production environment, lead to consider product-driven control approaches. However, scientific or industrial development of this concept requires evaluation tools that enable experimentation, validation and proof-of-concept, taking into account the industrial scale and complexity. This article presents an evaluation methodology that can be used at the various stage of a product-driven control project, and define a framework to implement this methodology. Finally, a couple of applications will be presented.*

*MOTS-CLÉS : pilotage par le produit, émulation, simulation, évaluation, système multiagent*

*KEYWORDS: product-driven control, emulation, simulation, evaluation, multi-agent system*

---

## 1. Introduction

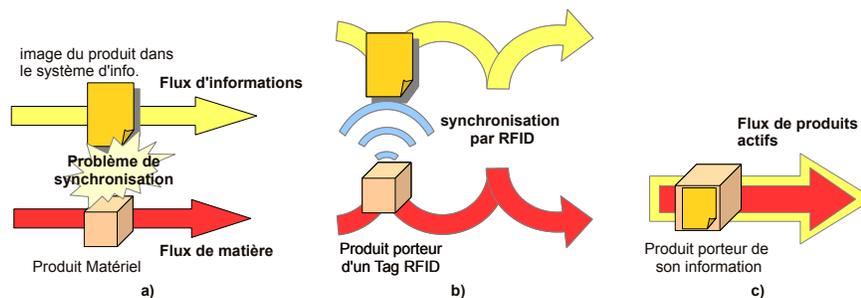
### 1.1. Le concept de contrôle par le produit

Les nouvelles exigences de personnalisation de masse des produits (*mass-customisation*) et de maîtrise d'un environnement devenu très changeant et incertain (raccourcissement du cycle de vie des produits, etc.) contraignent les entreprises à repenser leurs systèmes d'information, pour plus de flexibilité et d'adaptabilité.

Un large consensus au sein des communautés du contrôle holonique (Babiceanu *et al.*, 2006) considère que cet objectif d'agilité peut être atteint par l'utilisation conjointe des technologies multiagent et des technologies infotroniques. Ces dernières permettent de doter chaque entité de l'atelier (machines, produits, opérateurs...) d'une instrumentation, afin de l'intégrer pleinement au sein d'un système d'information. D'autre part, les technologies multiagent posent les bases formelles du fonctionnement de ces systèmes distribués basés sur la collaboration, et fournissent aussi un outillage précieux pour leur mise en œuvre.

Le *contrôle par le produit* (Morel *et al.*, 2003) est une déclinaison de cette approche distribuée, visant à articuler les diverses activités de l'entreprise autour de produits *actifs* (Mcfarlane *et al.*, 2003), capables d'informer les acteurs qui gravitent autour de lui.

La fonction principale du produit actif est de permettre une synchronisation naturelle des caractéristiques physiques du produit en tant qu'objet concret, avec les attributs du produit en tant que représentation informationnelle. Le produit actif est donc un élément clé dans la résolution du problème de synchronisation entre l'état du système physique à contrôler et l'état des objets informationnels qui lui sont associés (figure 1a), reconnu majeur par Plossl (1985).



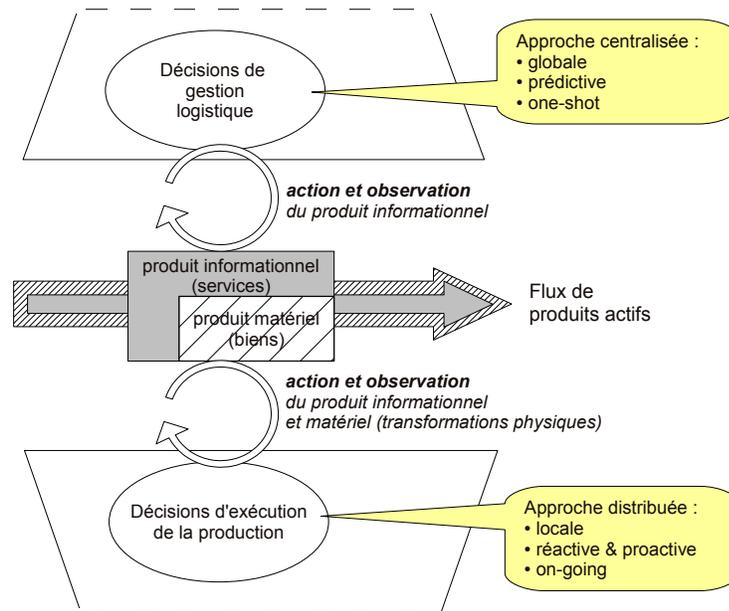
**Figure 1.** a) la problématique de synchronisation entre l'état du système physique et l'état de son image informationnelle ; b) cette problématique conduit à l'apparition de points de synchronisation, par exemple par RFID ; c) dans le concept de produit actif, les parties physiques et informationnelles du produit ne font plus qu'un

L'identification automatique, réalisée par exemple grâce à des puces à lecture sans contact (tags RFID), permet en partie de répondre à cette problématique de synchronisation, le produit étant observé en un certain nombre de points de synchronisation (figure 1b). Toutefois, la progression de ces technologies nous permet d'envisager conceptuellement un produit constamment intégré au système d'information, capable d'informer sur ses caractéristiques physiques, d'utiliser des protocoles de communication complexes, de mémoriser et de traiter des informations (figure 1c).

On assiste donc à une automatisation de plus en plus grande des fonctions d'observation et d'identification des produits. Cette automatisation croissante ne représente pas *en elle-même* une rupture fondamentale, dans la mesure où il n'existe pas de *différence de nature* entre l'identification automatique des produits grâce à de nouvelles technologies et une identification manuelle (par exemple une feuille de suivi adjointe au produit, qui serait lue et annotée par les opérateurs humains en fonction de l'état du produit). Cependant, les *différences de niveau* dans la rapidité, la facilité ou la précision d'identification peuvent induire de profonds changements dans l'organisation du système de contrôle de la production. Le principe de cette nouvelle organisation est de mettre le *produit actif* au centre de l'entreprise. En effet, le produit est un objet commun, partagé et vu sous différentes facettes par tous les services de l'entreprise. Le principe de l'architecture proposée est de coordonner les activités de cette multiplicité d'acteurs, *à travers le produit actif*.

Ainsi, du point de vue du pilotage de la production, le produit pourra en particulier servir à faire coexister des approches réactives et locales de la prise de décision (par exemple des initiatives prises par des équipes d'opérateurs responsabilisés), avec des approches plus prédictives et globales (supportées par exemple par des progiciels de gestion intégrés ou des systèmes de planification avancés). Les apports espérés du pilotage par le produit sont la simplification du système d'information, centré sur un objet unique, ainsi que la possibilité d'hybrider les approches centralisées et distribuées de la prise de décision, pour arriver à des systèmes à la fois performants et flexibles (Thomas, 2004).

Le produit actif se place donc à l'interface entre deux manières différentes mais complémentaires d'envisager le pilotage de la production (figure 2). La première, que nous qualifions de *centralisée*, vise à planifier les décisions à prendre, sur une période pouvant être longue, en envisageant le système à optimiser dans sa globalité. Les outils utilisés pour réaliser cette optimisation reposent sur une résolution ponctuelle (*one-shot* (Valckenaers *et al.*, 2003)) du problème, qui aura auparavant été explicitement modélisé. La seconde manière d'envisager le pilotage, que nous qualifions de *distribuée*, consiste à réagir au mieux aux événements se produisant dans l'environnement proche, avec un horizon temporel relativement faible, ce qui n'exclut toutefois pas une certaine proactivité dans la prise de décision (Valckenaers *et al.*, 2005). Les outils utilisés reposent sur une observation constante du système opérant et une activité en continu des fonctions de décision (*on-going concerns*).



**Figure 2.** Le produit actif comme le lien entre les diverses approches de pilotage

### 1.2. La problématique de l'évaluation des systèmes contrôlés par le produit

Cependant, de nombreuses questions restent ouvertes. Ainsi, comme l'ont écrit Marik *et al.* (2007) au sujet des systèmes multiagent : « Il y a encore un long chemin à parcourir avant que ces architectures ne soient adoptées dans l'industrie. » De même, d'après Parunak (1993), l'évaluation des performances d'un système de contrôle novateur est un point crucial pour son adoption industrielle. Certaines difficultés sont d'ordre technique, car les équipements pouvant être embarqués dans les produits sont encore très limités. Les problèmes plus conceptuels ne sont toutefois pas résolus pour autant. Ainsi, on peut par exemple s'interroger sur le choix de la nature des informations devant être embarquées dans les produits, ou encore sur le mode d'interaction des produits avec leur environnement.

Les réponses à ces questions résident sans doute dans la formulation de méthodes génériques de conception des systèmes contrôlés par le produit. Mais cet objectif ne saurait être atteint sans la possibilité d'évaluer la pertinence d'un système de pilotage par le produit en regard d'une situation d'application particulière. Nous avons donc besoin d'un outil :

- pour *expérimenter* afin de formuler et de valider de nouvelles architectures de contrôle par le produit ;

- pour *démontrer* la pertinence des travaux de recherche, et faciliter leur transfert vers l'industrie ;
- pour *valider* les choix lors de la mise en place industrielle d'un pilotage par le produit.

Cet article propose une réponse à ces besoins d'évaluation, sous la forme d'un environnement d'évaluation adapté au pilotage par le produit.

Nous présenterons tout d'abord l'état de l'art sur l'évaluation des systèmes de pilotage distribués de la production, en présentant les particularités du pilotage par le produit. Dans un second temps, nous distinguerons les différentes approches de l'évaluation de pilotage par le produit, et nous proposerons une méthode permettant de les intégrer à chaque phase d'un projet de mise en œuvre industrielle. Ensuite, nous définirons les composants logiciels qui constituent l'environnement d'évaluation. Enfin, nous présenterons leur application à des cas d'étude.

## **2. Evaluation des systèmes de pilotage dits « intelligents »**

### **2.1. Approches d'évaluation par simulation**

Pour évaluer les performances d'un système de contrôle, il est possible de procéder soit analytiquement, soit empiriquement, en expérimentant directement sur l'atelier réel, ou en utilisant un modèle de simulation de celui-ci.

Dans le contexte du pilotage par le produit, l'évaluation par simulation offre de nets avantages par rapport aux deux autres méthodes. En effet l'utilisation de l'atelier réel est très coûteuse et contraignante, puisqu'elle influe sur la production. Cette difficulté peut être contournée en employant une plate-forme d'essais en laboratoire. Cependant, ces plates-formes, qui restent coûteuses et longues à mettre en place, sont nécessairement beaucoup plus simples que les cas d'études industriels, alors que c'est justement la maîtrise de la complexité qui est le cœur du problème. De plus, les conditions expérimentales sur un système réel (atelier ou plate-forme de laboratoire) sont rarement maîtrisables (en particulier, l'occurrence des perturbations n'est pas contrôlable) ce qui rend difficile la comparaison des cas et l'étude de comportements spécifiques.

De la même manière, une étude analytique est contrainte par les limitations de la théorie, et un cas réaliste est souvent trop complexe pour pouvoir être résolu. Enfin, les formalismes permettant d'estimer le comportement d'un système distribué sont rares. La simulation apparaît donc comme l'une des voies principales permettant l'évaluation d'un pilotage distribué dans un tel contexte.

### **2.2. Vers des environnements de benchmarking**

Le point de départ des travaux d'évaluation par simulation consiste à comparer deux types de pilotage pouvant être antagonistes, typiquement le système proposé et

le système existant, sur un cas d'étude simulé. Cavalieri *et al.* (2000) ont comparé une coordination décentralisée, s'inspirant des mécanismes de marché, avec une autre hiérarchique (basée sur un agent superviseur), en utilisant le modèle de simulation d'un cas simple. Par ailleurs, des méthodes de pilotage prédictives et réactives ont aussi été comparées, à l'aide d'un dispositif expérimental modulaire (Brennan, 2000). Ces premiers travaux se basaient sur des cas d'études particuliers, les modèles d'essais étaient difficilement réutilisables. De plus, les cas d'applications modélisés étaient souvent assez simplistes.

Des travaux ont donc été entrepris pour augmenter la généralité des modèles, afin de fournir à l'ensemble de la communauté de recherche une base d'évaluation homogène. C'est sur ces objectifs qu'a été fondé le quatrième groupe d'intérêt spécifique du réseau d'excellence IMS (IMS-NoE SIG4). Il s'est consacré à la définition et au développement d'un service de benchmarking (Cavalieri *et al.*, 2003 ; Valckenaers *et al.*, 2006) ayant pour but de fournir un environnement de modélisation ainsi qu'une bibliothèque de cas de tests accessible mondialement.

Malgré le succès mitigé du groupe, un tel service est hébergé, et continue d'être développé, par l'université catholique de Louvain<sup>1</sup>. Par ailleurs, la problématique de la simulation des systèmes complexes a été abordée (Monch, 2007), ce qui a conduit à proposer une architecture logicielle assez similaire à celle évoquée par les membres du SIG4. Les travaux de notre équipe sont inscrits dans ce contexte et ont permis d'apporter quelques éléments répondant à cette problématique (Pannequin *et al.*, 2009), (Klein *et al.*, 2006), (El Haouzi *et al.*, 2008).

### 2.3. Architecture des environnements d'évaluation

On constate donc une convergence des architectures de ces dispositifs vers une forme commune, consistant à séparer le ou les systèmes de pilotage à tester, du cas industriel servant de base aux différents tests (Thierry *et al.*, 2008).

Cette architecture utilise la notion d'*émulation*. Celle-ci consiste à remplacer un composant concret d'un système par un modèle informatique capable de reproduire de manière transparente son comportement. L'émulation est utilisée de longue date en génie automatique, pour tester les programmes des automates avant leur mise en production. Pour l'évaluation des systèmes de pilotage, l'émulation consiste à utiliser un *atelier virtuel* pour lui appliquer le système de pilotage.

L'émulation implique trois entités. Il s'agit du système de contrôle  $SC$ , du système opérant  $SO$  devant être contrôlé, et d'un objectif de performance  $P$ . L'évaluation consiste alors à mettre en relation  $SC$  et  $SO$ , pour vérifier que l'objectif de performance  $P$  est atteint.

Le système de contrôle et le système opérant peuvent apparaître soit sous leur forme réelle  $SO_r$  et  $SC_r$ , soit comme modèles (on les désigne alors comme  $SO_m$

1. <http://www.mech.kuleuven.be/benchmarking/>

et  $SC_m$ ). Il existe donc quatre modes d'expérimentation, qui résultent des diverses combinaisons (Pfeiffer *et al.*, 2003) :

- ( $SO_r, SC_r$ ) l'expérimentation consiste à déployer le système de pilotage au système opérant réel ; c'est le cas le plus classique ;
- ( $SO_r, SC_m$ ) un modèle du système de pilotage est appliqué au système réel. Cette configuration peut être employée pour le prototypage du système de pilotage, par exemple en pilotant manuellement l'atelier pour faire apparaître les règles de bon comportement ;
- ( $SO_m, SC_r$ ) le système de pilotage réel est appliqué à l'atelier virtuel ;
- ( $SO_m, SC_m$ ) le système de pilotage et le système opérant sont tous deux présents sous forme de modèle.

#### **2.4. Particularités du pilotage par le produit**

L'expérimentation sur des modèles peut cependant être mise en doute ; en effet, la modélisation peut introduire un certain biais, puisqu'elle fait – par nature – abstraction de nombreux aspects de la réalité.

Le problème de la validation des modèles a été posé de nombreuses fois dans le domaine de la simulation, mais il apparaît un peu différemment dans le cadre de l'émulation. En effet, l'approche traditionnelle de création des modèles de simulation conduit à une imbrication entre le système de contrôle et le système à contrôler (ainsi, dans le logiciel Arena, une entité peut représenter à la fois le produit physique qui va être transformé, l'événement qui va déclencher la transformation, et, de par ses attributs, les informations qui vont la paramétrer), ainsi qu'à des simplifications (par exemple supposer qu'une file d'attente est infinie). Dans l'approche d'émulation, il y a une correspondance beaucoup plus importante entre les équipements et leurs modèles d'émulation, ce qui facilite la validation, puisque chaque composant du modèle peut être validé unitairement. La question : « Ce qui est représenté dans le modèle correspond-il à la réalité ? » est donc plus facilement tranchée, mais fait apparaître la question plus fondamentale : « Le modèle comporte-t-il effectivement tous les aspects de la réalité ayant une importance par rapport au pilotage ? »

Le contexte du pilotage par le produit nécessite d'abord une complète neutralité du système d'émulé vis-à-vis du contrôle, ce qui permet la comparaison d'une approche dans laquelle le produit est actif, avec une approche où il ne l'est pas. Il nécessite aussi de tenir compte de toutes les possibilités d'évolution des produits dans le modèle d'émulation. Cela inclut de représenter aussi bien le routage du produit que la flexibilité de son élaboration (gamme alternatives).

### 3. Méthode incrémentale d'évaluation

L'environnement d'évaluation que nous proposons permet d'assister le praticien à chaque phase d'un projet de mise en place d'un pilotage par le produit. Nous distinguons, selon un découpage assez classique, trois phases successives dans un tel projet : définition, développement, déploiement.

Comme nous allons le montrer, ces phases correspondent respectivement à des problématiques conceptuelles (évaluation des algorithmiques), logique (évaluation des logiciels) ou technologiques (évaluation des équipements).

Cette décomposition permet de réduire la complexité de chaque étape : le système de pilotage est d'abord défini conceptuellement, en faisant abstraction des aspects techniques. Ceux-ci sont progressivement pris en compte, permettant de réduire la complexité de chaque étape pour mieux la maîtriser.

#### 3.1. Définition : évaluation des algorithmes

D'abord, le projet passe par une phase de *définition*. Il s'agit de définir les informations devant être associées aux produits, ainsi que la manière dont ces informations sont initialisées et exploitées par les divers acteurs du pilotage.

Dans cette phase du projet, on manipule donc des règles de décision, qui peuvent être locales ou globales. Les règles globales correspondent généralement à des processus d'optimisation du système, qui s'exécutent typiquement en début de période, et influent sur l'ensemble du système opérant (par exemple un ordonnancement). A l'inverse, les règles locales ont un impact limité à un équipement, et sont déclenchées par des événements relatifs à cet équipement ou son environnement proche. Il s'agit par exemple d'algorithmes permettant de décider d'un changement de série sur une ligne, ou de décider de diriger un produit vers l'une ou l'autre des ressources pouvant l'admettre.

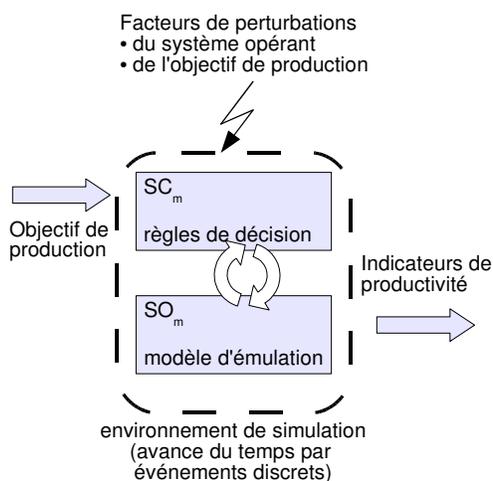
Dans le contexte du pilotage par le produit, il faut de plus déterminer précisément la localisation des points d'observation des produits et leur nature (simple détection d'une présence, détermination du type de produit, identification...). De plus les structures de données associées aux produits devront être définies, ainsi que les comportements que ces derniers exécutent éventuellement.

A ce stade, le concepteur se pose la question de la transformation des informations venant des capteurs en commandes adressées aux actionneurs, sans chercher à déterminer l'architecture logicielle qui devra les supporter. Les problématiques spécifiquement abordées sont donc de nature *algorithmique*.

La pertinence des divers algorithmes de décision pouvant être mis en place dépend de leur impact sur la productivité du système. Celle-ci peut être mesurée en évaluant le flux sortant (*throughput*). Cependant, d'autres indicateurs tels que le délai global, le

taux de service client, l'occupation des ressources ou les niveaux d'en-cours peuvent aussi être considérés pour estimer plus finement le comportement du système.

Le dispositif d'expérimentation doit permettre à ce stade une formulation rapide et facile des algorithmes de décision. De plus, pour pouvoir réaliser de nombreuses expériences, et pouvoir prendre en considération de longues périodes, l'émulation doit être plus rapide que la réalité. Pour ces deux raisons, le mode expérimental ( $SO_m, SC_m$ ) est le plus adapté. En effet, le fait d'utiliser deux modèles facilement intégrables dans un même environnement de simulation permet de réaliser facilement des simulations à événements discrets, qui permettent la simulation de longues durées (figure 3).



**Figure 3.** Configuration du dispositif expérimental au cours de la phase de définition

De plus, la formulation du système de pilotage comme un modèle permet de s'abstraire de nombreux problèmes de développement, ce qui facilite le choix du meilleur mode de pilotage, grâce à l'intégration du modèle d'émulation et des règles de décision au sein d'un même environnement.

### 3.2. Développement : évaluation des logiciels

La phase précédente a permis d'obtenir un ensemble d'algorithmes de décision validés. Sur la base de ces algorithmes, il faut maintenant *développer* le logiciel de pilotage.

La phase de développement se focalise sur les problématiques de stockage et de transfert de l'information, dont on avait fait abstraction dans la phase précédente. Elle permet aussi de tenir compte des délais nécessaires à la prise de décision. Ces aspects peuvent en effet conduire à une désynchronisation entre l'état réel de l'atelier et la

vision qu'en a le système de pilotage. Il est donc particulièrement intéressant d'évaluer la pertinence du pilotage par le produit par rapport à ce risque de désynchronisation. L'expérimentation porte donc sur des problématiques *logicielles*.

En effet, pour un ensemble donné de règles comportementales, il existe très souvent plusieurs manières de développer le logiciel de pilotage par le produit. Ainsi, il est possible d'utiliser des approches orientées agent, en utilisant divers protocoles pour réaliser le transfert de la bonne information au bon acteur. De même, il est possible de projeter les comportements distribués vers une architecture plus centralisée reposant sur une base de données commune. Il faut donc évaluer la pertinence des choix architecturaux. Deux qualités principales sont requises : d'une part, la facilité de développement et de maintenance du code (capacité à s'adapter aux changements à long terme) et, d'autre part, la performance du logiciel, en termes de charge des ressources informatiques, ainsi qu'en termes de robustesse face aux divers types de perturbations.

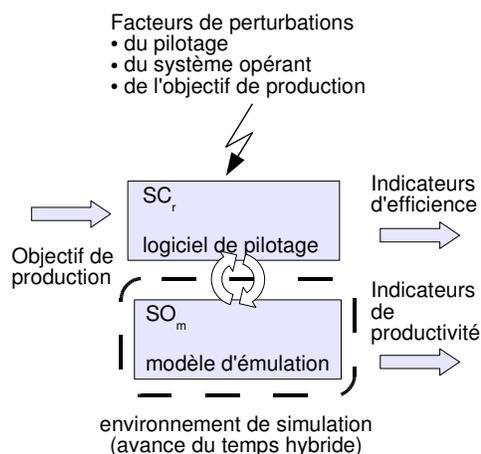
La facilité d'écriture et de maintenance du code est, par sa nature, subjective, et donc très difficile à évaluer objectivement ; on se focalise alors sur des indicateurs mesurant l'efficacité du logiciel de pilotage. Ce type d'indicateurs est beaucoup moins usuel que les indicateurs de productivité. Cependant, une mesure de base pourrait être la quantité de données devant être transmises entre chaque acteur, la taille des bases de données (au sens large) utilisées, ou encore les délais de prise de décision.

Pour pouvoir expérimenter sur ce type de problématiques, le logiciel réel doit être utilisé. On se trouve donc dans la situation  $(SO_m, SC_r)$ , dans laquelle le système de pilotage réel est connecté à l'atelier émulé (figure 4). De plus, pour une prise en compte des délais de prise de décision, l'émulation doit se dérouler en « temps réel », c'est-à-dire être synchronisée sur l'horloge de l'ordinateur exécutant l'émulation. Cependant, il est possible d'accélérer le déroulement des expériences, en utilisant un mode hybride d'avancement du temps (Saint Germain *et al.*, 2003). Celui-ci repose sur une alternance entre une gestion du temps par événements discrets, lorsque le pilotage est en attente, et une gestion en temps réel, lorsqu'une décision est en cours.

### 3.3. Déploiement : évaluation des technologies

La dernière phase du projet est relative au déploiement effectif du pilotage par le produit dans un atelier. En effet, les précédentes étapes avaient fait l'hypothèse d'un produit *actif* dont les états informatique et physique pouvaient être synchronisés. Il faut donc effectivement choisir une instrumentation du produit pour réaliser cette synchronisation.

Il nous faut, à cette étape, tenir compte des caractéristiques réelles des équipements instrumentant le produit, du point de vue de leur fiabilité, de leur rapidité, de l'étendue de leur champ de perception, ainsi que de l'impact des conditions particulières de l'atelier (perturbations électromagnétiques, agressions thermiques, mécaniques ou chimiques, poussières...) sur ces paramètres de fonctionnement.



**Figure 4.** Configuration du dispositif expérimental au cours de la phase de développement

L'émulation ne semble pas la méthode adaptée pour représenter ces contraintes. En effet il est difficile d'obtenir des données de fiabilité tenant compte de l'ensemble de ces facteurs. De plus, le modèle d'émulation ne comporte que les limitations *connues* des équipements, mais seule l'utilisation de l'équipement réel permet de faire apparaître les limitations qui n'étaient pas *a priori* connues.

L'expérimentation en phase de déploiement doit être réalisée sur le système opérant réel. Il s'agit par exemple d'un atelier pilote ou encore d'une plate-forme d'essais de laboratoire, qui doit représenter des problématiques de pilotage suffisamment similaires à ce qu'on observe sur le terrain. La transition de l'émulé au réel peut aussi être réalisée en de multiples phases, au cours desquelles coexistent équipements réels et émuls. Par exemple, le procédé de fabrication peut être émulé, tandis que l'architecture de communication est réelle. Il est aussi envisageable de placer une partie du système de production réel dans un environnement émulé.

L'expérimentation sur une telle plate-forme d'essais permet de confronter le logiciel de pilotage par le produit à des perturbations réelles, afin d'évaluer son adaptabilité et sa robustesse.

### 3.4. Mise en œuvre de la méthode d'évaluation

La figure 4 et le tableau 1 présentent de manière synthétique la méthode que nous proposons. Celle-ci repose sur un raffinement progressif des contraintes, et permet d'intégrer l'évaluation à chaque étape du projet, tout en distinguant les problématiques

relatives à la définition des algorithmes de pilotage, de leur implémentation sous forme logicielle, et sous forme technique.

Phase	Mode d'expérimentation	Avance du temps	Éléments étudiés
Définition	$SO_m, SC_m$	par événements discrets	<ul style="list-style-type: none"> <li>– informations associées aux produits</li> <li>– algorithmes de décision</li> <li>– positionnement des points de synchronisation</li> </ul>
Développement	$SO_m, SC_r$	hybride : événements discrets/temps réel	<ul style="list-style-type: none"> <li>– architecture logicielle</li> <li>– mode de collaboration entre pairs</li> <li>– mode de stockage des informations</li> </ul>
Déploiement	$SO_r, SC_r$	temps réel	instrumentation des produits

**Tableau 1.** Récapitulation des étapes expérimentales successives

#### 4. Spécification de l'environnement

##### 4.1. Architecture globale de l'environnement

L'environnement d'évaluation est constitué de deux éléments principaux :

- le premier sert à créer les modèles d'émulation, et à les exécuter suivant les différents modes d'interaction ;
- le second permet de construire facilement un système de pilotage par le produit selon diverses configurations.

Le premier, l'outil de création et d'exécution des modèles d'émulation, est basé sur une architecture à base de composants. En effet, la modélisation repose sur l'utilisation de *modules d'émulation*, dont l'assemblage constitue le modèle. Ces modules sont dotés d'une interface qui leur permet d'interagir avec le système de pilotage (par l'émission de comptes-rendus et la réception d'ordres). La modularité de l'interface permet la connexion avec un système de pilotage intégré à l'émulateur, programmé sous la forme d'un ensemble de fonctions, ou avec un système de pilotage réel, à distance à travers le réseau (figure 6).

Le second est l'infrastructure multiagent de pilotage par le produit. Il s'agit d'un ensemble d'agents, que l'on peut facilement configurer pour adopter diverses configurations de prise de décision.

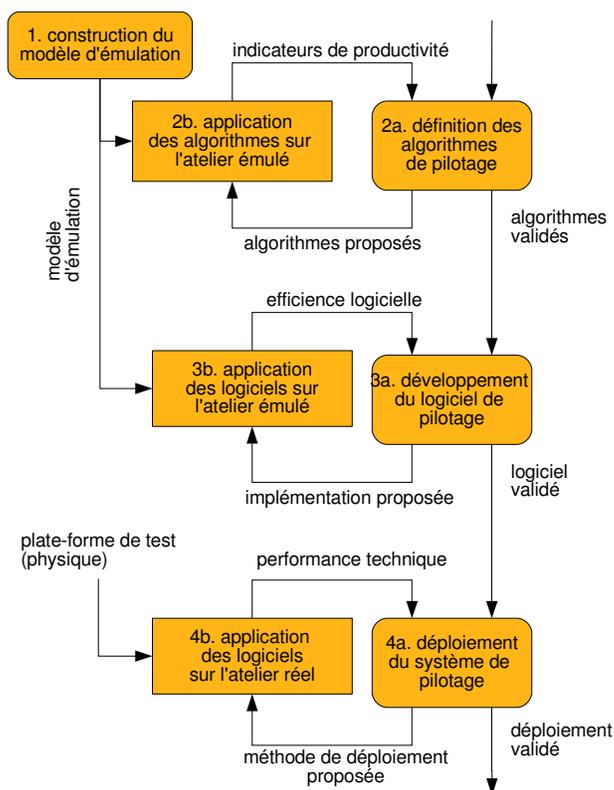


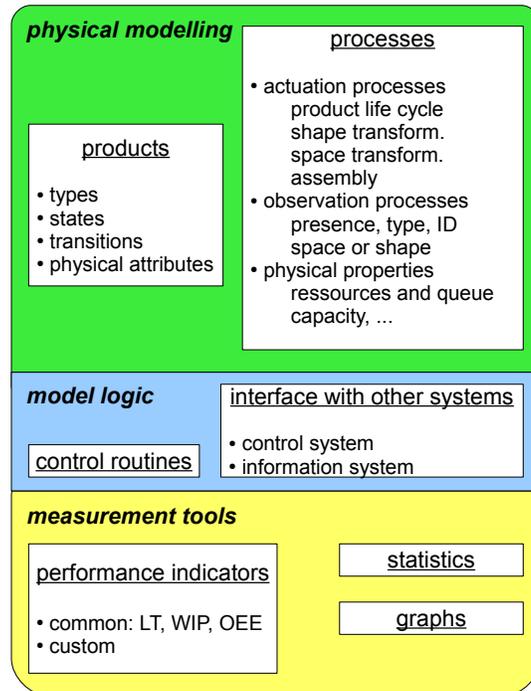
Figure 5. Les étapes de la méthodologie d'évaluation

Ces deux éléments, dont l'implémentation constitue un prototype de l'environnement d'évaluation spécifié, sont présentés dans les parties suivantes.

#### 4.2. Composants orientés-objets pour l'émulation du système opérant

La création du modèle d'émulation est basée sur une vision systémique et processive de l'atelier. En effet, nous nous focalisons sur les produits, et nous cherchons à en représenter l'évolution physique. D'après la systémique, ces évolutions peuvent être vues comme des transformations de forme, d'espace et de temps (Le Moigne, 1977).

Ainsi, la première étape de modélisation est l'analyse de l'atelier, nous permettant de déterminer l'ensemble des états possibles des produits et de les caractériser en termes de position (espace) et de morphologie (forme). Les produits émulés seront caractérisés par ces deux attributs. Nous introduisons donc deux modules d'émulation :



**Figure 6.** Architecture de l'environnement d'émulation, mode de contrôle local ou distant

le premier a pour fonction d'introduire les produits dans le modèle d'émulation, tandis que le second représente les périodes d'attente des produits.

Ensuite, nous représentons les actions sur les produits. Il s'agit de modéliser les équipements de l'atelier en fonction des transformations d'espace et de forme qu'ils peuvent effectuer sur les produits. Pour assurer l'indépendance du modèle d'émulation par rapport au système de pilotage, toute transformation physiquement possible doit être représentée. Pour représenter l'action sur les produits, deux primitives de modélisation sont introduites, les actionneurs de forme et d'espace. Ces primitives tiennent compte des contraintes physiques qui régissent l'action, par exemple des contraintes de capacité, de vitesse maximale, de temps de réglage, etc.

Enfin, il nous faut représenter les fonctions d'observation des produits, en tenant compte des caractéristiques d'identification (détection d'une présence, d'un type de produit, d'une instance précise d'un produit), et de la nature de l'observation (observation d'espace, ou perception de la morphologie du produit) et des contraintes physiques d'observation (par exemple un champ de perception limité, ou un délai de traitement). L'observation des produits est réalisée grâce à une primitive d'événement produit.

Un prototype basé sur ces spécifications a été développé comme une extension au logiciel de simulation à événements discrets Arena, en utilisant la capacité de ce logiciel à développer des bibliothèques d'objets spécifiques. Les fonctions de communication avec les modules ont, quant à elles, été réalisées sous la forme d'une bibliothèque de fonctions (dll).

L'utilisation de cette approche systémique nous permet de spécifier un ensemble de module d'émulation à la fois génériques et bien adaptés au pilotage par le produit.

### 4.3. Composants multiagents pour le développement du système de pilotage

Les technologies multiagents sont particulièrement adaptées à l'étude du pilotage par le produit, dans la mesure où elles permettent de tenir compte du fort parallélisme de l'exécution de ces systèmes ; elles forcent aussi à une formulation explicite des transferts d'information entre agents, et permettent d'ailleurs l'analyse de ces échanges. Cependant, malgré l'existence de kits de développement comme par exemple la bibliothèque JADE, la mise au point d'un système multiagent reste difficile et requiert une forte spécialisation. Pour rendre plus accessible le développement d'un système multiagent de pilotage par le produit, nous avons développé un ensemble d'agents pouvant être facilement configurés.

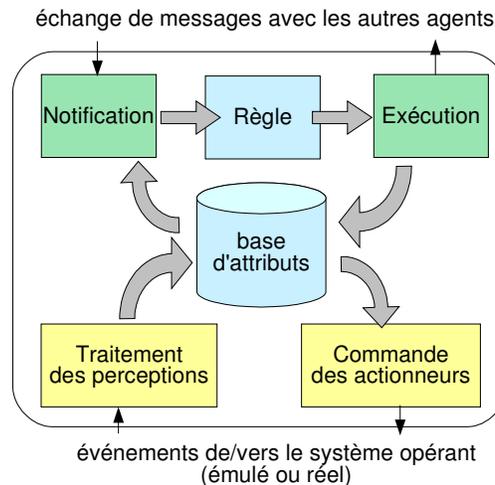
Le fonctionnement du système repose sur l'association d'un *agent*<sup>2</sup> à chaque élément du système opérant, réel ou émulé. Le cas particulier de l'association d'un agent à un produit physique permet de constituer un *produit intelligent*, l'agent pouvant être exécuté dans un équipement embarqué sur le produit, ou sur un ordinateur déporté.

Les agents sont dotés de divers sous-systèmes, permettant en particulier l'interaction avec le système opérant (traitement des perceptions, et commande des actionneurs) ainsi que la communication avec les autres agents. Ces sous-systèmes sont structurés autour d'une base d'attributs, qui transpose l'état de l'entité du système physique auquel l'agent est associé dans le domaine informationnel. Enfin, chaque agent peut être doté d'une règle comportementale qui déclenche des actions en fonction de la valeur d'attributs (figure 7).

Le mode de communication entre les agents est basé sur un système à base de règles par notification. Une première application au pilotage par le produit de tels systèmes a été faite par Simao (2005). Ainsi, lors de leur initialisation, les agents physiques déclarent les attributs qu'ils possèdent à leurs pairs. Les agents voulant être informés de la valeur de certains attributs « s'abonnent », et sont ensuite notifiés lors des changements de valeurs. A la réception d'un message le notifiant de la nouvelle valeur prise par un attribut (par exemple, l'attribut « état » de la ressource M1 devient « disponible »), la règle embarquée dans l'agent est évaluée, et peut produire une action. Les différents types d'actions possibles sont :

2. Dans la mesure où chacun de ces agents est en communication très étroite avec un composant du système physique (produit, ressource...) nous parlerons d'*agents physiques*.

- le déclenchement d’une opération par un pair ;
- l’interrogation sur la valeur d’un attribut ;
- le changement de la valeur d’un attribut d’un pair.



**Figure 7.** Structure des agents associés à des objets physiques

La configuration des agents est réalisée à travers le paramétrage de la base d’attribut (sous forme d’un fichier XML), et de la règle (sous la forme d’une classe java). Le paramétrage du système de pilotage en est donc très nettement simplifié.

## 5. Application

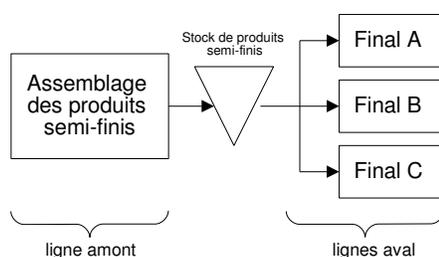
### 5.1. Application à un cas d’étude industriel

L’environnement d’évaluation a tout d’abord été appliqué à un cas d’étude industriel. Ce cas est décrit plus en détail du point de vue du développement d’un système de pilotage par le produit performant chez (Pannequin *et al.*, 2009). Dans cette section, nous nous focaliserons sur les apports de l’environnement d’évaluation dans le processus de développement.

L’entreprise considérée est une filiale de l’industrie automobile, caractérisée par de fort volumes de production, et une forte variété des produits. L’atelier est découpé en une dizaine de cellules de fabrication dédiées à un client particulier. Dans chaque cellule, la production est réalisée en deux étapes (figure 8) :

- d’abord, des produits semi-finis sont assemblés sur une ligne de production ;
- ensuite, l’assemblage final a lieu sur trois lignes parallèles.

La réalisation des produits semi-finis étant environ trois fois plus rapide que l'assemblage final, la cellule est globalement équilibrée. Cependant, le problème de pilotage consiste à séquencer la production des divers ordres à réaliser, ce qui revient à décider des changements de série. Cette prise de décision doit tenir compte des délais de livraison aux clients, tout en minimisant les en-cours de produits semi-finis et en évitant la famine sur les lignes en aval. De plus, le changement de série est une opération relativement longue, et un nombre de changements de série trop important réduit considérablement les capacités de production.



**Figure 8.** Schéma des flux dans la cellule de production étudiée

Comme nous l'avons montré précédemment, ce problème de pilotage peut être résolu selon deux approches :

- d'une part, les ordres de fabrication peuvent être fractionnés grâce à un calcul de taille optimale de lot, puis séquencés en utilisant une heuristique, dans le cas présent un classement en fonction de leur ratio critique. Ce calcul réalisé *hors-ligne* est ensuite mis en œuvre sur le terrain, et éventuellement réexécuté en cas d'aléa important ;
- d'autre part, les décisions peuvent être prises en réaction à l'état du système de production (*en ligne*). Ainsi, des règles de décision simples ont pu être définies en s'inspirant du comportement des opérateurs sur le terrain. Ces règles spécifient le déclenchement d'un changement de série en fonction de variables telles que les niveaux d'en-cours, les taux d'occupation des ressources, etc.

Les expériences menées sur ce cas industriel visent à étudier la pertinence de l'architecture par produits actifs pour réaliser une *hybridation* entre ces deux approches de pilotage. En particulier, nous souhaitons étudier la pertinence d'algorithmes de pilotage par le produit par rapport à un pilotage centralisé classique, dans des contextes plus ou moins perturbés. Le principe de cette hybridation repose sur l'association à chaque produit de données issues du pilotage centralisé. Après cette association, qui a lieu *hors-ligne*, les données associées aux produits sont utilisées lors des prises de décisions concernant les changements de série. Nous appliquons donc en partie l'architecture présentée figure 2, dans laquelle les décisions centralisées sont transcrites en attributs portés par les produits, ces attributs venant enrichir la prise de décision locale.

Trois systèmes de pilotage ont été définis : il s'agissait, d'une part, d'un pilotage centralisé (basé sur une heuristique de ratio critique), nous servant de base de comparaison, et d'autre part, de deux types de pilotage par le produit. Le premier (désigné comme PP-contraint) soumet entièrement les décisions locales à la valeur d'un attribut de *priorité* des produits, ce dernier étant initialisé d'après l'ordonnement centralisé (le classement par priorité permet de retrouver la séquence ordonnée). Le second (PP-autonome) prend en compte, en plus de cet attribut de priorité, les paramètres de fonctionnement locaux, comme les niveaux d'en-cours.

La première étape a été la réalisation du modèle d'émulation de la cellule de production. Celui-ci est relativement simple, l'une des difficultés principales dans la création d'un modèle de simulation étant la modélisation du système de décision (van der Zee, 2006), qui est – par construction – absent d'un modèle d'émulation. Les données d'entrée du modèle sont le plan de production, qui comporte environ 7 000 produits, répartis en 12 références (une semaine de production).

Ensuite, l'environnement d'évaluation nous a permis de valider les algorithmes de décision utilisés. Il s'agit, d'une part, de s'assurer que les règles comportementales définies répondent bien à l'objectif global (par exemple, de détecter et de corriger les comportements qui entraînent d'incessants changements de séries). Il s'agit, d'autre part, de s'assurer que ces règles sont correctement formulées (en éliminant les blocages et défaillances dus à des erreurs de codage).

En plus de cet accompagnement qui a lieu tout au long de la phase de développement, l'environnement d'évaluation nous a permis, en clôture de la phase de définition, de valider le système de contrôle. En effet, une mauvaise programmation des règles de pilotage produirait une baisse artificielle des performances du système testé : nous parlerons d'un *biais d'implémentation*. Pour s'assurer de l'absence de ce biais, nous avons comparé le niveau de performance entre les différentes configurations du système de pilotage, sur un cas *non-perturbé* (tableau 2). On constate, d'après ces résultats, que le pilotage PP-contraint permet – en l'absence de perturbations – de reproduire l'algorithme centralisé. Le pilotage PP-autonome se comporte différemment, mais permet de retrouver un niveau de performance équivalent.

Type de pilotage	Délai global (min)	WIP (unités)	Taux de rendement moyen
centralisé	6985,57	250,78	88,70 %
PP-contraint	6995,53	260,55	88,59 %
PP-autonome	7185,76	141,29	86,23 %

**Tableau 2.** Valeur des divers indicateurs de performance pour chaque mode de pilotage, dans une situation non perturbée, servant de référence

Enfin, l'environnement d'évaluation nous a permis de comparer la performance des trois modes de pilotage en fonction de divers facteurs de perturbation. Deux types de perturbations ont été introduites :

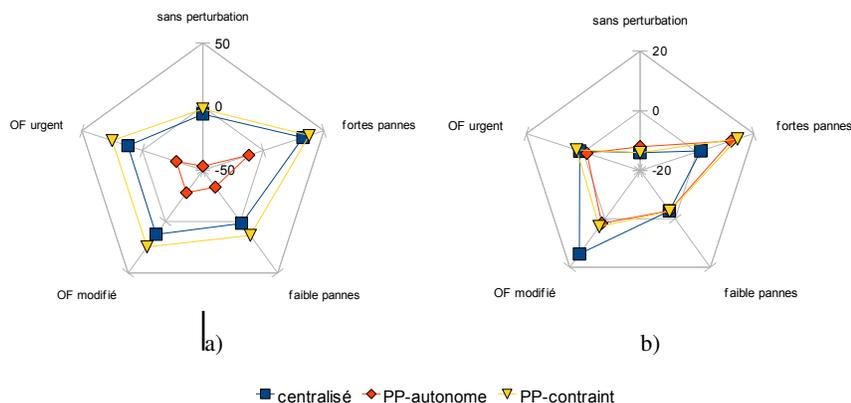
– des perturbations du *process*, qui se traduisent par des indisponibilités des ressources,

– des perturbations du *business*, qui se traduisent par des modifications des ordres de fabrication à réaliser.

Les diverses situations d'expérimentation sont les suivantes : pour chaque type de perturbation, trois modalités ont été considérées. Ainsi, pour les perturbations du process, une situation « faibles pannes » a été modélisée, sur la base de mesures réalisées sur le système industriel réel. Elle sont caractérisées par des indisponibilités courtes se produisant fréquemment. De même, une situation « fortes pannes » a été définie en augmentant la durée des pannes et en réduisant leur probabilité d'apparition. Pour les perturbations du *business*, une situation « OF urgent » a été introduite, dans laquelle un ordre concernant la fabrication d'une petite quantité dans un délai très court est ajouté en cours de production, ainsi qu'une situation « modification d'OF », dans laquelle un OF voit sa quantité augmentée de 10 %.

L'environnement d'évaluation nous a permis de modéliser ces différentes situations, de les combiner afin d'étudier leurs interactions éventuelles, pour évaluer la performance des divers systèmes de pilotage (centralisé, PP-contraint, PP-autonome). Pour s'assurer d'une exploration exhaustive de l'espace de paramètres, et pour faciliter l'étude des interactions entre facteurs, un plan d'expériences  $L_{27}$  (correspondant à trois facteurs ayant chacun trois modalités) a été utilisé. Ces expériences ont nécessité un temps total de simulation de 30 heures, sur un ordinateur de bureau.

La figure 9 présente de manière synthétique l'effet sur les niveaux d'en-cours et sur le délai global de ces différentes perturbations. D'après ces résultats expérimentaux, le pilotage par le produit avec autonomie (PP-autonome) conduit à des niveaux d'en-cours nettement plus faibles que le pilotage centralisé. A l'inverse, le pilotage par le produit sans autonomie (PP-contraint) conduit à une légère diminution des performances par rapport au pilotage centralisé.

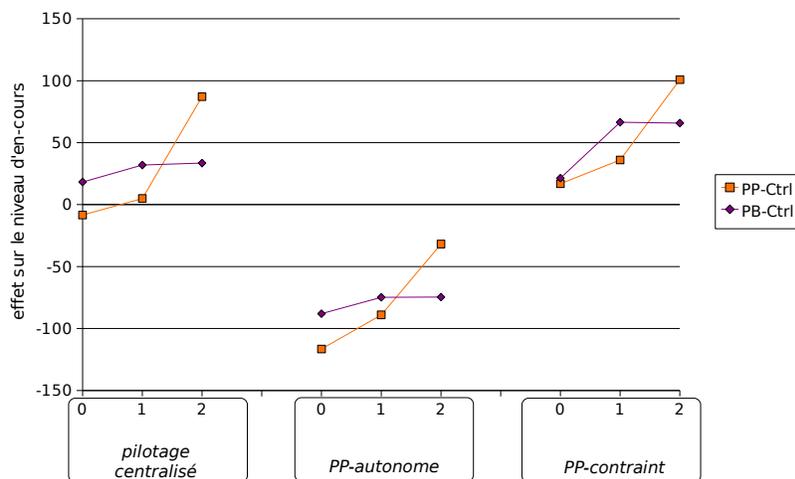


**Figure 9.** a) effet des facteurs de perturbation sur le niveau d'en-cours et, b) sur le délai global

L'étude des effets sur le délai global des interactions entre le mode de pilotage et les perturbations montre que l'effet du facteur de perturbation domine largement l'effet du mode de pilotage, surtout dans le cas de perturbations du process. On peut cependant remarquer que lorsque les perturbations sont les plus importantes (modalités 2), le pilotage PP-autonome s'avère meilleur. De la même manière, le pilotage par le produit PP-autonome est meilleur pour certains types de perturbations du business.

On ne détecte donc pas une amélioration majeure du pilotage par le produit par rapport à la réduction des délais. Toutefois, celui-ci ne dégrade pas non plus les performances (par rapport au pilotage centralisé qui sert de référence) y compris dans une situation non perturbée.

Contrairement aux effets sur le délai global, les effets des facteurs d'interaction sur les niveaux d'en-cours (figure 10) dépendent nettement du mode de pilotage. En particulier, le pilotage PP-autonome conduit à une forte réduction des en-cours, tandis que le pilotage PP-contraint conduit à leur augmentation, surtout en cas de perturbations du business. Ce résultat doit cependant être nuancé, car l'heuristique choisie pour l'ordonnancement centralisé n'est pas focalisée sur une réduction des en-cours. Ces graphiques montrent donc l'apport potentiel du pilotage par le produit par rapport à la réduction des en-cours, ainsi que l'importance de la manière de le mettre en œuvre.



**Figure 10.** Effet sur les niveaux d'en-cours des interactions entre le système de pilotage et les perturbations

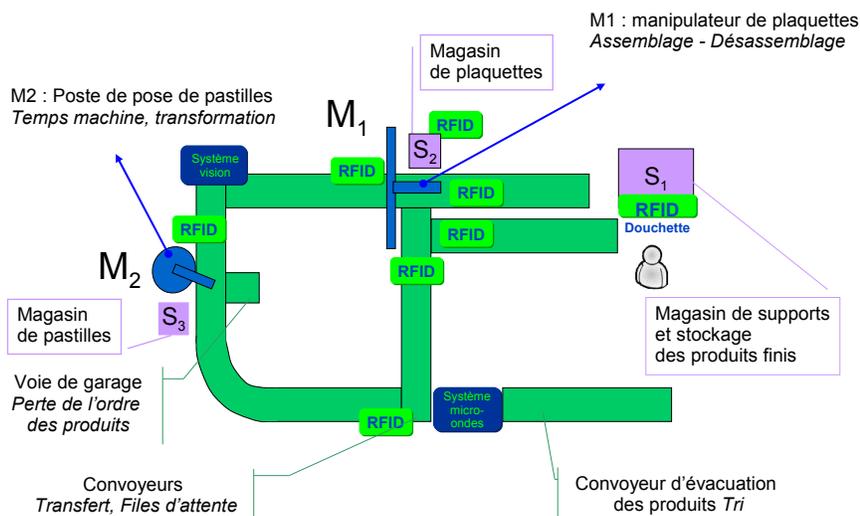
Pour conclure l'étude de ce cas, on peut dégager les apports de l'environnement d'évaluation, qui reprennent les trois objectifs avancés en section 1.2. D'abord, il permet l'expérimentation. En effet, il assiste le développeur tout au long de son travail, en lui permettant de mettre rapidement en œuvre ses idées, afin de valider la correction

du code écrit, de détecter les effets globaux potentiellement imprévus des règles locales, et ainsi d'affiner sa compréhension du système. Ensuite, il permet la *validation* des propositions, en assurant l'absence d'un biais d'implémentation sur la base de résultats objectifs. Enfin, il permet de mener à bien une série d'expériences sur un cas concret d'une échelle réaliste, et permet ainsi de *faire la démonstration* de l'impact du pilotage par le produit ; les résultats quantitatifs pouvant servir de base à de nouvelles réflexions sur le pilotage par le produit.

## 5.2. Application à la cellule d'assemblage Tracilog

La sous-section précédente a montré l'utilité de l'environnement d'évaluation dans la phase de définition du pilotage par le produit. Sur un autre plan, l'expérimentation sur les problématiques logiques et techniques liées au pilotage par le produit est aussi d'un grand intérêt. Dans ce but, des essais ont été réalisés sur une plate-forme de laboratoire. Ceux-ci nous ont permis de valider les composants multiagents proposés, par une application concrète. Ces essais correspondent donc aux phases de développement et de déploiement, présentés aux sous-sections 3.2 et 3.3.

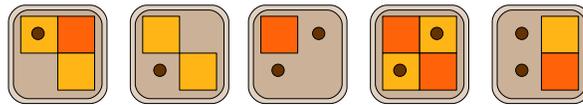
La cellule étudiée est un système automatisé classique mais doté d'une instrumentation RFID (identification par radio-fréquences). Elle comporte deux postes d'assemblage, deux aiguillages, reliés entre eux par un système de convoyage (figure 11).



**Figure 11.** Vue schématique de la cellule d'assemblage Tracilog

L'assemblage des produits sur la cellule d'essais se fait en deux temps : d'abord, de zéro à quatre plaquettes jaunes ou oranges sont déposées sur un support. Ensuite, sont posées de zéro à quatre pastilles. La figure 12 présente quelques exemples de produits

réalisables. Grâce à cette combinatoire, le nombre de produits différents est important (plus de 1 200 produits différents), ce qui permet d'étudier les problématiques liées à la diversité des produits. Après ces deux opérations, les produits peuvent être dirigés vers deux stock différents, ou revenir sur la boucle de production. Au moment de cette phase finale, un désassemblage peut être effectué.



**Figure 12.** *Quelques exemples de produits réalisables*

Un modèle d'émulation de la plate-forme a été créé, et a servi pour une première mise en œuvre du système multiagent. Comme dans le cas précédent, la possibilité de tester le bon fonctionnement du système de pilotage au cours de son développement est appréciable. Ainsi, nous avons pu paramétrer les agents en utilisant des règles simples, puis en complexifiant le système progressivement.

Après cette phase de développement, le système de pilotage a été déployé au système automatisé réel. Ceci a été réalisé en développant, d'une part, un programme de supervision de la cellule automatisée, et d'autre part, un composant spécifique capable de se connecter à ce superviseur. Une fois ces composants spécifiques développés, le « branchement » du système de pilotage à l'émulateur ou à la cellule d'assemblage peut être réalisé très rapidement par un changement de configuration simple. De cette manière, il devient possible de substituer très facilement le système automatisé à son modèle d'émulation.

Lors des expériences de déploiement, nous avons pu constater l'influence de paramètres techniques, comme la fiabilité des lecteurs RFID, ou les délais de transmission des événements. Ces expériences nous permettent d'affirmer la faisabilité technologique du pilotage par le produit, et nous permettent aussi d'évaluer qualitativement les performances de cette approche. Les points forts sont tout d'abord la modularité du système de contrôle qui peut être très facilement étendu ou modifié. De plus, l'approche de pilotage par le produit permet de mettre en place simplement des fonctionnalités avancées de gestion *individuelle* des produits qu'il serait difficile d'obtenir avec un système plus classique. Ainsi, chaque produit du flux peut avoir une gestion (incluant le routage, le suivi de qualité, la notification du client en fin de production, etc.) individuel et indépendant.

Les points négatifs sont principalement la très forte dépendance du système au lien entre agent et produit taggé : si les événements provenant des produits sont perdus, le fonctionnement est gravement compromis. Toutefois, la fiabilité de ce lien peut être augmentée par diverses méthodes. En perspective de ces résultats qualitatifs, le système est actuellement en cours d'amélioration afin d'obtenir des résultats quantitatifs.

Cette application démontre la capacité de l'architecture d'émulation à se substituer de manière transparente au système automatisé réel, et sa pertinence dans les phases de développement du système de pilotage. Nous avons pu aussi constater le bon fonctionnement du système multiagent proposé, de par sa capacité à contrôler un système réel.

## 6. Conclusion

Cet article a proposé un environnement d'évaluation adapté à l'expérimentation, à la validation et à la démonstration du potentiel du pilotage par le produit. Cet environnement a été appliqué à différents modes de fonctionnement, et nous a donc permis de montrer la manière dont ceux-ci peuvent être exploités au cours des différentes étapes d'un projet de mise en œuvre d'un pilotage par le produit.

Ainsi, partant de la création d'un modèle d'émulation de l'atelier, souvent plus facile à réaliser et à valider qu'un modèle de simulation traditionnel, il est possible d'évaluer d'abord les règles comportementales des acteurs d'un pilotage par le produit, et ensuite la manière de mettre en œuvre ces comportements. Enfin, il s'agira de substituer au modèle d'émulation de l'atelier, l'atelier réel.

La méthodologie d'évaluation permet donc de distinguer les aspects algorithmiques, logiciels et techniques d'un projet de mise en place d'un pilotage par le produit. L'environnement proposé facilite de plus les allers et retours entre ces trois phases.

Le prototype développé est ainsi une aide précieuse pour que les réponses aux questionnements sur le pilotage par le produit puissent être soutenues par des mesures expérimentales objectives. Nous prévoyons de continuer son application à notre plateforme d'essais et à de nouveaux cas industriels, afin de constituer petit à petit une bibliothèque de cas de tests.

Le développement de ce prototype est au premier plan de nos perspectives. En plus des tâches directement liées à l'outil (facilité de prise en main, fiabilité, interface utilisateur...), nous aurons aussi à étudier des questions plus fondamentales. Parmi celles-ci, la modélisation du produit actif occupe une place importante. En effet, l'approche développée ici consiste à considérer que les éléments informationnels liés au produit (par exemple les données ou les traitements effectués par un appareil embarqué sur le produit) font partie du système de contrôle. D'autres approches sont envisageables, et devront être étudiées. Plus généralement, la séparation entre le système opérant et le système de contrôle se doit d'être affinée, en particulier dans le cas de modèles récursifs.

## 7. Bibliographie

- Babiceanu R., Chen F., « Development and applications of holonic manufacturing systems : a survey », *Journal of Intelligent Manufacturing*, vol. 17, n° 1, p. 111-131, February, 2006.
- Brennan R. W., « Performance comparison and analysis of reactive and planning-based control architectures for manufacturing », *Robotics and Computer-Integrated Manufacturing*, vol. 16, n° 2-3, p. 191-200, April, 2000.
- Cavalieri S., Garetti M., Macchi M., Taisch M., « An experimental benchmarking of two multi-agent architectures for production scheduling and control », *Computers in Industry*, vol. 43, n° 2, p. 139-152, October, 2000.
- El Haouzi H., Thomas A., Pétin J.-F., « Contribution to reusability and modularity of manufacturing systems simulation models : application to distributed control simulation within DFT context », *International Journal of Production Economics*, vol. 112, n° 1, p. 48-61, 2008.
- Klein T., Thomas A., « A simulation testbed for decision system evaluation in a furniture manufacturing group », *12th IFAC Symposium on Information Control Problems in Manufacturing, INCOM'2006, St-Etienne, France*, vol. 3, p. 573, 2006.
- Le Moigne J. L., *La théorie du système général. Théorie de la modélisation*, Presse Universitaires de France, 1977. réédition de 1994.
- Marik V., Lazansky J., « Industrial applications of agent technologies », *Control Engineering Practice*, vol. 15, n° 11, p. 1364-1380, 2007.
- Mcfarlane D., Sarma S., Chirn J. L., Wong C. Y., Ashton K., « Auto ID systems and intelligent manufacturing control », *Engineering Applications of Artificial Intelligence*, vol. 16, n° 4, p. 365-376, June, 2003.
- Monch L., « Simulation-based benchmarking of production control schemes for complex manufacturing systems », *Control Engineering Practice*, vol. 15, n° 11, p. 1381-1393, 2007.
- Morel G., Panetto H., Zaremba M., Mayer F., « Manufacturing enterprise control and management system engineering : paradigms and open issues », *Annual Reviews in Control*, vol. 27, n° 2, p. 199-209, 2003.
- Pannequin R., Morel G., Thomas A., « The performance of product-driven manufacturing control : an emulation-based benchmarking study », *Computer in Industry*, vol. 60, n° 3, p. 195-203, 2009.
- Parunak H. v. D., MASCOT : A virtual factory for research and development in manufacturing scheduling and control, Technical report, ITI Tech Memo 93-02, 1993.
- Pfeiffer A., Kádár B., Monostori L., « Evaluating and improving production control systems by using emulation », *Applied Simulation and Modelling*, 2003.
- Plossl G. W., *Production and inventory control : principles and techniques (2nd Edition)*, Prentice Hall, March, 1985.
- Saint Germain B., Valckenaers P., Zamfirescu C., Bochmann O., Vanbrussel H., « Benchmarking of manufacturing control systems in simulation », *Proc. of the 3rd Int'l Workshop on Performance Measurement (IfIP WG5.7), Bergamo*, p. 357-369, 2003.
- Simao J., Contribution au développement d'un outil de simulation de systèmes holoniques de production et proposition d'un meta-modèle de contrôle holonique, PhD thesis, Université Henri Poincaré & Centro Federal de Educação Tecnológica do Paraná (CEFET-PR) - Brasil, 2005.

- Thierry C., Thomas A., Bel G. (eds), *La simulation dans les chaînes logistiques*, Hermès Science Publications, 2008.
- Thomas A., « De la planification au pilotage pour les chaînes logistiques », Habilitation à Diriger des Recherches, Université Henri Poincaré - Nancy 1., 2004.
- Valckenaers P., Van Brussel H., « Fundamental insights into holonic systems design », *Lecture Notes in Computer Science : Holonic and Multi-Agent Systems for Manufacturing*, vol. 3593, p. 11-22, 2005.
- Valckenaers P., van Brussel H. B., Karuna H., Bochmann O., Saint Germain B., Zamfirescu C., « On the design of emergent systems : an investigation of integration and interoperability issues », *Engineering Applications of Artificial Intelligence*, vol. 16, p. 377-393, 2003.
- Van der Zee D. J., « Modeling decision making and control in manufacturing simulation », *International Journal of Production Economics*, vol. 100, n° 1, p. 155-167, March, 2006.