

# MODÉLISATION D'UNE CHAÎNE LOGISTIQUE HYBRIDE PAR LA SIMULATION À ÉVÉNEMENTS DISCRETS

Najat Bara<sup>1</sup>, Frédéric Gautier<sup>1,2</sup>

<sup>1</sup>EMINES School of Industrial Management, University Mohammed VI – Polytechnique, Ben Guerir, Morocco

<sup>2</sup>IAE de Paris, Université Paris I Panthéon-Sorbonne Paris, France

[najat.bara@emines.um6p.ma](mailto:najat.bara@emines.um6p.ma)  
[frederic.gautier@emines.um6p.ma](mailto:frederic.gautier@emines.um6p.ma),  
[frederic.gautier@univ-paris1.fr](mailto:frederic.gautier@univ-paris1.fr)

Vincent Giard<sup>1,3</sup>

<sup>3</sup>Université Paris-Dauphine, PSL Research University, Paris, France

[Vincent.Giard@emines.um6p.ma](mailto:Vincent.Giard@emines.um6p.ma),  
[Vincent.Giard@dauphine.fr](mailto:Vincent.Giard@dauphine.fr)

**RÉSUMÉ :** La simulation permet d'évaluer les conséquences des décisions opérationnelles dans une chaîne logistique complexe. Une chaîne logistique hybride (CLH) enchaîne, sans restriction, des processus continus (PC) et/ou discrets (PD). Aucun logiciel de simulation à événements discrets (SED) ou de la dynamique des systèmes (DS) ne permet à lui seul de simuler le fonctionnement d'une CLH au niveau de granularité voulu, restreignant l'usage de la simulation ou conduisant à des mécanismes complexes de différentes approches. La prise en charge de PC par la SED implique la création d'items par une discrétisation périodique de flux émis par certains PC de la CLH. Ces items « continus » (IC) peuvent constituer des inputs d'autres PC et être stockés, ce qui pose plusieurs problèmes : les caractéristiques d'IC sortant d'un PC diffèrent souvent de celles requises par le PC qui l'utilise comme input ; la capacité d'un stock ne peut être définie en un nombre maximal d'IC si ceux-ci ne sont pas homogènes ; il faut permettre la circulation, régulée par des vannes, d'IC entre stocks. Nous proposons ici les bases d'un ensemble de six primitives obtenues par combinaison de composants de base d'un SED permettant de modéliser une CLH. Un cas réel complexe d'utilisation de ces primitives est présenté.

**MOTS-CLÉS :** chaîne logistique hybride, simulation à événements discrets, flux poussés, flux tirés.

## 1 INTRODUCTION

La définition des processus élémentaires discrets et continus (§1.1) est un préalable à celle des chaînes logistiques discrètes, continues et hybrides (§1.2). On s'intéresse ici à l'évaluation des conséquences de décisions de pilotage opérationnel d'une chaîne logistique (§1.3). L'approche retenue est celle de la modélisation/ simulation de chaînes logistiques hybrides (§1.4), ce qui nous conduira à présenter l'organisation de notre article (§1.5).

### 1.1 Processus élémentaires discrets et continus

Un *processeur* discret a pour vocation de traiter (transformations physiques ou spatiale (transport)) des items correspondant à des objets (produits, composants...), que l'on qualifie ici d'*items*. Un processus élémentaire discret (PED) est constitué d'un processeur discret ; il est alimenté par un ou plusieurs stocks, dans des quantités régies par des nomenclatures et, après un traitement, dont la durée est déterminée par une gamme opératoire, il expédie un ou plusieurs items dans un ou plusieurs stocks. La définition du processeur dépend du niveau d'observation retenu et de son usage dans une modélisation.

De la même façon, un processus élémentaire continu (PEC) a pour vocation de transformer des flux (liquides, granulés...) dans un processeur continu qui prélève ces

flux d'un ou plusieurs stocks et émet des flux en sortie, vers un ou plusieurs stocks. Les débits de ces flux d'entrée et de sortie sont liés par des proportions correspondant à une nomenclature, qui dépendent des performances du processeur. Sous certaines conditions, on peut calculer analytiquement le temps moyen de séjour des flux entrants dans le processus de transformation. Cette durée s'apparente à un temps opératoire.

### 1.2 Chaînes logistiques discrètes, continues et hybrides

On note {PED} un enchaînement de PED, le lien s'opérant habituellement par des stocks partagés (un stock de sortie d'un PED constituant un stock d'entrée du PED suivant). De la même façon, on note {PEC} un enchaînement de PEC, liés par des stocks. Le graphe correspondant à ces enchaînements peut être quelconque. Il s'ensuit qu'une chaîne logistique discrète (CLD) relève de la configuration {PED} en y ajoutant un ou plusieurs points d'entrée {PE} pour faire arriver des items dans des stocks qui seront traités dans la CLD, ainsi que un ou plusieurs points de sortie {PS} par lesquels sortent les items après leur dernier traitement. La CLD se définit donc par la séquence {PE} → {PED} → {PS}.

De la même façon, une chaîne logistique continue (CLC) relève de la configuration {PEC} en y adjoignant des

points d'entrée {PE} et de sortie {PS}. Une CLC se définit donc par la séquence {PE} → {PEC} → {PS}.

Une chaîne logistique hybride (CLH) combine {PED}, {PEC}, {PE} et {PC}. Par la suite, on sera amené à distinguer deux types de CLH :

**CLH Type 1** {PE} → {PEC} → {PED} → {PS}

**CLH Type 2** {PE} → {PEC} → {PED} → {PS}

### 1.3 Décisions de pilotage des chaînes logistiques

La typologie classique (Anthony, 1965) distingue les décisions stratégiques tactiques et opérationnelles ; par la suite, on y adjoint les décisions de type temps réel. On s'intéresse ici aux décisions opérationnelles. Ceci implique, d'une part, l'usage d'une maille temporelle allant de plusieurs minutes à quelques heures et un horizon pouvant aller jusqu'à quelques semaines et, d'autre part, de travailler avec une maille spatiale privilégiant les échanges entre des machines (ou groupe de machines) reliés par des stocks, sans se préoccuper de ce qui se passe dans ces machines.

Cette orientation décisionnelle implique de tirer les conséquences de la distinction classique entre productions à flux tirés et flux poussés dans la modélisation / simulation d'une CLH.

- Dans la production à flux tirés, la production se fait à la commande (carnet de commandes, kanbans...). Les flux d'informations (état du système, décisions programmées...) pilotent les flux physiques. La modélisation / simulation doit synchroniser des flux informationnels et des flux physiques et prendre en compte la disponibilité prévisionnelle des ressources (décision initiale).

- Dans la production à flux poussés, les processeurs produisent tant qu'ils disposent de l'ensemble requis des matières (composants ou fluides ou granulés) et que les stocks de destination ne sont pas saturés. Les flux circulent alors dans le système productif sans que cela ne relève de décisions explicites. Dans ce cadre, les décisions portent principalement sur la disponibilité prévisionnelle des ressources productives. Certaines décisions programmables peuvent être intégrées dans le modèle de simulation mais elles relèvent d'une logique cybernétique.

Les CLH auxquelles on s'intéresse ici combinent pilotages opérationnels par flux poussés, en amont de la CLH, et par flux tirés, en aval. L'un des objectifs de la simulation est de vérifier la cohérence d'un ensemble de décisions prises à différents endroits de la CLH et de faciliter la comparaison de scénarios décisionnels en termes d'efficacité et d'efficience.

### 1.4 Modélisation/ simulation des chaînes logistiques discrètes, continues et hybrides et éclairage de la prise de décisions opérationnelles

Le problème auquel on s'intéresse ici est celui de la modélisation simulation d'une CLH utilisée pour éclairer la prise de décisions opérationnelles ou tactiques.

La modélisation / simulation des CLD et CLC dispose d'instrumentations éprouvées : en particulier, la simulation à événements discrets (SED) pour une CLD et la dynamique des systèmes (DS) pour une CLC. Plusieurs approches sont envisageables pour la modélisation / simulation de la CLH (on reviendra sur ce point en section 2).

- La première repose sur un mécano d'instrumentations prenant chacune en charge un {PED} ou un {PEC}, dans le cadre d'une approche intégrée gérant les échanges d'informations entre ces instrumentations.

- La seconde, privilégiée ici, consiste à mobiliser, en l'adaptant si nécessaire, l'une des instrumentations dédiées (SED ou DS). On peut souligner dès à présent que la granularité spatio-temporelle qu'implique la modélisation / simulation du pilotage d'une CLH par un ensemble de décisions opérationnelles rend très difficile l'utilisation de la DS pour décrire des processus discrets.

L'utilisation directe de la SED pour une modélisation/simulation d'une CLH de type 2 n'est pas possible pour deux séries de raisons.

- La capacité d'un stock dans une SED est définie en nombre d'items, ce qui n'est utilisable pour un fluide que si les caractéristiques volumiques ou pondérales de tous les IC qu'il contient sont identiques, ce qui n'est le cas que dans des exemples simplistes. Ensuite, dans les processus continus, les stocks peuvent être directement connectés et s'alimenter mutuellement, les échanges étant contrôlés par des vannes. Les primitives de base des SED ne permettent pas ce type d'échange.

- Dans l'enchaînement de deux PEC (séparés par un stock), les caractéristiques volumiques de l'IC produite par le PEC-amont a peu de raison de correspondre à celles requises par le PEC-aval. Il faut alors disposer d'un mécanisme effectuant la transformation requise, tout en garantissant la conservation de la matière

### 1.5 Organisation de la communication

On commencera par une revue de la littérature (section 2) qui permettra de montrer qu'il n'existe pas de travaux significatifs proposant de modéliser une CLH de type 2, par la SED, afin d'évaluer des décisions opérationnelles. Dans une section 3, nous proposerons un ensemble de *primitives* combinant les composants de base des SED, pour être en mesure d'effectuer une modélisation simulation d'une CLH, au niveau de granularité pertinent pour l'analyse de décisions opérationnelles. Enfin dans une section 4, nous présenterons un exemple réel d'une simulation d'un CLH complexe, mobilisant la SED et les composants proposés ici. Nous terminerons par une rapide conclusion en section 5.

## 2 REVUE DE LA LITTÉRATURE

Notre objectif de cette revue de littérature est de présenter les raisons du choix de modéliser une CLH par la SED, et de faire un état des lieux des techniques déjà utilisées ; permettant la modélisation des processus hybrides par la SED. Dans plusieurs travaux de recherche, le choix du type de la simulation est souvent basé sur le niveau déci-

sionnel étudié (niveau opérationnel, tactique ou stratégique) (Tako et Robinson, 2012), (Jahangirian *et al.*, 2010). Dans ce présent article, en plus du niveau de décision, nous nous intéressons à étudier le choix de la technique de simulation (SED, DS, SH) vis-à-vis la nature ; discrète ou continue ; des processus composant la CL étudiée.

## 2.1 Méthodologie de la revue de littérature

Aucune restriction n'a été mise dans la recherche des articles sur l'année, la conférence ou la revue. Nous avons essayé les combinaisons suivantes de mots clés : « Continuous process », « discrete process », « discretization », « operational management », « supply chain », « System dynamics » et « Discrete event simulation ». Les questions que nous nous posons sont les suivantes :

- La DS et les SED ne sont-ils utilisés respectivement que pour une CLC et une CLD ?
- Quel lien existe-t-il entre le modèle de simulation et le niveau décisionnel à éclairer ?
- Si la SED est utilisée pour une CLH, quelles sont les techniques de discrétisation mobilisées ?

## 2.2 Etat des lieux

J. Forrester, fondateur de la DS, définit cette méthodologie comme une approche de modélisation des politiques de l'entreprise et comme un outil aidant à résoudre des problématiques du top management. Elle modélise le pilotage de l'évolution des systèmes dans lesquels des flux circulent et leurs évolutions sont représentées par des équations différentielles. La DS modélise la CL à un niveau agrégé et permettent d'avoir une vision macroscopique du fonctionnement du système étudié. (Scholz-Reiter *et al.*, 2005), (Kleijnen, 2005), (Venkateswaran et Son, 2004). Elle a été utilisée pour étudier l'impact du phénomène du *Bullwhip Effect*. (Pierreval *et al.*, 2007), (Higuchi *et al.*, 2004), (Rozman *et al.*, 2014)... La DS est également appropriée pour modéliser l'évolution en temps réel des processus continus et d'en faciliter le pilotage ; dans ce cadre, elle permet de comprendre l'évolution de paramètres physiques (température, pression...) à garder sous contrôle.

La SED est utilisée pour étudier le comportement des systèmes en réponse à certains événements (franchissement de seuils, décisions de pilotage...) survenant à des moments précis au cours de la simulation (Venkateswaran et Son, 2004). La SED est plus appropriée pour l'étude des décisions opérationnelles et tactiques (Andersson *et al.*, 1998). Elle permet une représentation du système productif à une granularité (spatiale, temporelle...) pertinente pour évaluer l'impact de décisions de pilotage.

Nous distinguons trois conceptions possibles de la simulation hybride :

- Deux modèles différents de simulation sont mobilisés pour la modélisation de deux niveaux décisionnels différents. Généralement un premier modèle basé sur

la DS est développé pour répondre à des problématiques stratégiques à un niveau agrégé (planning annuel de production, politique d'externalisation ...). Par la suite un modèle de SED est utilisé pour simuler le comportement des décisions opérationnelles intégrées dans la solution obtenue par le premier modèle (Jamalnia *et al.*, 2013), (Rabelo *et al.*, 2007), (Venkateswaran and Son, 2004)... Cette approche se retrouve dans d'autres combinaisons d'instrumentations couplant des décisions stratégiques et opérationnelles ou tactiques. Il est très fréquent d'utiliser la programmation mathématique au niveau stratégique pour définir des niveaux de ressources permanentes à affecter à des besoins ; la SED est ensuite utilisée pour vérifier la viabilité de ces décisions stratégiques à un niveau fin, pouvant prendre en compte des perturbations aléatoires. La rapidité de l'obtention de la solution finale dépend du couplage réalisé entre les deux modèles. Dans un premier cas où les résultats du premier modèle sont automatiquement utilisés par le deuxième modèle par un interfaçage direct, le couplage est fort. Il est faible dans le cas contraire, ce qui implique plus de temps pour prendre une décision satisfaisante.

- La seconde conception est souvent qualifiée de simulation combinée. Elle correspond à une modélisation intégrée incorporant simultanément des variables discrètes et continues, dans un même modèle (Lee *et al.*, 2002). Certains auteurs (Barbosa et Azevedo, 2017) qualifient d'enrichissement ce type de SH qui implique l'usage d'un logiciel fondé sur l'une des deux approches de simulation, avec une mobilisation de méthodes caractérisant l'autre approche. Par exemple, dans (Music et Matko, 1999) et (Fabre *et al.*, 2011), la DS est utilisée pour le suivi des paramètres physicochimiques des réactions chimiques se faisant par batch : ces processus sont de nature hybride, des événements discrets déclenchent le lancement ou l'arrêt d'une transformation chimique contrôlée ensuite par un système d'équations différentielles. Dans ce contexte, la partie discrète est représentée par des contrôleurs logiques modélisés par exemple, à l'aide des réseaux de Petri. Symétriquement, un exemple d'intégration d'un module continu dans un SED peut être trouvé dans (de Ugarte, 2008) ; dans ce cas, un PED qui produit à la commande des lots différents, de manière continue, c'est-à-dire sans purge entre deux lots, ce qui implique un certain mélange en début et fin de production d'une commande. Un module continu (basé sur des équations différentielles) est encapsulé dans le premier PED d'un {PED} ; ce programme sélectionne le meilleur ordonnancement des commandes et détermine les temps opératoires.
- L'utilisation de la SH à un même niveau décisionnel pour une chaîne logistique de type 2, modéliser les PEC continus par la DS et les PED par la SED et opérer un couplage fort entre ces modèles. Nous n'avons pas trouvé d'articles qui défendent cette approche. La raison en est sans doute que cette approche implique

l'usage des hypothèses restrictives de modélisation et augmente la complexité de la programmation logicielle et les temps de calcul (Chen *et al.*, 2002).

Dans le reste de cette revue de littérature nous regarderons comment la SED est utilisée pour modéliser des chaînes logistiques hybrides.

### 2.3 Simulation à événements discrets pour une CLH

Quand la technique de simulation retenue est la SED, on se retrouve face des problèmes techniques nécessitant des règles de traduction permettant la représentation d'un processus continu par un processus discret. On retrouve dans la littérature, des exemples d'utilisation de la SED pour des CLH de type 1. Le PEC situé en amont de la CLH a pour vocation de discrétiser un flux continu pour émettre, à intervalles réguliers, des IC correspondant à une quantité de liquide ou de granulé. Cette fragmentation de flux correspond généralement à un conditionnement (paquet, sac, camion...) utilisé par le transport. Par la suite, les items circulant dans la CLH ne font normalement plus l'objet de traitement par un PEC ou peuvent le faire par un traitement par lot effectué par un PED (Chen *et al.*, 2002).

(Meyer *et al.*, 2011) ont développé une approche basée sur des modèles séparés pour la chaîne logistique de la production des carburants, un modèle de simulation pour la partie gaz et un autre pour la partie liquide. Pour la production des gaz ou des liquides les inputs sont ajoutés dans le réacteur par lots. Les lots émis par le premier processus (par un PED), deviennent des inputs du second processus (PED), après transformation, les caractéristiques volumiques des lots émis ne correspondant pas à ceux demandés. On retrouve ici l'une des raisons évoquées en introduction sur l'impossibilité d'utiliser directement la SED pour modéliser une CLH.

Auteurs	Problème traité	Niveau de décision			Nature de la CL			Simulation		
		Stratégique	Tactique	Opérationnel	CLC	CLD	CLH	SD	SH	
Higuchi et al (2004)	Prévision de la demande (Bullwhip effect)	X				X		X		
Rabelo et al (2007)	Pilotage opérationnel des configurations d'externalisation	X	X						X	
Doloi et al (2002)	Etude de l'impact de phase développement des projets sur les coûts de leur cycle de vie	X				X		X		
Lehr et al (2013)	Simulation des stratégies de recouvrement	X				X		X		
Jamali et al (2013)	Planification de la production	X				X			X	
Roosman et al (2014)	Etude de l'impact de l'investissement dans la production	X				X		X		
Floodas et al (2004)	Vue générale sur les modèles discrets et continus		X		X			X	X	
Gianfranceschi et al (2006)	Modèle généralisé à événements discrets					X		X		
Hamadi et al (1994)	Etude de l'impact des microbes sur les aliments			X		X		X		
Kyamalya et al (2005)	Problème de localisation	X				X		X		
Weston et al (1999)	Planification de la capacité		X			X		X		
Owens et al (2002)	Gestion de la capacité: alternatives de design	X	X			X		X		
Lynes (2000)	Prévisions et impact de la dynamique du marché					X		X		
Lin et al (2014)	Bullwhip effects	X				X		X		
Pierrelat et al (2007)	Impact des perturbations de la demande	X				X		X		
Lee et al (2002)	Gestion des stocks (du fournisseur au client)	X				X		X		
Misic and Marko (1999)	Contrôle des processus chimiques			X					X	
Venkateswaran and Son (2004)	Gestion des stock et management opérationnel	X	X			X			X	
Chen et al (2002)	Optimisation de la configuration des processus		X			X		X		
Schedroff et Cheng (2003)	Propagation de l'impact des événements sur la CL		X			X		X		
Meyer et al (2011)	Etude de l'impact de la variabilité des installations sur la CL des carburants	X	X		X			X		
Fabre et al (2011)	Contrôle des paramètres physicochimiques		X	X		X		X		
de Ugarte et al (2009)	Ordonnement à temps réel			X		X		X		

Tableau 1 : Grille d'analyse des articles

Dans la suite nous présentons des primitives génériques indépendante du logiciel de simulation discrète utilisé et dépassant ce que proposent (Chen *et al.*, 2002) et (Meyer *et al.*, 2011). Ces primitives permettent de modéliser, avec un seul modèle et avec un degré de détail suffisant tout

type de configuration d'une CLH, pour éclairer la prise des décisions de type opérationnel.

### 3 PRINCIPES DE MODÉLISATION D'UNE CHAÎNE LOGISTIQUE HYBRIDE PAR LA SIMULATION À ÉVÉNEMENTS DISCRETS

Dans cette section on s'intéresse à la description d'un ensemble de primitives génériques permettant de modéliser un {PEC}, et donc une CLH, dans le niveau de granularité induit par l'analyse du pilotage opérationnel. Les fondements logiques de certaines de ces primitives ont été présentés très sommairement dans (Degoun *et al.*, 2015) ; nous explicitons ici un certain nombre de principes et de règles permettant de créer ces primitives génériques dans n'importe quel SED.

Après avoir rappelé les caractéristiques des composants de base d'un SED (§3.1), nous précisons comment s'effectue le processus de discrétisation d'un flux continu (§3.2), avant de détailler la primitive *processus continu* permettant de représenter correctement un PEC dans un SED (§3.3). Ce processus continu pouvant mobiliser plusieurs inputs et produire plusieurs outputs, il convient d'adapter au cas continu, la définition des nomenclatures (§3.4). Les IC, séjournent dans des stocks ; la définition de la capacité d'un stock dans un SED passe par un nombre maximum d'items qu'il peut contenir, ce qui ne convient pas si les IC ont des caractéristiques volumiques différentes, ce qui est généralement le cas ; on est donc amené à créer la primitive *stock capacitaire* (§3.5). Les caractéristiques volumiques des IC produits diffèrent généralement de celles des PEC qui les utilisent, nécessitant la création de la primitive *convertisseur* (§3.6). On examinera enfin les problèmes posés par le contrôle des flux, ce qui conduit à créer une primitive *vanne* pour contrôler des échanges entre les stocks (§3.7) ou d'ordres de fabrications (OF) liés au pilotage à flux tirés, ce qui conduit à un éclatement de ces OF lorsque ces commandes portent sur des quantités de flux (§3.8).

Dans la présentation de ces primitives nous nous appuyons sur des exemples tirés de la CLH de l'OCF (cf. § 4).

#### 3.1 Composants de base d'un SED

La SED décrit le processus de production discrète d'un ensemble de biens ou de services par un système productif (SP) dans lequel circulent des *items* qui arrivent dans ce système par des *points d'entrée* et le quittent par des *points de sortie*, après avoir transité par une séquence de *processeurs* et de *stocks*, en utilisant des *chemins* prédéfinis. Six classes d'objets sont donc mobilisées : la terminologie utilisée ici est générique et donc indépendante du logiciel utilisé ; il en est de même pour les propriétés de ces classes d'objets qui peuvent être implémentées de différentes façons.

- Les **items** correspondent à des objets ou des personnes qui font l'objet d'un traitement dans le SP, ou d'informations (commandes, ordres de fabrication...) utilisées dans le pilotage des flux (principalement dans la production à flux tirés). Les caractéristiques spécifiques d'un item, utiles à son traitement par le SP sont consignées dans des

*labels*, qui jouent le rôle similaire à celui des variables qualitatives ou quantitatives utilisées pour caractériser les éléments d'un ensemble en statistique descriptive. Nous utiliserons ultérieurement la convention *Item.Label* pour accéder (en écriture ou lecture) au label d'un item.

- Les **points d'entrée** permettent l'introduction d'items dans le SP. Ils sont caractérisés par des *lois d'arrivée* de ces items (stochastiques ou déterministes) et ils définissent la *caractérisation initiale* de ces items entrants par des labels.

- Les **points de sortie** permettent l'évacuation d'items ayant séjourné dans le SP, après traitement. Ils ne possèdent pas de caractéristiques particulières et sont passifs, ce qui implique que les items devant sortir sont envoyés à ces points de sortie par des processeurs.

- Les **processeurs** ont pour fonction de réaliser une *opération* sur un ou plusieurs items, prélevés dans des stocks. Ces opérations modifient certaines caractéristiques des items qui peuvent être morphologiques (transformation de matière...), informationnelles (analyse...) ou spatiales (déplacements). Elles peuvent conduire à la modification de certains labels d'items traités. Elles peuvent également nécessiter des ressources humaines et matérielles mais ces ressources n'ont pas besoin d'être détaillées si elles sont dédiées au processeur. Un processeur est caractérisé par son temps opératoire (logique de *gamme*) qui peut être déterministe ou aléatoire et dépendre des valeurs des labels des items à traiter.

Le traitement peut être exécuté par un processeur sur un item unique. Il peut aussi nécessiter le prélèvement de plusieurs items dans un ou plusieurs stocks. Dans ce dernier cas, les items peuvent être de nature différente (définie par un label), et l'opération relève d'une logique d'assemblage, ou de nature similaire, et l'opération relève d'une logique de lotissement. Dans les deux cas, l'opération mobilise une *nomenclature*. À l'issue du traitement, le processeur expédie un (ou plusieurs items) vers un stock (ou plusieurs stocks), le processeur est alors inoccupé. Un ou plusieurs items ne peuvent être admis dans un processeur que si celui-ci est inoccupé.

Un processeur fictif (caractérisé par un temps opératoire nul) peut être mobilisé dans la description de processus complexes, en particulier pour décrire le routage en entrée ou en sortie de plusieurs items (par exemple, règles combinant des conditions logiques ET et OU).

Un processeur possède un certain nombre de labels par défaut (décompte des items traités...). Il est également possible d'y ajouter des labels spécifiques pour permettre un meilleur contrôle du pilotage du SP. Nous utiliserons ultérieurement la convention *Processeur.Label* pour accéder (en écriture ou lecture) au label d'un processeur.

- Un **stock** est un lieu où peuvent séjourner des items mais sans y subir de traitement, contrairement aux processeurs. Le stock est passif, c'est-à-dire qu'un item n'entre en stock que s'il est envoyé par un point d'entrée ou un processeur et ne peut en sortir que s'il est appelé par un processeur. Un stock possède un certain nombre de labels par défaut (capacité...). Il est également possible d'y ajouter des labels spécifiques pour permettre un meilleur contrôle

du pilotage du SP. Nous utiliserons ultérieurement la convention *Stock.Label* pour accéder (en écriture ou lecture) au label d'un stock.

- Les **routes**, représentées par des flèches reliant deux des icônes représentant des objets des quatre classes précédentes, matérialisent des déplacements possibles d'un item d'un point à un autre. L'itinéraire emprunté par un item est piloté par les valeurs de ses labels et un ensemble de règles utilisées par les processeurs. On associe parfois un temps de transport associé à ce déplacement.

L'exécution de programmes est possible lorsqu'un événement survient (date ; arrivée d'un item à un point d'entrée, dans un processeur ou dans un stock ; en cours d'exécution d'une opération par un processeur ; à la sortie d'un item à un point de sortie, d'un processeur ou d'un stock) ou à des dates. Ces programmes utilisent les labels des items, des processeurs et des stocks, plus d'autres informations stockées en mémoire, pour mieux gérer les flux dans le SP.

### 3.2 Discrétisation d'un flux continu

La modélisation d'un processus continu par la SED repose sur la discrétisation de flux admis en entrée du système productif, avant admission dans un stock capacitair (voir § 3.5). Cette discrétisation crée des IC qui héritent des caractéristiques chimiques du flux et dont les caractéristiques volumiques et pondérales dépendent du **débit** du flux et de l'intervalle de temps  $\theta$  qui délimite le lot de liquide ou de granules associé aux IC. Par exemple, avec un pas de discrétisation  $\theta=9'$ , l'arrivée de souffre liquide dans la CLH avec un débit de 35,3 T/H conduit à la création d'un IC de 5,3 T, toutes les 9'. Ces IC sont traités ensuite par des processus discrets ou continus.

### 3.3 Modélisation d'un processus continu par la SED

Un processus continu se caractérise par le passage d'un (ou plusieurs, cf. § 3.4) flux dans une installation productive, où il séjourne un temps moyen  $\mu$  avant d'en sortir avec des caractéristiques morphologiques et chimiques différentes. La modélisation de ce processus continu par un processeur discret de temps opératoire  $\mu$  est possible, dans une CLH possédant plusieurs processeurs continus de temps opératoires  $\mu$  différents, à condition d'utiliser des convertisseurs (cf. §3.6). Le problème qui se pose alors est que l'étude du comportement de la CLH, implique de recueillir périodiquement la production des processeurs, mesurée en sortie de processeurs, la plage  $\Delta$  de recueil de ces productions pouvant être, par exemple, de 1 heure. La solution évoquée ci-dessus conduit à des productions horaires très irrégulières, même en régime de croisière, dès lors que  $\mu > \Delta$ . Il convient alors de dissocier la définition de l'intervalle  $\theta$  séparant deux admissions successives dans un PC (et donc deux sorties successives de ce PC), de son temps opératoire  $\mu$ .

Il est évident que plus la valeur de  $\theta$  est faible, meilleur est le lissage de la production de ce PC (modélisé par un

PD). Cette définition de l'intervalle  $\theta$  entre deux admissions successives dans un PC donné, doit tenir compte de trois contraintes.

- L'intervalle d'admission  $\theta$  ne peut dépasser l'intervalle  $\Delta$  de recueil d'information dans la simulation sans perturber les résultats de cette simulation ( $\rightarrow \theta \leq \Delta$ ).

- Si  $\theta > \mu$ , le processeur est réputé ne pas produire durant la période  $\theta - \mu$ , d'où une diminution de production ne correspondant pas à la réalité

- Si  $\theta < \mu$ , on se heurte au problème de l'impossibilité, pour le processeur de base des SED, de modéliser le traitement simultané d'un ensemble d'items arrivant et partant à des dates différentes.

Pour tenir compte de ces contraintes, dans le cas logique où  $\theta \leq \text{Min}(\Delta, \mu)$ , nous proposons une primitive générique d'un **processus continu** (voir figure 1), consistant à le représenter par trois composants de base d'un SED dont l'usage est « dévoyé ».

- En amont, ce stock est alimenté par un **processeur d'admission et de début de traitement** (Processeur\_1, dans la figure 1), doté d'un temps opératoire  $\theta$  qui correspond à la fois à la cadence d'admission dans le processus continu et au début du temps de traitement du processeur continu. Ce processeur est connecté aux stocks des différents inputs du processus continu et prélève un item dans chaque stock, conformément à la nomenclature de fabrication (voir § 3.4). À titre d'exemple, on a retenu les trois stocks des produits intervenant dans la fabrication de l'acide phosphorique (voir §3.4).

- Un **stock fictif de fin de traitement** (Stock\_1), doté d'un temps de séjour minimal  $\mu - \theta$  correspondant au temps résiduel de traitement par le processus continu. Il permet de prendre en compte des items se trouvant à différents stades d'avancement dans le processus continu.

- En aval, le stock fictif est connecté à un **processeur fictif d'enlèvement** (Processeur\_2), de temps opératoire nul, dont la fonction est de prélever, du stock fictif, tout item ayant séjourné le temps minimal  $\mu - \theta$ . En général, une fois l'opération terminée, des items de natures différentes sont expédiés dans autant de stocks, conformément à la nomenclature utilisée (cf. 3.4), ces stocks étant généralement capacitaires (cf. § 3.5).

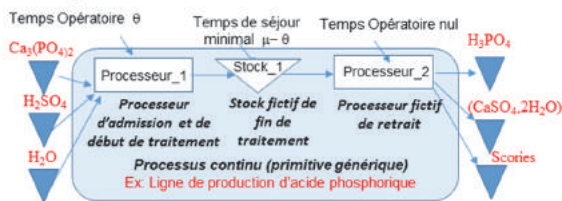
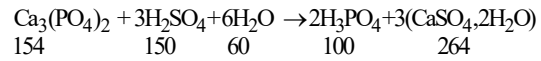


Figure 1 : Modélisation de la primitive processus continu

### 3.4 Nomenclature d'un processus continu

D'une manière générale, un processus continu prélève un certain nombre d'inputs différents et produit des outputs différents. Appuyons-nous sur l'exemple de la nomenclature d'acide phosphorique. Sa production fait appel à

l'équation molaire suivante, complétée par les nombres atomiques de ces molécules, qui vérifie le principe de conservation de la matière dans la transformation et dans laquelle, le phosphogypse ( $\text{CaSO}_4 \cdot 2\text{H}_2\text{O}$ ) est un solide qui contient de l'eau.



En toute rigueur, les débits entrants et sortants d'un processus continu sont des dérivées.  $dq/dt$ , où  $q$  est un poids. Dans le cadre retenu, ces débits sont définis sur une courte période de temps ( $\rightarrow \Delta q / \Delta t$ ) ; on conviendra ici de retenir la minute ( $\rightarrow \Delta t = 1$ ) et de définir les débits en kilogrammes/minute. On définit ensuite l'un des flux comme étant le *flux-maitre* dont le débit sert de référence pour décrire les débits des autres flux. Il est préférable de retenir le flux du produit fabriqué (ici acide phosphorique) comme flux-maitre, mais n'importe quel autre intrant ou extrant aurait pu être retenu. On peut alors écrire, en indiquant  $q$  par les produits chimiques :

$$\frac{\Delta q_{\text{Ca}_3(\text{PO}_4)_2}}{\Delta t} = \frac{154}{100} \cdot \frac{\Delta q_{\text{H}_3\text{PO}_4}}{\Delta t} = 1,54 \cdot \frac{\Delta q_{\text{H}_3\text{PO}_4}}{\Delta t}$$

$$\frac{\Delta q_{\text{H}_2\text{SO}_4}}{\Delta t} = \frac{150}{100} \cdot \frac{\Delta q_{\text{H}_3\text{PO}_4}}{\Delta t} = 1,5 \cdot \frac{\Delta q_{\text{H}_3\text{PO}_4}}{\Delta t}$$

$$\frac{\Delta q_{\text{H}_2\text{O}}}{\Delta t} = \frac{60}{100} \cdot \frac{\Delta q_{\text{H}_3\text{PO}_4}}{\Delta t} = 0,6 \cdot \frac{\Delta q_{\text{H}_3\text{PO}_4}}{\Delta t}$$

$$\frac{\Delta q_{(\text{CaSO}_4 \cdot 2\text{H}_2\text{O})}}{\Delta t} = \frac{264}{100} \cdot \frac{\Delta q_{\text{H}_3\text{PO}_4}}{\Delta t} = 2,64 \cdot \frac{\Delta q_{\text{H}_3\text{PO}_4}}{\Delta t}$$

Dans la simulation du processus de fabrication d'acide phosphorique, la fixation du débit de production d'acide phosphorique détermine automatiquement ceux des arrivées de  $\text{Ca}_3(\text{PO}_4)_2$ , de l'acide sulfurique et de l'eau. Avec un pas d'admission  $\theta = 60'$  et un débit horaire de production d'acide phosphorique de la ligne de production de 144 T/heure, ce processus chimique admet toutes les 60 minutes un item correspondant à 221,76 T de  $\text{Ca}_3(\text{PO}_4)_2$ , un item correspondant à 216 T d'acide sulfurique et un item correspondant de 86,4 T d'eau. Toutes les 60 minutes, sortent un item correspondant à 144 T d'acide phosphorique et un item correspondant à 380,16 T de phosphogypse. La conservation des masses est assurée :  $144 + 380,16 = 221,76 + 216 + 86,4$ .

Cette nomenclature théorique ne se retrouve pas dans le contexte industriel qui ne permet pas d'atteindre ces rendements théoriques. Pour atteindre la production périodique de 144 T, il faut un peu plus d'inputs. Sur une ligne existante, par exemple, il faut 217,08 T d'acide sulfurique (au lieu de 216 T) pour produire 144 T d'acide phosphorique. Cette augmentation d'inputs consommés (qui dépend des installations industrielles utilisées) s'accompagne d'une production inévitable de scories, ce qui permet à l'équation de conservation d'être respectée. La Figure 2 représente la nomenclature industrielle utilisée par une ligne de production de Jorf (cf. § 4).

Enfin, pour effectuer cette production des inputs additionnels n'intervenant pas dans la réaction chimique sont nécessaires (énergie, liquides de refroidissement et, éventuellement, catalyseurs). Ces inputs additionnels peuvent être importants à prendre en compte pour faciliter la gestion de ces ressources, qui peuvent être onéreuses, et dont la consommation a souvent un fort impact dans l'écosystème dans lequel ces unités de production sont insérées.

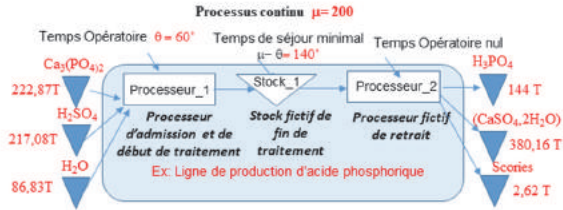


Figure 2 : Nomenclature d'un processus continu

### 3.5 Gestion des stocks capacitaires contenant des liquides ou des granulés

Dans les SED la capacité d'un stock est exprimée en nombre d'items qu'il peut contenir. Dans les processus continus la capacité est exprimée en poids ou en volume. Un stock capacitair (à capacité limitée) est généralement alimenté par des processus continus différents, ce qui implique que les IC stockés n'ont pas de caractéristiques homogènes et, donc, qu'il n'est pas possible de définir la capacité de ce type de stock par un nombre maximal d'IT. Nous avons créé une primitive générique du **stock capacitair** (voir figure 3), consistant à le représenter par trois composants, un processeur fictif et deux stocks. Le stock capacitair est caractérisé par les labels `Stock_3.Capacite_Offerte` et `Stock_3.Capacité_Utilisée`, dont les valeurs, exprimées en poids (ou en volume), sont initialisées en début de simulation. Le premier label est statique et garde la valeur définie en début d'initialisation. Le second label varie dynamiquement en fonction de l'utilisation instantanée de ce stock. Les items entrants sont caractérisés par le label `Item.Poids`.

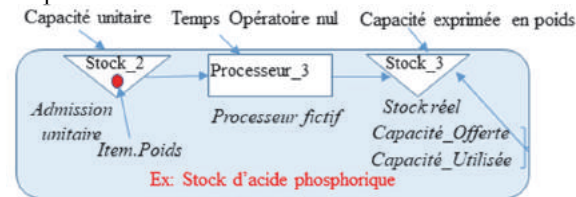


Figure 3 : Modélisation d'un stock capacitair

- Le stock *Admission unitaire* (Stock\_2), accueille un item caractérisé par son label `Item.Poids`. Pour éviter certains problèmes de désamorçage en amont, la capacité maximale de ce stock, qui est normalement de 1, devient égale au nombre de processus continus qui l'alimentent.

- Un *processeur fictif* (Processeur\_3), avec un temps opératoire nul n'accepte d'items que si la capacité résiduelle du stock réel est suffisante ce qui est réalisé par le programme suivant exécuté avant prélèvement de l'item  
`Si Stock_3.Capacité_Utilisée+Item.Poids < Stock_3.Capacite_Offerte`  
`alors admettre l'item dans Processeur`  
`Sinon Bloquer le prélèvement de l'item`

- À l'entrée d'un item dans *stock réel capacitair* (Stock\_3), la mise à jour de la capacité utilisée est effectuée par le programme suivant:

```
Stock_3.Capacité_Utilisée ← Stock_3.Capacité_Utilisée + Item.Poids
```

À la sortie d'un item de ce stock, la mise à jour de la capacité utilisée est effectuée par le programme suivant:

```
Stock_3.Capacité_Utilisée ← Stock_3.Capacité_Utilisée - Item.Poids
```

Un stock capacitair peut être alimenté par d'autres stocks (cf. § 3.7) ou par des processus continus. Un processus continu qui émet des outputs alimentant des stocks capacitaires, doit s'arrêter dès que l'un de ces stocks capacitaires est saturé. La solution consiste alors à ajouter un programme s'inspirant du test décrit ci-dessus pour bloquer l'alimentation du processus continu ; ce test est à effectuer à l'entrée du Processeur\_1 de la figure 1.

### 3.6 Couplage de deux processus continus par un convertisseur

Le couplage entre deux processus continus, caractérisés par les cadences d'admission  $\theta_1$  et  $\theta_2$ , se fait nécessairement par l'intermédiaire d'un stock (capitair, le plus souvent). Le problème auquel on s'intéresse ici est celui posé par la non-coïncidence du poids des items produits par l'amont et demandés par l'aval, ce qui conduit à créer une primitive générique appelée **convertisseur** qui mobilise quatre composants (deux processeurs fictifs et deux stocks fictifs) permettant de créer un item ayant le poids requis en aval à partir d'un ou plusieurs items produits en amont.

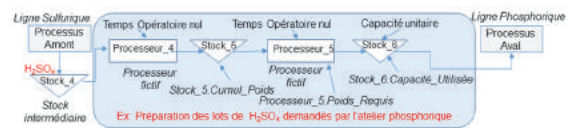


Figure 4 : Modélisation d'un convertisseur

- Le Stock\_6 est utilisé pour recevoir le lot ayant la valeur du poids demandée. Ce *stock* a une capacité unitaire pour assurer une conversion au fur et à mesure des besoins de la production. Ce stock possède le label `Capacité_Utilisée`.
- Le Stock\_5 est utilisé pour recevoir les items qui vont permettre la création d'un item ayant le poids requis. Ce *stock* possède les labels `Cumul_Poids` et `Nombre_items`. À l'arrivée d'un item dans ce stock le programme suivant est exécuté :

```
Stock_5.Cumul_Poids ← Stock_5.Cumul_Poids + Item.Poids
```

- Un premier *processeur fictif*, Processeur\_4, de temps opératoire nul, a pour fonction de prélever, du stock intermédiaire (Stock\_4), des items pour les pousser dans le stock Stock\_5 à condition que l'on ne dispose pas déjà d'un item ayant le poids voulu, et que l'on n'ait pas déjà dépassé le poids requis par le processus continu aval, valeur consignée dans le label `Processeur_5.Poids_requis` (valeur mise à jour dynamiquement en fonction de l'évolution des productions). Ces conditions sont vérifiées par le programme suivant, exécuté avant l'admission par le Processeur\_4.

```
Si Stock_6.Capacité_Utilisée = 1 alors Bloquer le prélèvement de l'item
Si Stock_5.Cumul_Poids ≥ Processeur_5.Poids_requis
alors Bloquer le prélèvement de l'item
```

- Un second *processeur fictif*, Processeur\_5, de temps opératoire nul, est utilisé pour prélever et fusionner tous

les items séjournant dans Stock\_5. Il ne peut rien faire si le Stock\_6 a un item de poids requis en stock ou si le cumul du poids des items dans le stock Stock\_5 (valeur du label Stock\_5.Cumul\_Poids) n'est pas suffisant. Ces conditions sont vérifiées par le programme suivant exécuté avant l'admission par le Processeur\_5 ; ce programme utilise un paramètre  $k$  pour gérer l'utilisation de ce processeur

```

k ← 0
Si Stock_6.Capacité_Utilisée = 1 alors k ← 1
Si Stock_5.Cumul_Poids < Processeur_5.Poids_requis alors k ← 1
Si k = 1 alors Bloquer le prélèvement des items
Sinon :
Prélever tous les items du Stock_5 et les fusionner en un item unique
avec Item.Poids ← Stock_5.Cumul_Poids - Processeur_5.Poids_requis
Mettre à jour la capacité utilisée du Stock_5 : Stock_5.Cumul_Poids ←
Stock_5.Cumul_Poids - Processeur_5.Poids_requis
    
```

Le Processeur\_5 émet ensuite deux items identiques, le premier vers le Stock\_5 et le second vers Stock\_6. À l'entrée de l'item dans Stock\_6, le programme suivant est exécuté.

```
Item.Poids ← Processeur_5.Poids_requis
```

### 3.7 Le contrôle des flux

On abordera ici le cas du pilotage en flux tirés, qui suppose la synchronisation d'un flux d'informations (commandes...). Le transfert d'un stock de fluide à un autre stock relève de cette logique décisionnelle et fait intervenir une vanne, puisque les stocks sont passifs et ne peuvent émettre d'items vers un autre stock. La gestion des flux se fait généralement suivant un programme de transferts, chaque transfert est planifié pour une date bien définie, pour une quantité et un débit bien définis. Ceci fait que l'intervalle de temps séparant deux transferts successifs dépend d'un planning des transferts prédéfini. Le démarrage d'un transfert et le réglage des débits se font via le réglage des vannes.

La primitive générique proposée, appelé **vanne** et présentée dans la figure 5, mobilise un Convertisseur (cf. § 3.6), un point d'entrée, un **stock** (Stock\_8) et deux **processeurs fictifs** (Processeur\_6 et Processeur\_7).

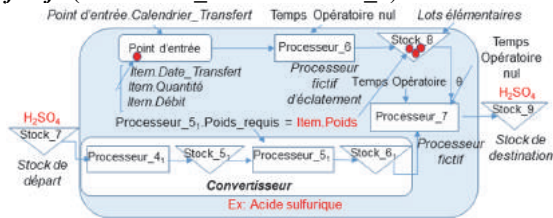


Figure 5 : Modélisation de la primitive Vanne

Le **Point d'entrée** est utilisé pour recevoir les items correspondant à des ordres de transfert d'un stock (Stock\_7) à un autre stock (Stock\_9). Les labels Item.quantité et item.Débit, sont lus dans un tableau et affectés à l'item entrant.

Le Processeur\_6 est utilisé pour dupliquer l'item entrant en  $m$  items en sortie de ce processeur. Ce nombre  $m$  est calculé par l'exécution du programme suivant lors de l'admission de l'item dans le Processeur\_6 :

```
 $m \leftarrow \text{arrondi}[\text{Item.Quantité}/\text{Item.Débit}]$ 
```

L'item sortant se voit attribuer la valeur

```
Item.Poids ← Item.Quantité/m
```

Chacun de ces  $m$  items correspond à la commande d'un lot à expédier. La préparation des lots est assurée par le convertisseur reliant le stock de départ (Stock\_7) au Processeur\_7. Ce dernier fusionne l'item prélevé dans le Stock\_7 (information) à celui prélevé dans le Stock\_6 (matière), ce qui correspond à un processus d'assemblage classique. Le temps opératoire  $\theta$  du Processeur\_7 correspond à l'intervalle séparant deux expéditions successives par la vanne.

### 3.8 Modélisation du pilotage d'un processus continu par des ordres de fabrication

Dans la modélisation d'un atelier de production, travaillant à la commande, le flux physique est piloté par le flux informationnel. On suppose un atelier comportant plusieurs lignes de production parallèles (on prendra l'exemple de lignes de production d'engrais). Le carnet des ordres de fabrication (OF) contenant l'affectation des commandes sur les lignes est un input du modèle de la simulation. Chaque ligne traite les commandes qui lui sont affectées suivant la logique FIFO selon l'ordre indiqué sur le carnet des OF. Le début du traitement d'une commande n'est donc pas défini par une date bien définie (comme dans le cas des transferts, cf. § 3.7) mais il dépend de la date de fin de la commande précédente et du temps de lancement qui dépend quant à lui du séquençement des commandes.

La modélisation que nous proposons pour la représentation d'une ligne de production d'atelier géré à flux tirés est présentée dans la figure 6.

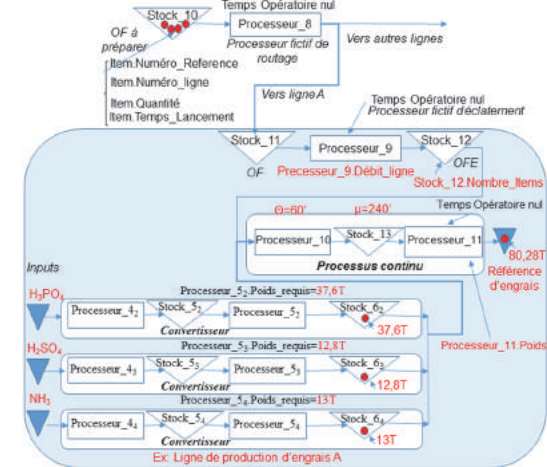


Figure 6 : Modélisation d'un atelier à flux tirés

- Le Stock\_10 est initialisé en début de simulation à partir d'une liste d'OF à exécuter. Chaque OF est caractérisé par la référence à produire (Item.Numéro\_Référence) et un label contenant le numéro de la ligne pour laquelle elle est affectée (Item.Numéro\_ligne), la quantité à fabriquer (Item.Quantité) et le temps de lancement (Item.Temps\_Lancement).
- Le **processeur fictif** Processeur\_8 assure, suivant le label item.Numéro\_ligne, le routage des OF vers les lignes de production. Les OF impliquant la ligne de production étudiée ici sont reçues dans le Stock\_11.



- Le **processeur fictif** Processeur\_9 éclate ensuite chaque OF en  $n$  OFE (OF Élémentaires), ce nombre étant calculé comme le quotient de la quantité à produire (par exemple 11 640T), par le débit de production de la ligne (par exemple 1,347T/minute), multiplié par l'intervalle d'admission  $\theta$  (par exemple 60 minutes). Dans notre exemple, l'OF de 11 640T est éclaté en 145 OFE caractérisés par un poids à produire de 80,28 T. Le Processeur\_9 ne peut pas accepter nouveau item que si le stock\_12 est vide (stock\_12.Nombre\_item = 0). Ces conditions sont vérifiées par le programme ci-dessous exécuté à l'entrée du Processeur\_9 :

```
Si Stock_11.Nombre_Items > 0 alors
  Si Stock_12.Nombre_Items > 0 alors
    Bloquer prélèvement de cet item
```

Le processus continu, piloté par le Processeur\_10, prélève ensuite, à la cadence d'admission retenue pour ce processus ( $\theta=60'$ ), un OFE (information) et les items (matière) correspondants aux inputs nécessaires (acides...) et les assembler en un item unique dont le poids 80,28 T. Dans cet exemple, le poids des composants utilisés est de 37,6 T pour l'acide phosphorique, 54,13 T pour l'ammoniac et 12,8 T, pour l'acide sulfurique.

Nous ne tenons pas en compte ici de la mise à jour du temps d'admission  $\theta$  et du temps de traitement  $\mu$  qui pourraient dépendre de la gamme en cours, mais elle facile et passe par une paramétrisation de  $\theta$  et de  $\mu$  en fonction de numéro de la ligne et la référence de l'OF en cours.

Les nomenclatures et gammes de deux commandes successives, sur une même ligne, n'ont aucune raison d'être les mêmes. Une actualisation des poids des inputs, à préparer par les convertisseurs et le poids des lots de l'OF, est réalisée juste après l'admission d'un nouvel item dans le Processeur\_9, en exécutant le programme suivant :

```
Lire Item.Numéro_Reference
Si référence à produire différente de la référence en cours de production
alors mettre à jour :
  Processeur_5.2.Poids_requis
  Processeur_5.3.Poids_requis
  Processeur_5.4.Poids_requis
```

#### 4 ETUDE DE CAS DE LA CL DE L'OCP

L'Office Chérifien des Phosphates (OCP) est la plus grande entreprise Marocaine. Il est intégrée dans toute la chaîne de valeur des phosphates. La partie de la CL de l'OCP qui se trouve à Jorf (voir figure 7), est caractérisée par des processus de production continus (production d'acide phosphorique) et discrets (production d'engrais), ce qui lui confère les caractéristiques d'une CLH.

Ce site industriel comporte quatre ateliers successifs. L'atelier sulfurique, l'atelier de production de l'acide phosphorique concentré à 29, l'atelier de concentration de cet acide phosphorique à 54% et l'atelier des engrais.

La CL de l'OCP est caractérisée par deux modes de pilotage de la production. Les ateliers de production d'acide sulfurique et phosphorique fonctionnent à flux poussés tandis que l'atelier des engrais travaille à flux tirés en suivant un carnet des OF qui décrit les références et les quantités à produire, ainsi que leurs affectations sur les lignes. Le site industriel de Jorf est en interaction permanente avec d'autres entités : en amont avec l'approvisionnement du soufre et l'arrivée de la pulpe de phosphate, en aval par

un plan de chargement des navires et latéralement avec des joint-ventures (JV).

#### 4.1 Système décisionnel de Jorf et objectif du simulateur

Le comportement du système productif de Jorf est déterminé (sauf cas d'aléas) par l'ensemble des informations représentatives des différentes décisions prises sur l'horizon retenu. On s'intéresse ici aux décisions opérationnelles correspondant : au carnet des OF d'engrais à produire, à l'ouverture ou la fermeture des lignes de production ainsi que leurs régimes de marche, au programme de maintenance, et au programme d'échanges d'acide avec les JV.

Le simulateur a été développé dans l'optique d'amélioration des processus décisionnels de Jorf. Il permet d'anticiper l'impact des décisions prises, de détecter d'éventuelles incohérences décisionnelles, ainsi que d'évaluer la propagation des conséquences d'une décision prise au niveau d'un maillon sur les autres maillons de la CL.

Outre la détection d'éventuelles incohérences décisionnelles, le simulateur a pour vocation d'aider à la prise de décision, en facilitant l'exploration rapide de décisions alternatives permettant de stabiliser le comportement du système productif.

#### 4.2 Développement du simulateur

Dans le développement du simulateur, nous sommes partis de l'ensemble des règles de modélisations décrites en section 3 et nous les avons dupliquées autant de fois qu'elles sont appelées dans le modèle.

- La connexion entre les entités du SP est assurée par la primitive vanne que nous avons créée et dupliquée onze fois.

- Pour la modélisation des différents ateliers nous avons mobilisé la primitive de discrétisation du processus continu en nous l'avons dupliquée 39 fois (nombre total des lignes de production dans le site)

- Les différents ateliers sont connectés par des stocks capacitaires intermédiaires. La primitive stock capacitaire est utilisée 37 fois.

- Chaque processus, continu correspondant à une ligne, prélève du stock qui lui est lié moyennant un convertisseur. Pour la modélisation de ces connexions et la modélisation de l'ensemble des vannes nous avons mobilisé la primitive convertisseur et nous l'avons dupliquée 133 fois.

La granularité temporelle retenue dans la modélisation est de type opérationnelle (de l'ordre de l'heure ou de la dizaine de minutes). La simulation est effectuée sur un horizon de deux à trois semaines.

#### 4.3 Benchmark du simulateur

Nous avons effectué plusieurs benchmarks du simulateur pour le valider en partant de données réelles. Ces tests ont permis de détecter quelques incohérences décisionnelles, en particulier des arrêts non programmés des lignes de production induits par une rupture de stock en amont ou, surtout, par une saturation des stocks en aval (avec propagation de l'impact de l'arrêt sur l'amont). L'anticipation

de ces incohérences est difficile à faire par les méthodes classiques.

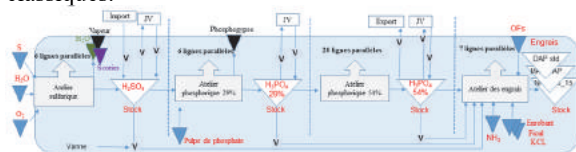


Figure 7 : chaîne logistique de Jorf

## 5 CONCLUSION

Dans cet article nous avons établi un ensemble de règles permettant d'utiliser les SED pour piloter une CLH, ce qu'aucun logiciel commercial, à notre connaissance, ne sait faire. Cette approche nous a permis de créer un outil d'analyse et d'aide à la prise de décision pour l'un des plus grands sites de production chimique d'une CLH phosphorique ; nous en avons rapidement montré l'architecture. L'usage de ces règles est en cours d'utilisation pour aider au pilotage d'autres activités de l'OCP.

## RÉFÉRENCES

- Andersson, M. and G. Olsson, 1998. A simulation based decision support approach for operational capacity planning in a customer order driven assembly line. *Proceedings of the 30th conference on Winter simulation*, p. 935–942.
- Anthony, R.N., 1965. *Planning and Control Systems: a framework for analysis*. Harvard University Press
- Barbosa, C. and A. Azevedo, 2017. Hybrid simulation for complex manufacturing value-chain environments. *Procedia Manufacturing*, vol. 11, p. 1404–1412, 2017.
- Chen, E.J., Y.M. Lee, and P. L. Selikson, 2002. A simulation study of logistics activities in a chemical plant, *Simulation Modelling Practice and Theory*, vol. 10, n° 3-4, p. 235–245.
- De Ugarte, B.S., A. Hajji., R. Pellerin, and A. Artiba, 2009. Development and integration of a reactive real-time decision support system in the aluminum industry. *Engineering Applications of Artificial Intelligence*, vol. 22, n° 6, p. 897–905.
- Degoun, M., P. Féniès., V. Giard., K. Retmi and J. Saadi, 2015. Propositions de règles de modélisation pour une simulation discrète d'une chaîne logistique hybride. Congrès international du génie industriel (CIGI2015). Québec, Canada.
- Doloi, H. K. and A. Jaafari, 2002. Toward a dynamic simulation model for strategic decision-making in life-cycle project management, *Project Management Journal*, vol. 33, n° 4, p. 23–23.
- Fabre, F., G. Hêtreux., J.-M. Le Lann, and P. Zaraté, 2011. Dynamic hybrid simulation of batch processes driven by a scheduling module. *Computers & Chemical Engineering*, vol. 35, n° 10, p. 2098–2112.
- Floudas, C.A. and X. Lin, 2004. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Computers & Chemical Engineering*, vol. 28, n° 11, p. 2109–2129.
- Forrester, J.W, 1961. *Industry dynamics*. Cambridge, Massachusetts
- Giambiasi, N. and J. C. Carmona, 2006. Generalized discrete event abstraction of continuous systems: GDEVS formalism. *Simulation Modelling Practice and Theory*, vol. 14, n° 1, p. 47–70, 2.
- Hamaidi, L., P. Bourseau., G. Muratet., O. Cerf, and A. K. Sadoudi, 1994. A mixed discrete-continuous approach for the simulation of bacterial contamination and growth in a food process. *Computers & chemical engineering*, vol. 18, p. S633–S637.
- Higuchi, T. and M. D. Troutt, 2004. Dynamic simulation of the supply chain for a short life cycle product—Lessons from the Tamagotchi case. *Computers & Operations Research*, vol. 31, n° 7, p. 1097–1114.
- Jamalnia, A. and A. Feili, 2013. A simulation testing and analysis of aggregate production planning strategies. *Production Planning & Control*, vol. 24, no 6, p. 423–448.
- Jahangirian, M., T. Eldabi., A. Naseer., Stergioulas, and T. Young, 2010. Simulation in manufacturing and business: A review. *European Journal of Operational Research*, vol. 203, no 1, p. 1–13
- Kleijnen J. P., 2005. Supply chain simulation tools and techniques: a survey. *International Journal of Simulation and Process Modelling*, vol. 1, n° 1-2, p. 82–89.
- Kyamakya, K. and K. Jobmann, 2005. Location management in cellular networks: classification of the most important paradigms, realistic simulation framework, and relative performance analysis. *IEEE transactions on vehicular technology*, vol. 54, n° 2, p. 687–708.
- Lee, Y. H., M. K. Cho, and Y. B. Kim, 2002. A discrete-continuous combined modeling approach for supply chain simulation. *Simulation*, vol. 78, n° 5, p. 321–329.
- Lehr, C. B., J.-H. Thun, and P. M. Milling, 2013. From waste to value—a system dynamics model for strategic decision-making in closed-loop supply chains. *International Journal of Production Research*, vol. 51, n° 13, p. 4105–4116.
- Lin, W.-J., Z.-B. Jiang., R. Liu, and L. Wang, 2014. The bullwhip effect in hybrid supply chain. *International Journal of Production Research*, vol. 52, n° 7, p. 2062–2084.
- Lyneis, J. M., 2000. System dynamics for market forecasting and structural analysis. *System Dynamics Review*, vol. 16, n° 1, p. 3.
- Meyer, M., H. Robinson., M. Fisher., A. Van der Merwe., G. Streicher., J.J. Van Rensburg., H. Van den Berg., E. Dreyer., J. Joubert, and G. Bonthuys., 2011. Innovative decision support in a petrochemical production environment », *Interfaces*, vol. 41, n° 1, p. 79–92.
- Mušič, G. and D. Matko, 1999. Combined simulation for process control: extension of a general purpose simulation tool. *Computers in Industry*, vol. 38, n° 2, p. 79–92.
- Owens, S.F. and R. R. Levary, 2002. Evaluating design alternatives of an extruded food production line using simulation. *Simulation*, vol. 78, n° 10, p. 626–632.
- Pierreval, H., R. Bruniaux, and C. Caux, 2007. A continuous simulation approach for supply chains in the automotive industry. *Simulation Modelling Practice and Theory*, vol. 15, n° 2, p. 185–198.
- Rabelo, L., H. Eskandari., T. Shaalan, and M. Helal, 2007. Value chain analysis using hybrid simulation and AHP », *International Journal of Production Economics*, vol. 105, n° 2, p. 536–547.
- Rozman, Č., A. Škraba., K. Pažek, and M. Kljajić, 2014. The development of sugar beet production and processing simulation model—A system dynamics approach to support decision-making processes. *Organizacija*, vol. 47, n° 2, p. 99–105.
- Scholz-Reiter, B., M. Freitag., C. De Beer, and T. Jagalski, 2005. Modelling dynamics of autonomous logistic processes: Discrete-event versus continuous approaches. *CIRP Annals-Manufacturing Technology*, vol. 54, n° 1, p. 413–416.
- Stchedroff, N. and R. C. Cheng, 2003. Supply chain management simulation: modelling a continuous process with discrete simulation techniques and its application to LNG supply chains », in *Proceedings of the 35th conference on Winter simulation: driving innovation*, p. 1607–1611.
- Tako A. A. and S. Robinson, 2012. The application of discrete event simulation and system dynamics in the logistics and supply chain context. *Decision support systems*, vol. 52, n° 4, p. 802–815.
- Venkateswaran, J. and Y.-J. Son, 2004. Distributed and hybrid simulations for manufacturing systems and integrated enterprise. in *IIE Annual Conference. Proceedings*, p. 1.
- Weston Jr, F.C., F. Herrmann, and P. H. Davidoff, 1999. Capacity planning and process analysis a simulation study of a microbrewery. *Production and Inventory Management Journal*, vol. 40, no 2, p. 48.