



HAL
open science

De la simulation dynamique hybride vers la conduite de procédés batch et semi-continus.

Gilles Hétreux, Raphaële Théry Hétreux, Nelly Olivier Maget, Jean-Marc Le Lann

► **To cite this version:**

Gilles Hétreux, Raphaële Théry Hétreux, Nelly Olivier Maget, Jean-Marc Le Lann. De la simulation dynamique hybride vers la conduite de procédés batch et semi-continus.. Journal Européen des Systèmes Automatisés (JESA), 2007, 41 (5), pp.585-616. 10.3166/jesa.41 . hal-03477863

HAL Id: hal-03477863

<https://hal.science/hal-03477863>

Submitted on 13 Dec 2021

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.

De la simulation dynamique hybride vers la conduite de procédés batch et semi-continus

G. Hétreux, R. Théry, N. Olivier, J.M. Le Lann

Laboratoire de Génie Chimique – INPT ENSIACET
5 rue Paulin Talabot,
31106 Toulouse Cedex 1, France

Gilles.Hetreux@ensiacet.fr, Raphaela.Thery@ensiacet.fr, Nelly.Olivier@ensiacet.fr

RÉSUMÉ. PrODHyS est un environnement de simulation dynamique hybride dont la particularité est de fournir des composants orientés objets généraux et réutilisables pour la modélisation d'appareils et d'opérations présents dans les procédés industriels. Les grands principes sur lesquels est fondé PrODHyS sont d'abord exposés en développant plus particulièrement les aspects relatifs au niveau commande. Ces concepts sont ensuite illustrés à travers la simulation d'un exemple académique de procédé batch permettant d'illustrer les potentialités du simulateur. Enfin, la dernière partie aborde brièvement des travaux en cours montrant l'exploitation de ce simulateur pour la conduite de ce type de procédés (ordonnancement et supervision) .

ABSTRACT. PrODHyS is an object-oriented environment dedicated to the hybrid dynamic simulation of industrial processes. The major aspects of the modelling approach proposed by this tool are detailed with a special attention to the command part. Then, these principles are illustrated through the modelling and the simulation of an academic example of batch plant which enables to illustrate the potentialities of the simulator. Finally, the use of this simulator for the scheduling and the monitoring of these processes are described.

MOTS-CLÉS : composants logiciels orientés objet, modélisation et simulation dynamique des systèmes hybrides, ordonnancement et supervision des procédés batch et semi-continus.

KEYWORDS: object-oriented software component, hybrid systems, dynamic simulation, scheduling and monitoring of batch and semi-continuous processes

1. Introduction

Aujourd'hui, la simulation dynamique constitue un outil d'étude essentiel pour l'analyse et la conception des procédés industriels. En effet, l'évolution rapide de la puissance des calculateurs et les avancées dans les stratégies de résolution des systèmes d'équations algèbro-différentielles, ont permis de valider des approches basées sur l'utilisation de modèles mathématiques rigoureux pour la représentation des opérations unitaires classiques de génie chimique. Dans ce contexte, l'unification

des travaux de recherche en modélisation et simulation de procédés menés depuis de nombreuses années au sein du Laboratoire de Génie Chimique a conduit au développement de la plate-forme de simulation dynamique hybride nommée *PrODHyS*. Si jusqu'à présent, cet environnement a été exploité pour simuler le fonctionnement d'opérations unitaires isolées, les dernières évolutions permettent désormais de traiter des procédés batch complets composés de plusieurs opérations unitaires complexes.

La première partie de cet article permet de présenter et d'illustrer les principes et concepts fondamentaux de *PrODHyS* par la mise en place de tels systèmes. Pour cela, la section 2 présente brièvement la structure de la bibliothèque et le formalisme hybride mise en œuvre (*Réseaux de Petri Différentiel Objet*). La section 3 décrit les principes de modélisation de la partie opérative du procédé, alors que la section 4 s'intéresse plus précisément au niveau commande. Dans la section 5, la modélisation d'un procédé académique permet de mettre en œuvre l'ensemble de ces éléments. Enfin, dans la section 6, les potentialités de la plateforme sont montrées au travers de travaux de recherche entrepris récemment, visant à exploiter la simulation dynamique hybride pour la *conduite* de ce type de procédés. Dans ce cadre, une première partie aborde la conception et le développement d'un outil d'ordonnancement fondé sur le couplage de ce simulateur avec un module d'optimisation, alors que la seconde concerne la *supervision* des procédés et plus précisément les fonctions de *surveillance* et de *diagnostic de fautes* en intégrant un modèle de simulation du procédé utilisé comme référence.

2. L'environnement *PrODHyS*

Après avoir montré l'intérêt de la simulation dynamique hybride pour l'analyse des procédés batch, les concepts fondamentaux sur lesquels est fondé *PrODHyS* sont abordés. Dans un second temps, le formalisme hybride mis en œuvre et la structure générale du modèle de simulation sont brièvement décrits.

2.1. Intérêts de la modélisation hybride pour la simulation dynamique des procédés discontinus

Les *procédés batch* et *semi-continus* sont souvent les systèmes privilégiés dans les industries du secteur pharmaceutique, cosmétique, de l'agroalimentaire, des polymères ou de la biochimie car ils permettent de fabriquer de petites quantités de produits variés et à haute valeur ajoutée en utilisant les mêmes ressources (équipements, opérateurs, cuves de stockage, matières premières, utilités, etc.). Si par nature, les phénomènes physiques évoluent de manière continue, la modélisation de ces systèmes fait souvent apparaître des aspects événementiels. En effet, la complexité de ces opérations conduit parfois à simplifier ou négliger certains phénomènes physiques. Il en résulte que les modèles dynamiques associés apparaissent alors comme des modèles continus par morceaux (par exemple, changement de phase de la matière) ou même discontinus (par exemple, la

commutation d'une vanne Tout ou Rien (*TOR*)). De plus, ces modèles dynamiques sont souvent à structure variable car le jeu d'équations constituant le modèle de simulation dépend généralement de l'état courant du système. Ceci nécessite donc un noyau capable de gérer automatiquement ces changements de configurations, point qui caractérise spécifiquement les *simulateurs dynamiques hybrides* (Zaytoon, 2001).

Par ailleurs, ces unités marient généralement des séries d'opérations qui fonctionnent en continu (qu'il s'agisse d'un régime transitoire ou permanent) avec des opérations purement discontinues. De plus, elles sont généralement constituées de nombreux appareils de traitement interconnectés et partagés. Il en résulte que la *recette* comporte une forte composante événementielle mais dépend aussi de paramètres à valeurs continues (quantités de matière utilisées, conditions opératoires telles que température, pression, concentration, etc.). De ce fait, elle est souvent définie par des événements d'état (seuil de température ou de composition par exemple) et non par des durées ou des dates d'occurrence fixées a priori. C'est pourquoi la simulation de telles opérations et des transformations physico-chimiques associées nécessite la résolution de modèles *phénoménologiques* hybrides.

Ainsi, les outils classiques de simulation dynamique continue ou de simulation à événements discrets d'ateliers s'avèrent mal adaptés à cette classe de problèmes et la capacité des simulateurs dynamiques hybrides à gérer simultanément et de manière cohérente les comportements continus et discrets devient un atout majeur.

2.2. Concepts fondamentaux de l'environnement PrODHyS

Une des bases fondamentales de *PrODHyS* est la représentation orientée objet des procédés industriels (Jourda *et al*, 1996, Sargousse, 1999, Moyse, 2000, Hétreux *et al*, 2002). *PrODHyS* est donc une bibliothèque de classes destinées à être dérivées en exploitant les mécanismes objets (polymorphisme, composition, héritage, généricité). Sa conception repose sur l'identification de *composants élémentaires généraux et réutilisables* dont l'assemblage permet de construire des modèles de simulation élémentaires, eux-mêmes étant des composants de modèles de simulation plus complexes et spécialisés. La détermination et la conception de ces briques élémentaires constituent un des premiers apports importants de ces travaux.

Le deuxième apport majeur est l'intégration dans *PrODHyS* d'un *noyau de simulation dynamique hybride* (Hétreux *et al*, 2003), (Perret, 2003) basé sur le formalisme *réseau de Petri différentiel objet* (*RdPDO*).

2.3. Architecture logicielle de l'environnement PrODHyS

PrODHyS fournit donc des composants logiciels permettant de traiter des opérations unitaires isolées ou couplées. Leur conception et leur développement s'appuient sur un processus unifié (*UP*), basé sur le formalisme *UML* et le langage objet *C++*.

Elle est organisée en 2 couches fonctionnelles indépendantes (*cf.* figure 1). La

couche simulation regroupe les classes correspondant au noyau de simulation (gestionnaire de simulation, formalisme *RdPDO*, intégrateurs *EDA*, solveurs *EANL*). Elle fournit donc les objets utiles à la simulation de tout système dynamique hybride, quel que soit le domaine d'application. La *couche modélisation* rassemble les classes utilisées spécifiquement pour la modélisation des procédés.

Actuellement, cette bibliothèque regroupe plus de 1000 classes réparties en 7 modules. Divers outils et *IHM* (Hétreux *et al.*, 2005) ont été développés (ou sont en cours de développement) selon le contexte d'utilisation de la bibliothèque (figure 1).

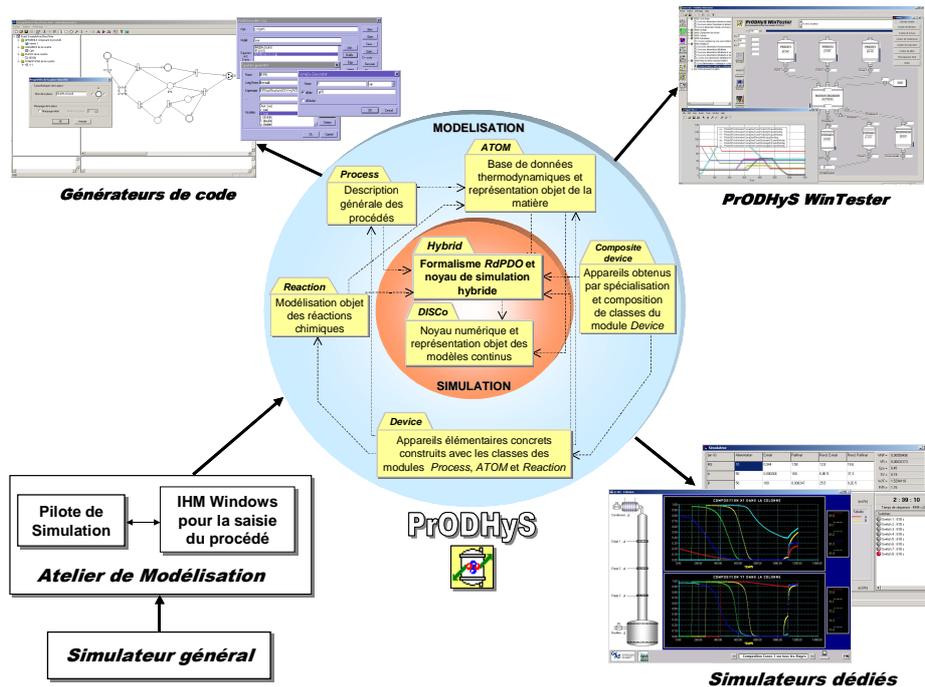


Figure 1. Environnement PrODHyS et outils associés

2.4. Le formalisme Réseau de Petri Différentiel Objet (RdPDO)

2.4.1. Principes et éléments du formalisme RdPDO

La modélisation et la simulation des systèmes hybrides a donné lieu à différents outils et formalismes (Zaytoon, 2001, Demongodin, 2001, Champagnat *et al.*, 1998, Andersson, 1994, Barton *et al.*, 1994, Nilsson, 1993, Stephanopoulos *et al.*, 1990, Piela, 1989). Lors de la conception du noyau de simulation de *PrODHyS*, il nous a semblé important de le fonder sur un formalisme mixte clairement défini permettant de décrire de manière aussi rigoureuse la dynamique continue que discrète. Si l'utilisation de systèmes d'équations algébro-différentielles paraît incontournable

pour la description des aspects continus, l'aspect séquentiel fort des systèmes considérés justifie l'emploi du modèle *réseau de Petri*. Dans ce cadre, les *Réseaux de Petri Différentiel Objet (RdPDO)* ont été mis en oeuvre pour spécifier le modèle de simulation associé à chaque composant (Hétreux *et al.*, 2003). Comme il n'est pas possible ici de développer de manière exhaustive les fondements théoriques de ce formalisme (voir Perret, 2003, Perret *et al.*, 2004), seuls sont présentés ici les principes élémentaires pour permettre d'appréhender la manière dont les modèles de simulation sont spécifiés via *ProDHYS*.

Indiquons d'abord que l'approche objet et les réseaux de Petri sont exploités selon une approche dite *combinée étendue*. Celle-ci consiste à les faire interagir de deux manières : la première est d'*introduire les objets dans les réseaux de Petri* et la seconde est d'*introduire les réseaux de Petri dans les objets*. Si la partie statique offre un haut niveau d'abstraction grâce aux concepts objets, la partie dynamique fait collaborer au sein d'une même structure des systèmes d'équations différentielles algébriques (*EDA*) pour décrire l'évolution continue du système, et des réseaux de Petri utilisés pour spécifier et gérer les séquences légales de commutation entre ces différents systèmes *EDA*.

2.4.2. Evolution dynamique dans les RdPDO

La figure 2 montre un exemple d'évolution dans les *RdPDO*. Lorsqu'un jeton marque une place différentielle, le système *EDA* est instancié par substitution des variables formelles par les attributs du jeton. La résolution induit l'évolution continue de ces attributs ainsi que l'évaluation de la condition associée à la transition (événements d'état ou de temps). Lorsque cette condition est atteinte, l'action est exécutée et la transition est franchie. Plusieurs jetons de même type peuvent marquer la place, déclenchant alors l'intégration du même système *EDA* à partir de la date de marquage. Par ailleurs, l'instanciation du système *EDA* d'une place différentielle peut nécessiter une combinaison de jetons de même type et/ou de types différents. Dans ce cas, la cohérence de l'ensemble doit être garantie par la modélisation ou validée a posteriori par la simulation. Notons toutefois qu'il s'agit là d'un problème difficile voire impossible qui reste ouvert.

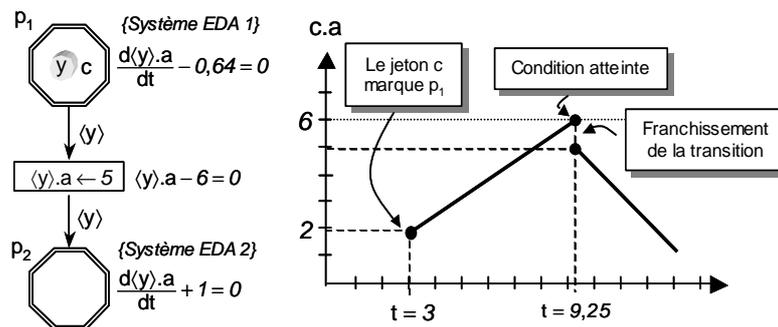


Figure 2. Principe d'évolution des *RdPDO*

Ainsi, le réseau de Petri peut être vu comme un moniteur d'EDA conduisant à la création dynamique d'un unique modèle de simulation dont la dimension et la structure fluctuent entre deux événements. Outre la résolution des systèmes EDA (basée sur une extension de la méthode de Gear (Gear, 1971)) et des modèles discrets (joueur spécifique à cette classe de RdP), le noyau effectue d'autres traitements indispensables à ce type de simulateurs tels que le calcul exact des instants de commutation, le recalage des états, la vérification de la cohérence des nouveaux modèles continus générés après commutation, la réinitialisation des variables d'état et de leurs dérivées.

2.5. Structure générale du modèle de simulation

Afin de pouvoir réaliser la simulation d'un procédé discontinu, il est nécessaire de modéliser à la fois la partie commande (le *superviseur*) et la partie opérative (le *procédé*). Dans PrODHyS (cf. figure 3), le modèle relatif au niveau *commande* (a priori spécifique à la recette, à la topologie du procédé considéré et au plan de production à réaliser) est complètement dissocié des modèles des appareils qui eux, doivent être réutilisables quelque soit le contexte (notion de *composant*).

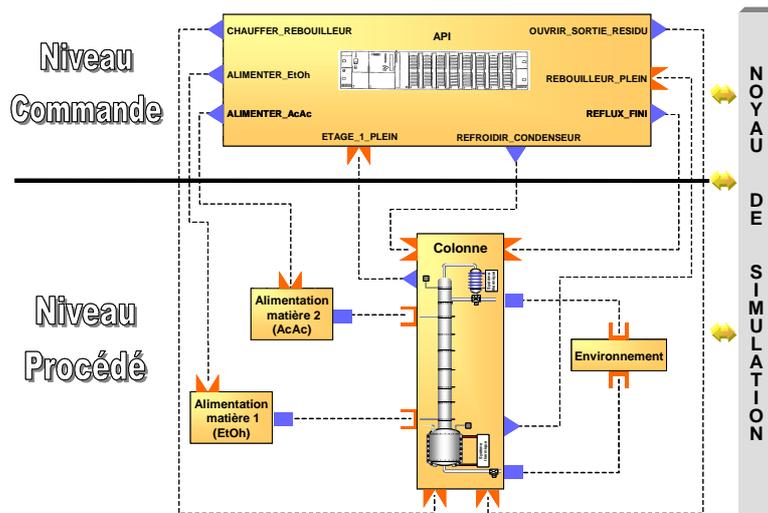


Figure 3. Modélisation topologique d'un procédé de distillation (niveau 1)

3. Modélisation de la partie 'opérative'

La modélisation des procédés (appareils et matière qu'ils contiennent) a déjà été présentée dans de précédentes communications (Hétreux *et al.*, 2004, Perret *et al.*, 2004, Perret, 2003). Dans la suite, seuls sont rappelés les principes essentiels de celle-ci. La spécification d'un procédé complet passe d'abord par la spécification de

chaque *entité élémentaire* composant le procédé. Remarquons cependant que ceux-ci sont généralement proposés par les modules *Process*, *Device* ou *CompositeDevice* et évitant ainsi à un utilisateur non modélisateur d'avoir à les spécifier. La spécification de la topologie du procédé complet est alors réalisé par un assemblage plus ou moins complexe de ces objets élémentaires conduisant à la définition de différents niveaux topologiques (cf. figure 3).

3.1. Modélisation d'un appareil élémentaire

Tout objet de procédé présent dans *PrODHyS* est caractérisé par (cf. figure 4) :

- un ensemble de *caractéristiques statiques* définies au moyen d'une configuration (dimensions, capacité, géométrie de l'appareil, etc.),
- un ensemble de *variables d'état continues* propres à l'entité (température, pression, composition, rétention, débit, etc.),
- le type et le nombre de *port* qu'il possède, sorte d'interface d'entrée/sortie à travers desquelles sont partagées les variables d'état entre plusieurs entités. Une variable propre est rendue accessible via un *port de sortie*, alors qu'une variable extérieure devient accessible à travers un *port d'entrée*. Ainsi, il n'existe aucune variable globale (principe d'*encapsulation*) et le partage d'une variable se fait toujours de manière explicite (*connexion* entre 2 ports).
- un ensemble de systèmes *EDA* composés des variables propres à l'entité et des variables partagées via les ports. Chaque modèle phénoménologique représente un comportement continu de l'entité (un morceau). Orientés *phénomène*, ils reposent sur les lois thermodynamiques et physico-chimiques ainsi que sur la notion de bilan (matière, énergie, etc.). Chaque élément de procédé est alors considéré comme une frontière de bilans. Nous parlons alors de *volume de contrôle* pour caractériser l'aspect *conteneur de phénomène* : un élément de procédé contient directement ou indirectement une portion de matière dans laquelle se produisent un ou plusieurs phénomènes physico-chimiques.
- un *RdPDO* qui décrit les séquences légales de commutation entre les systèmes *EDA* associés à chaque place différentielle. Rappelons cependant que l'approche combinée *Objet/RdP* permet de dissocier le modèle de la matière, de l'appareil qui la contient (notion de *container*). En l'occurrence, un jeton *matière* permet de gérer les équations du modèle liées à l'état physico-chimique et thermodynamique de la matière (au moyen d'un *RdPDO* propre) en fonction des conditions locales. De même avec les jetons *port* qui portent les variables de flux et de potentiel. Ces jetons « objet » présentent l'intérêt d'être réutilisables et de réduire notablement la complexité du *RdPDO* d'un appareil en ne faisant apparaître que les états inhérents strictement à celui-ci (et non en combinaison avec ceux de la matière qu'il contient, par exemple). Ces différents modèles ne sont finalement fusionnés qu'au moment de l'instanciation du modèle de simulation, en fonction de l'état courant du procédé.

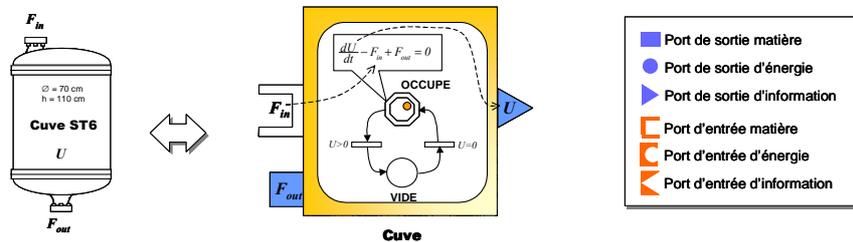


Figure 4. Exemple simplifié de modélisation d'un objet de procédé élémentaire

3.2. Modélisation d'un procédé complet

Il s'agit de représenter la structure du procédé selon une *vision système*, c'est à dire en terme de connexions entre diverses structures composées hiérarchiquement au travers desquelles circulent matière et énergie.

3.2.1 Modélisation topologique selon l'axe connexion

Il s'agit de décrire les *connexions* entre les différents éléments constitutifs du procédé (figure 3). Une connexion correspond à une visibilité offerte à un élément vers un autre par l'intermédiaire des *ports*. Elle s'établit par la mise en relation d'un port d'entrée et d'un port de sortie de même type. Toutefois, le modèle reste totalement acausal, la notion d'*entrée* ou de *sortie* permettant seulement de fixer le sens conventionnel du flux (positif si *sortie* → *entrée*). Selon cet axe, un procédé est donc représenté par un graphe orienté.

3.2.2 Modélisation topologique selon l'axe décomposition/composition

L'axe *décomposition/composition* permet de décrire la structure interne de chaque entité. Fondés sur le design pattern *Composite*, deux types initiaux d'éléments suffisent à son support : les *appareils élémentaires* (cf. section 3.1) qui sont non-décomposables et peuvent être spécialisés (fixent la granularité de la modélisation) et les *appareils composés* qui décrivent récursivement la décomposition. Selon cet axe, un procédé apparaît alors sous la forme d'un arbre dont les feuilles sont les appareils élémentaires et dont la racine est un objet général de type *Flowsheet*. Par ailleurs, chaque appareil élémentaire issu de la décomposition résulte de la spécialisation de composants encore plus généraux définis dans *PRODHYS* (Hétreux *et al*, 2002, Moysse, 2000).

4. Modélisation de la partie 'Commande'

L'évolution d'un procédé discontinu induit une succession de phases transitoires et de régimes continus. De ce fait, un niveau *commande* est explicitement défini afin de piloter le déroulement de la simulation. Pour cela, cette section décrit la structure du modèle du niveau commande qui doit par ailleurs, assurer la génération quasi-automatique du modèle de simulation à partir de la recette et du plan de production.

4.1. Connexion entre la partie commande et la partie opérative

Le modèle associé au niveau *commande* est un *RdPDO* maître dont l'évolution provoque l'évolution des *RdPDO* des entités du niveau *procédé*. Ce *RdPDO*, appelé *RdP Recette*, correspond en quelque sorte au programme GRAFCET qui serait exécuté par un automate programmable industriel (*API*) pour réaliser un cycle de travail (cf. figure 5). Les signaux échangés entre la partie *procédé* et la partie *commande* correspondent soit à l'émission d'une commande (signal *Open_V1*), soit à la réception d'une information provenant d'un capteur (signal *Signal_D1*). Un signal est modélisé par une place respectivement dite *de commande* ou *d'information* et son état est associé au marquage de cette place. Ces places sont le seul lien entre les appareils commandés et la recette. Considérés comme des boîtes noires, on distingue alors deux types de dispositif :

- les dispositifs *actifs* : appareils dont le *RdP* possède une ou plusieurs *places de commande* et/ou *d'information* tels que les vannes, les pompes, les alimentations d'énergie, les capteurs, etc (cf. VANNE V1, DETECTEUR D1),
- les dispositifs *passifs* : entités dont le *RdP* ne possède aucun lien direct avec le *RdP* recette tels que les cuves, les réacteurs ou la matière (cf. CUVE ST6).

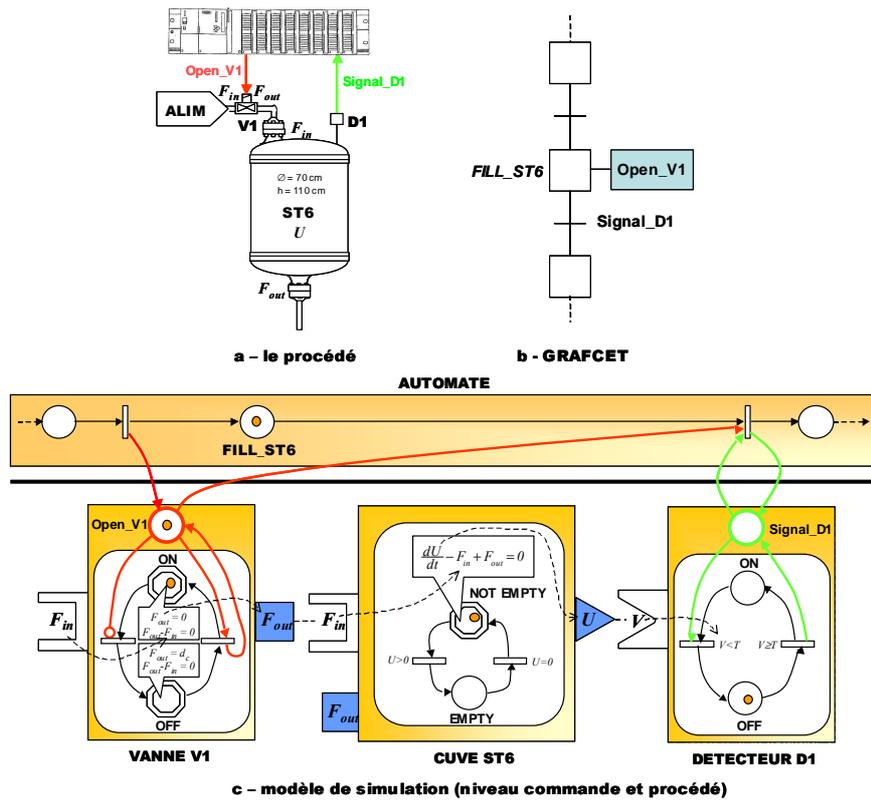


Figure 5. Interaction entre le niveau commande et le niveau procédé

Le marquage d'une place de commande d'un dispositif actif induit en général l'évolution de son *RdP*, lui-même provoquant éventuellement en cascade l'évolution des dispositifs passifs au travers du réseau constitué par la connexion des différents ports *matière* ou *énergie*. Par conséquent, l'évolution des différents modèles est conditionnée par deux types distincts d'événement :

- d'une part, des événements dits *externes* qui provoquent des *commutations commandées*. Il s'agit de tous les signaux échangés entre le niveau *commande* et le niveau *procédé* (soit les commandes émises depuis le *RdP* recette afin de piloter les dispositifs *actifs*, soit à l'occurrence d'un événement d'état (détection d'un seuil) ou temporel. Spécifiés par l'utilisateur, ces événements apparaissent explicitement sur le *RdP* recette.
- d'autre part, des événements dits *intrinsèques* dont l'occurrence dépend uniquement de l'évolution spontanée du procédé. Ces *commutations autonomes* correspondent au changement d'état de la matière (passage de l'état liquide à l'état liquide/vapeur lorsque la température d'ébullition est atteinte, par exemple) ou d'un dispositif *passif*. Ceux-ci n'apparaissent donc pas explicitement sur le *RdP* de la recette (l'utilisateur n'a donc pas à les spécifier) et sont traités exclusivement au sein du modèle de l'entité concernée.

Les interactions entre *RdP* *recette* et *RdP* *procédé* sont illustrées sur la figure 5. Celle-ci représente une séquence opératoire permettant d'alimenter un réacteur jusqu'à ce qu'un volume désiré soit atteint. L'opération de remplissage du réacteur est commandée par la recette qui envoie un jeton sur la place signal de l'appareil vanne. L'alimentation est maintenue ouverte tant que cette place de commande reste marquée. Afin de détecter l'instant où le volume souhaité de réactif est transféré, un détecteur de niveau est utilisé. Le marquage de la place de l'appareil détecteur indique que le volume de réactif a atteint la valeur cible. La transition est alors tirée. L'absence de jeton sur la place commande provoque alors la fermeture de la vanne.

4.2. Représentation hiérarchisée de la recette

La recette regroupe l'ensemble des informations utiles à la fabrication d'un produit. Cependant, lorsque le système de production possède de nombreux équipements ou si l'élaboration d'un produit nécessite de nombreuses phases de transformation, cette recette peut rapidement devenir complexe. C'est pourquoi la norme *ISA/SP88* (www.isa.org) propose une modélisation hiérarchisée de la recette et établit une approche standard pour traiter la complexité du contrôle/commande des procédés batch. De manière générale, une recette est constituée de quatre groupes d'information :

- l'*en-tête* : définit les informations administratives et d'identification,
- la *formule* : définit les paramètres à appliquer (ingrédients, proportions, conditions opératoires, etc.),
- la *procédure* : définit la suite de tâches ordonnées dans le temps à accomplir pour assurer la fabrication du produit,
- les *ressources* : définit les besoins en équipement pour la fabrication du produit.

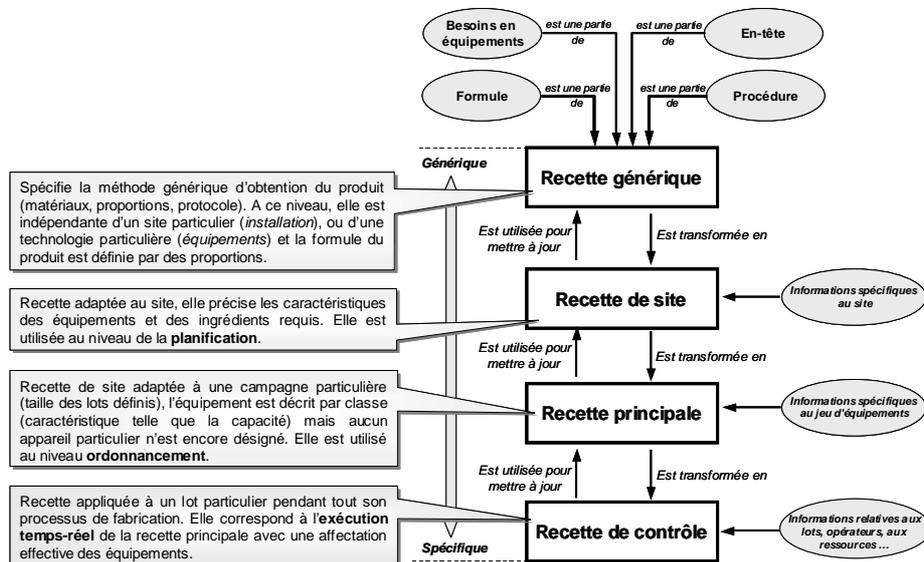


Figure 6. Hiérarchisation de la recette selon la norme ISA-SP88

Comme l'illustre la figure 6, quatre niveaux de recette sont définis, adaptés au niveau de finesse avec lequel le procédé est pris en compte. La *recette de contrôle* est une recette exécutable qui possède les informations nécessaires à la simulation (volume et composition des lots, équipements utilisés, conditions opératoires, etc.).

Cette structuration de la recette induit aussi une hiérarchisation en cinq niveaux des informations associées à la partie procédurale (figure 7) :

- la *procédure* définit la séquence d'opérations à exécuter sur une ligne de production pour fabriquer un produit particulier. Une opération correspond à la notion d'opération unitaire induisant une modification des propriétés physico-chimiques d'un produit (réaction, séparation, décantation, chauffe, etc.).
- chaque *opération* définit l'ensemble des phases exécutées en séquence ou en parallèle dans un équipement (réacteur, colonne, etc.).
- une *phase* est composée d'une succession de pas permettant de réaliser une fonction élémentaire. Elle est généralement délimitée par des frontières définies par les instants où le déroulement des pas peut être interrompu en toute sécurité pour l'installation et sans dommage pour le produit.
- un *pas* désigne une séquence temporelle ou événementielle d'instructions à exécuter dans un module d'équipement en vue d'accomplir une tâche spécifique.
- enfin, une *instruction* définit une commande élémentaire orientée équipement adressée à un élément ou à un composant afin qu'il effectue un traitement donné.

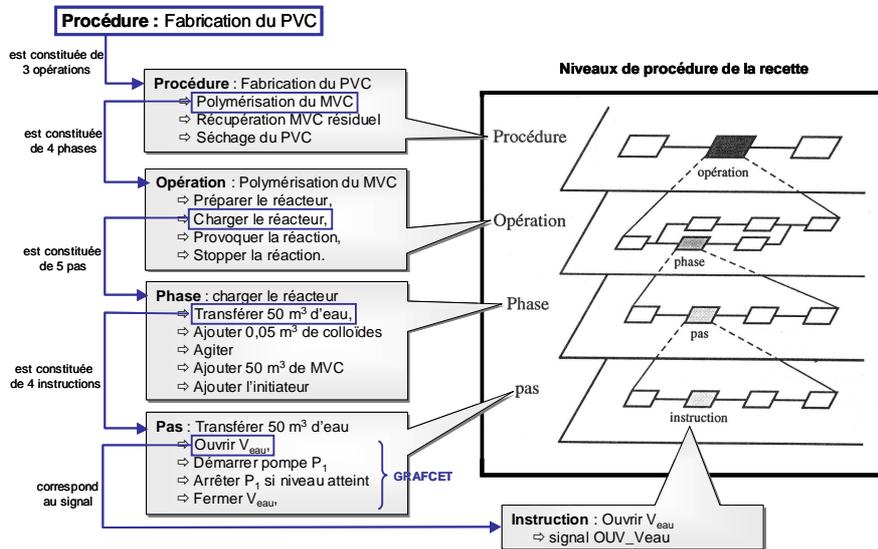


Figure 7. Décomposition de la procédure selon la norme ISA-SP88

4.3. Modélisation de la recette et de la procédure dans ProDHYS

S'appuyant sur la structure de recette préconisée par la norme ISA-SP88, la figure 8 montre un extrait de la modélisation U.M.L. de la recette mise en place dans ProDHYS.

La *recette générique* n'est pas implémentée actuellement car elle correspond à un niveau d'abstraction trop élevé pour les besoins de la simulation.

Le niveau *recette de site* est représenté par la classe SiteRecipe. Elle est constituée d'un lien vers le procédé et ses équipements constitutifs (classe Flowsheet), d'un lien vers le système matériel manipulé (classe ThermoUniverse) et d'un lien vers la procédure (classe Procedure). La recette de site fournit les informations nécessaires à la fonction de *planification* qui consiste à établir la liste d'ordres de fabrication à réaliser sur un horizon fixé (classes ManufacturingPlanning et ManufacturingOrder). Ces derniers définissent le volume de produits finis ou d'intermédiaires à fabriquer dans chaque période, compte tenu de la capacité agrégée de l'unité de production.

La notion de *recette principale* est représentée comme une spécialisation de la recette de site (MasterRecipe). Elle possède un lien vers la classe ManufacturingCampaign qui regroupe l'ensemble des informations relatives aux tâches impliquées dans la campagne de fabrication. Ces tâches résultent de la prise en compte de la procédure pour chaque OF à réaliser. Cette classe regroupe donc l'ensemble des informations nécessaires à la fonction *ordonnancement* qui doit effectuer un découpage de chaque OF en batch (détermination de leur nombre et de leur taille) compatibles avec la capacité de chaque ressource, et calculer les dates

de lancement de chacun de ces lots compte tenu de la procédure et de la disponibilité des ressources.

Enfin, la *recette de contrôle* regroupe les informations utiles à la réalisation des lots et au pilotage des équipements. Il s'agit notamment des conditions opératoires (température, pression, composition, etc.) utilisées pour définir les fonctions de transition sur événements d'état au sein des *RdPDO* du niveau *commande*.

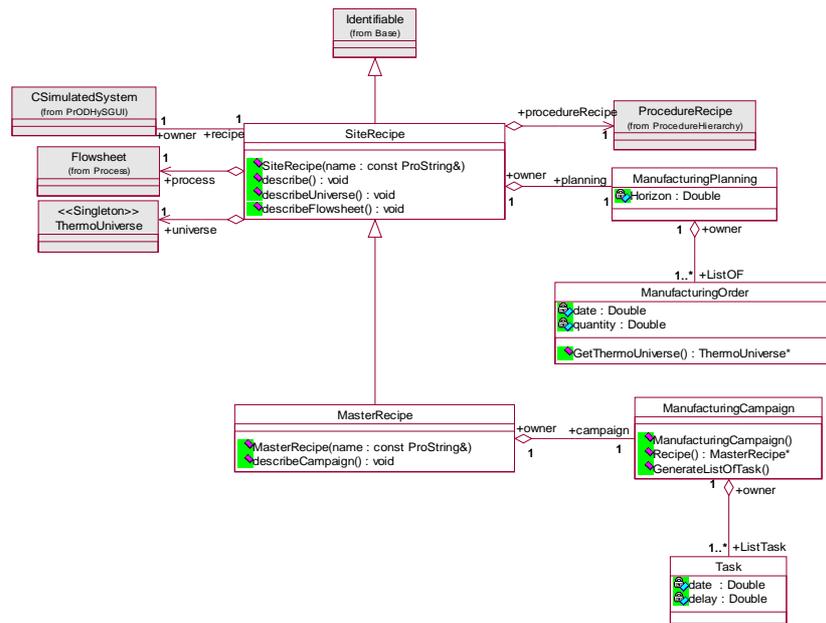


Figure 8. Extrait du diagramme de classes associé à la notion de recette

La figure 9 présente le modèle *U.M.L.* associé à l'aspect procédural de la recette.

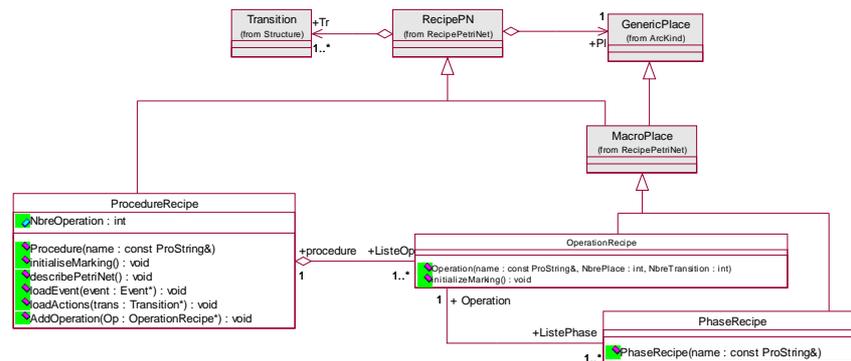


Figure 9. Extrait du diagramme de classes associé à la notion de procédure

Ici encore, la décomposition présentée précédemment a été adoptée pour la modélisation de la procédure sous *PrODHyS*. Dans ce cadre, la notion de *macroplace* a été introduite. Elle permet de remplacer une séquence de places/transitions relatives à une *opération* ou à une *phase* par une unique macro-place (figure 10). Cette séquence est alors délimitée par 2 places particulières **E** et **S** entre lesquelles les différents types de places définis dans les *RdPDO* peuvent apparaître (y compris d'autres macro-places).

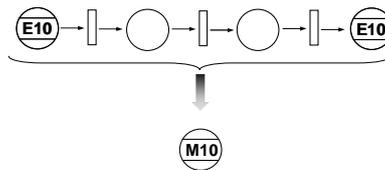


Figure 10. *Macro-place sous PrODHyS*

On retrouve ainsi la classe `ProcedureRecipe` définie comme une séquence d'opérations (classe `OperationRecipe`), elles-mêmes définies comme un ensemble de phases (classe `PhaseRecipe`). Une *phase* est alors définie par un *RdPDO* sans macro-place, où chaque place correspond à la notion de *pas*. La notion d'*instruction* est, dans notre cas, souvent confondue avec la notion de *pas*.

5. Application à un procédé didactique

Les différents éléments présentés dans les sections précédentes sont illustrés au travers de la modélisation et la simulation d'un procédé académique, typique des installations batch trouvées en milieu industriel.

5.1. Caractéristiques de la recette associée au procédé considéré

5.1.1 Formulation

Le but de l'installation est de fabriquer et de conditionner un produit *P* dont la pureté doit être au moins égale à 98 % molaire. La réaction considérée est une réaction équilibrée endothermique impliquant deux réactifs *R1* et *R2* et le produit *P* et qui suit une loi cinétique d'ordre 2 :



Comme le montre la figure 11, la fraction molaire de produit *P* ne peut excéder 0,83 et un temps de séjour infini serait nécessaire pour atteindre cette composition. Afin de maximiser le taux de conversion de la réaction sans pour autant pénaliser le temps de cycle du procédé, la réaction est stoppée dès que la composition en produit *P* atteint la valeur de 0,8. Par ailleurs, compte tenu de la loi de vitesse considérée, la

réaction (R) est d'autant plus rapide que la température T est élevée. La température retenue pour la réaction doit garantir une réaction rapide tout en maintenant les constituants dans l'état liquide. Une température de 383 K permet de satisfaire ces 2 contraintes. La spécification de pureté imposée pour le produit P étant égale à 98%, une étape de distillation est donc nécessaire.

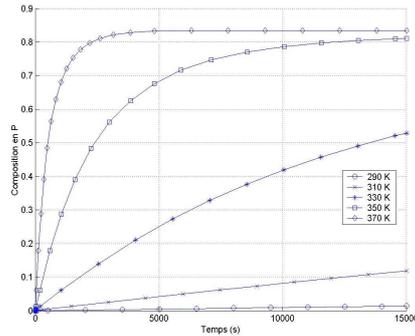


Figure 11. Vitesse de la réaction en fonction de la température

5.1.2. Procédure

La procédure de fabrication du produit P est représentée sur la figure 12 sous la forme d'un *S.T.N.* (*State Task Network*) générique.

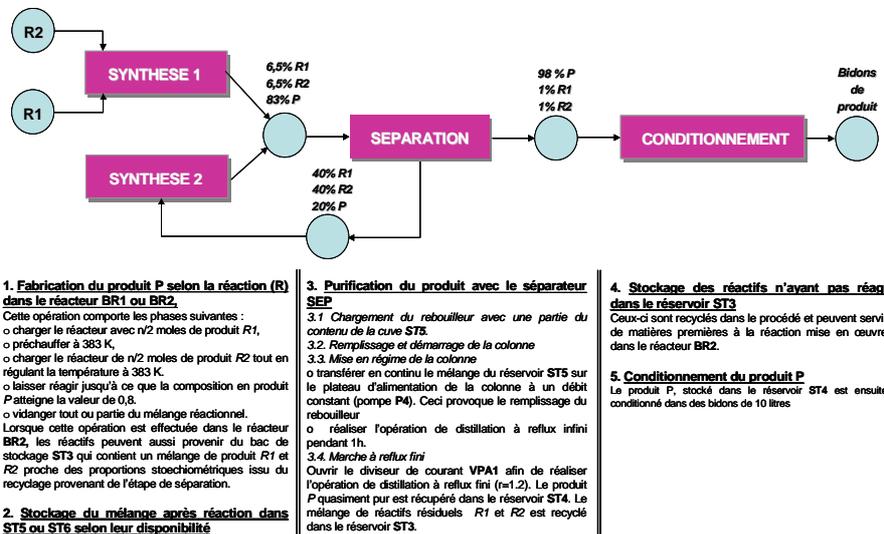


Figure 12. Procédure de fabrication du produit P

5.1.3. Ressources

L'unité de production est décrite sur la figure 13. Ce procédé combine des opérations discontinues (synthèses, conditionnement) et des opérations semi-continues (séparation). Chaque opération comprend des phases de transfert amont et aval avec les cuves tampon (par exemple, l'opération **SYNTHESE 1** intègre les transferts depuis **ST1** et **ST2** en amont et **ST5**, **ST6** en aval). Lorsque plusieurs cuves sont affectées au stockage, une politique spécifique de gestion de ces cuves doit être définie (dans notre cas, gestion de **ST5** et **ST6**).

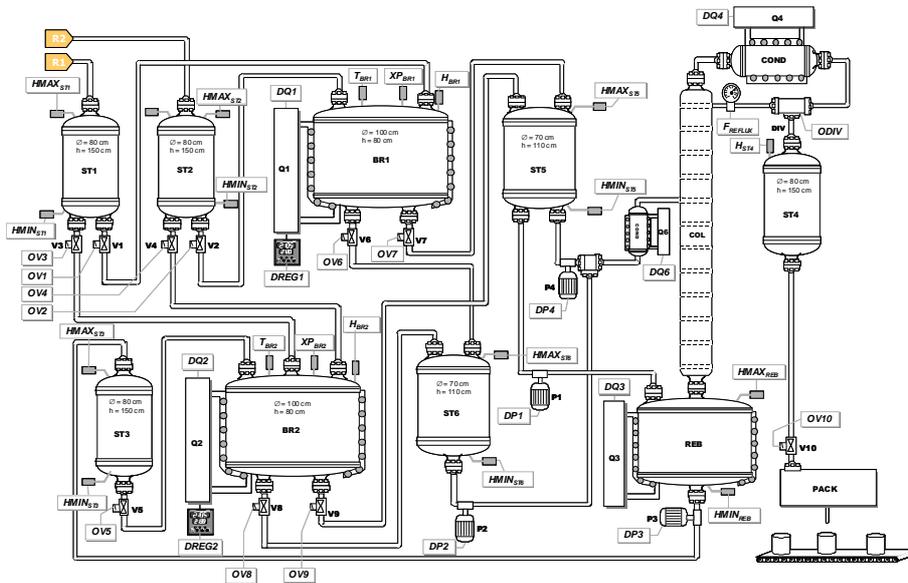


Figure 13. Schéma du procédé

5.2. Modélisation de la partie opérative

A titre d'illustration, la figure 14 montre la structure topologique interne de l'objet colonne selon l'axe composition/décomposition. Une description plus complète de la modélisation de cet objet peut être trouvée dans (Théry et al., 2002). Par ailleurs, la modélisation des autres équipements composant ce procédé est donnée dans (Perret, 2003).

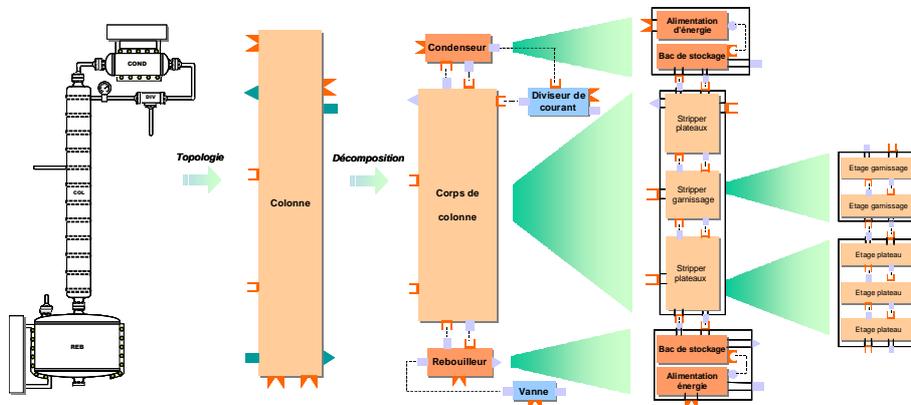


Figure 14. Structure de l'objet colonne (niveaux topologiques 2, 3 et 4)

5.3. Modélisation de la partie commande

Il n'est pas possible de donner ici le modèle complet du niveau *commande* associé à ce procédé. Aussi, seuls les *RdPDO* découlant de la décomposition de la procédure selon les différents niveaux qui correspondent à l'opération de séparation sont décrits dans la suite.

5.3.1. Le niveau procédure

La figure 15 montre le modèle associé à la procédure (spécifique à un mode de production fixé). Celle-ci est constituée des 4 opérations unitaires, chacune étant représentée par une *macro-place* : synthèse dans **BR1**, synthèse dans **BR2**, séparation, conditionnement et gestion des cuves de stockage **ST5** et **ST6**.

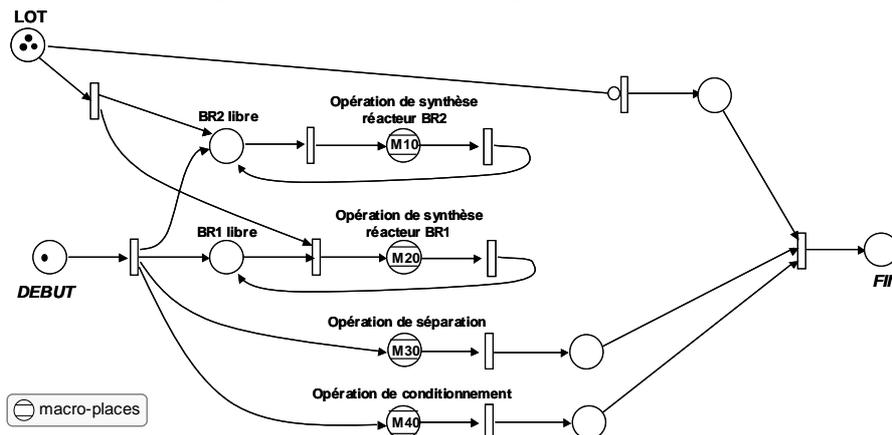


Figure 15. Réseau de Petri décrivant la procédure

Dans ce cadre, le lien prévu vers un objet de type `MasterRecipe` trouve ici sa justification car l'établissement de la procédure requiert la connaissance de l'ensemble des informations utiles pour la fabrication du produit (gammes de fabrication, équipements nécessaires).

5.3.2. Le niveau opération

Ce niveau de description de la procédure rejoint la notion d'opération unitaire couramment rencontrée en génie des procédés. La figure 16 représente l'opération de séparation, associée à la macro-place **M30** dans le *RdP* de la procédure. L'expansion de **M30** se compose d'une *phase composée* de distillation représentée par une *macro-place* notée **M300** et de *phases élémentaires* (non décomposables) représentées par une place (place nommée **FILL_REB**, par exemple).

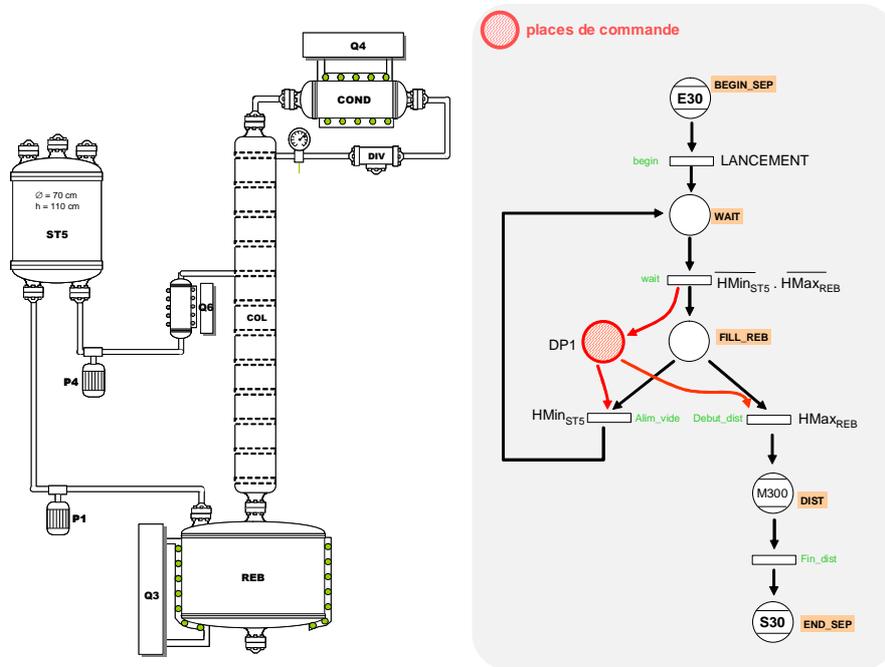


Figure 16. Expansion de la macro-place associée à l'opération de séparation

5.3.3. Le niveau phase

Une *phase* est une succession de *pas* permettant de réaliser une fonction élémentaire. La figure 17 représente l'expansion la macro-place **M300** associée à la phase de distillation (cf. figure 16). Le *RdPDO* représentant le fonctionnement

intrinsèque de la colonne est aussi représenté. Comme nous pouvons le constater les évolutions respectives de ces deux *RdP* sont intimement liées :

- l'évolution du *RdP* associé à la phase **M300** active des places de commande permettant l'évolution du *RdP* intrinsèque de la colonne.
- ce dernier permet lui-même de faire évoluer les *RdP* représentant l'évolution des objets composant la colonne (ici, le condenseur et le rebouilleur).
- inversement, l'évolution du *RdP* intrinsèque de la colonne active des places d'information qui permettent l'évolution du *RdP* de la phase de distillation.

Précisons par ailleurs, qu'au niveau de la phase, les *RdP* sont définis de manière générique afin de permettre leur réutilisation dans un autre contexte.

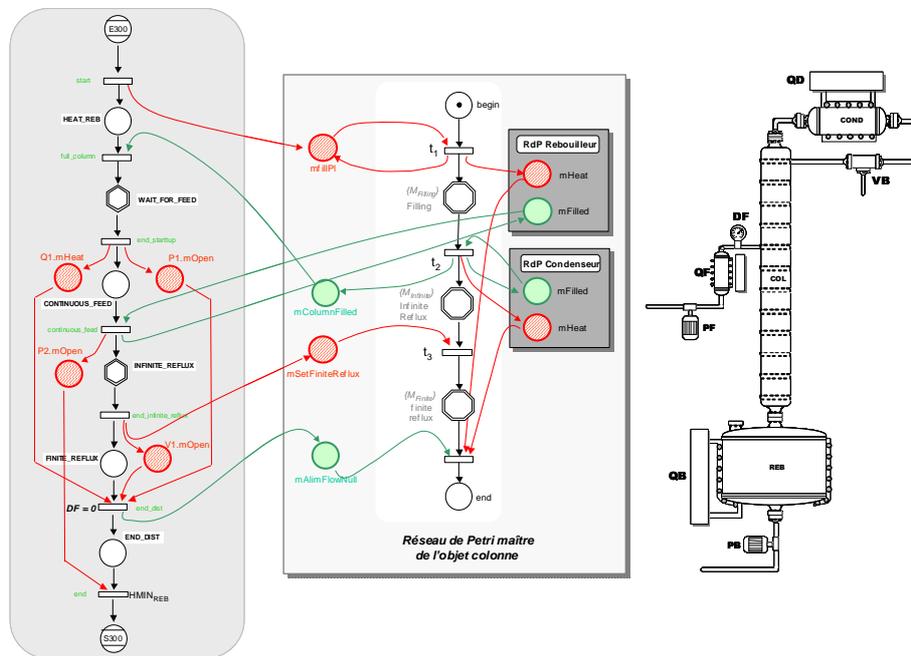


Figure 17. Expansion de la macro-place **M300** représentant la phase de distillation et connexion avec les places de commande et d'information de l'objet colonne

5.4. Analyse d'un procédé discontinu par simulation dynamique hybride : Intérêts et limites

5.4.1. Intérêt de la capacité du simulateur à gérer les événements d'état

Les modèles continus associés à chaque objet constituant ce procédé prennent en compte, entre autres, les bilans matière globaux et partiels, le bilan d'énergie, les équilibres liquide/vapeur, les vitesses de réaction, ainsi que les phénomènes

hydrauliques. En effet, hormis les conduites munies d'une pompe, les transferts entre cuves sont effectués par gravité. A titre d'illustration, la figure 18 donne une partie de l'évolution de composition au sein du réacteur **BR1** durant l'opération de réaction.

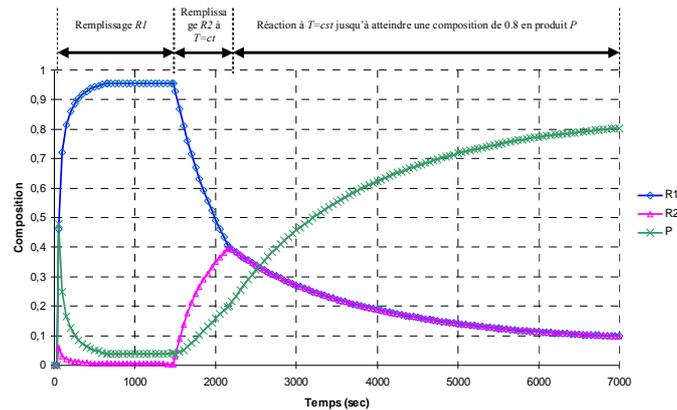


Figure 18. Evolution de la composition dans le réacteur **BR1**

La capacité du simulateur à gérer les événements d'état est un atout majeur car cela permet de :

- prendre en compte des événements dont la date d'occurrence n'est pas connue a priori,
- représenter avec plus de précision le comportement du procédé puisqu'il s'appuie sur la simulation des phénomènes physiques via des modèles prenant en compte les conditions opératoires (température, pression, etc.)

En effet, l'hypothèse d'une durée constante et connue des tâches constitue une restriction sévère face à la sensibilité de certaines opérations unitaires à ces paramètres. La colonne à distiller discontinue est un autre exemple où la durée opératoire est liée à plusieurs facteurs (qualité de la charge initiale, politique de chauffe au bouilleur, reflux en tête de colonne, débits de soutirage latéraux, pertes thermiques, etc). De plus, la durée d'une tâche peut aussi dépendre de l'état du système à un instant donné. Par exemple, la durée de chauffe d'un produit dépend de sa température initiale, elle-même pouvant dépendre de la durée d'attente du produit dans la cuve de stockage amont si des pertes thermiques existent. De même, lorsque les transferts entre cuves sont opérés par gravité, le temps de transfert dépend du débit qui devient une fonction du niveau de produit dans la cuve source à cause des phénomènes de pression hydraulique. L'estimation de cette durée est rendue encore plus délicate si des alimentations et des vidanges sont autorisées simultanément. Ne pas inclure ces paramètres dans la représentation du système impose une approximation de la durée opératoire.

Afin d'évaluer l'impact de cet aspect, une série de simulations conduites successivement sur événements de temps et sur événements d'état a été réalisée pour un procédé comportant des phases de transfert par gravité entre cuves de même géométrie. Les durées de transfert ont été évaluées en supposant successivement que le débit est minimum (fond de cuve), maximum (cuve pleine) ou moyen (milieu de cuve). Les résultats obtenus pour les différents scénarii envisagés sont reportés dans le tableau 1. Les écarts sur les temps de cycle entre les 2 types de simulation sont compris entre 3,5% (débit minimum) et 14,6% (débit maximum). En considérant le débit moyen, l'écart est de 5,6%. On constate donc que les écarts ne sont pas toujours négligeables et peuvent nuire à la qualité des résultats obtenus, notamment si l'on cherche à optimiser le fonctionnement de l'unité.

	Évènements temporels			Évènements d'états
	Minimum	Moyenne	Maximum	
Temps de cycle	3175	3474	3771	3289

Tableau 1. Comparaison entre simulation sur événements temporel et événements d'état

5.4.2. Détermination de la taille des lots et politique de transfert des lots

De façon générale, la politique de transfert des lots entre appareils est un facteur important de productivité d'une unité batch. En effet, la possibilité de pouvoir accumuler des produits intermédiaires dans des cuves tampon de capacité finie (telles que **ST1**, **ST2**, **ST3**, **ST4**, **ST5** et **ST6**) offre un maximum de souplesse dans la gestion des lots en découplant les phases de vidange et de remplissage de deux opérations de transformation successives. Une cuve tampon permet aussi de modifier le volume du lot entre les opérations amont et aval, avec un choix de volume appartenant à une gamme continue. Or, la taille et le nombre de lots nécessaire pour satisfaire un objectif de production est souvent une donnée d'entrée car ces paramètres sont difficiles à établir seulement par simulation.

5.4.3. Choix dans l'affectation des ressources

Dans un atelier où plusieurs ressources peuvent réaliser la même tâche, le choix dans l'affectation d'une ressource peut influencer sensiblement les performances de l'atelier. Or, choisir d'utiliser le réacteur **BR1** ou **BR2** à un instant donné, par exemple, peut dépendre de la disponibilité des matières premières dans les cuves de stockage amont (**ST1**, **ST2**, **ST3**), mais aussi de la capacité résiduelle des cuves de stockage aval (**ST5** et **ST6**). Là encore, l'affectation des tâches aux ressources est souvent une donnée d'entrée car la simulation dispose de peu de moyen pour faire ce choix judicieusement.

5.4.4. Transition entre mode de production

La présence d'un réservoir tampon représente la configuration la plus classique pour assurer la transition entre deux zones de procédé ayant un mode de production différent. Cependant, la gestion de ces transitions est particulièrement délicate en cours de simulation.

En effet, lorsque la matière circule de la zone discontinue vers la zone continue du procédé, l'activité de la zone aval est conditionnée à l'existence d'une quantité suffisante de matière dans le réservoir. Si le débit de soutirage de la zone continue provoque une baisse de niveau permanente du réservoir qui n'est pas compensée par l'arrivée des lots de la zone amont, une rupture de production de la zone continue est inéluctable au bout d'un certain temps. Dans notre exemple, les cuves **ST5** et **ST6** jouent précisément ce rôle de régulateur en assurant un flux continu de matière entre la zone réactionnelle qui fonctionne en mode batch et la colonne de distillation qui fonctionne en continu, une fois démarrée. La gestion de telles situations est particulièrement importante dans la mesure où la remise en régime d'une installation continue (telle qu'une colonne de distillation) est souvent longue et consommatrice d'énergie. A l'inverse, si le soutirage continu est trop faible, il y aura dépassement de la capacité de stockage et l'activité de l'atelier discontinu devra être freinée. Cette situation qui semble plus facile à gérer peut toutefois être délicate lorsque les lots ne peuvent patienter dans les appareils que durant un temps limité. Dans ces configurations, la détermination puis le respect des dates de lancement d'opération dans la zone discontinue devient une contrainte prépondérante.

Pour un sens de circulation allant du continu vers le discontinu, la gestion du réservoir tampon obéit aux mêmes principes. En effet, la rupture de production du procédé continu est déclenchée par l'atteinte d'une valeur seuil maximale de la capacité du réservoir. Inversement, l'absence de quantité de matière suffisante dans le réservoir pour lancer la fabrication d'un lot retarde son démarrage et diminue la productivité de la chaîne discontinue. Ceci correspond typiquement aux situations qui doivent être gérées au niveau des cuves **ST3** et **ST4**.

6. Exploitation de la simulation dynamique pour la conduite des procédés

Cette dernière section présente des travaux en cours exploitant le simulateur *PrODHyS*. La première application concerne la conception et le développement d'un outil d'ordonnancement basé sur le couplage de ce simulateur avec un module d'optimisation de type *MILP*. La deuxième application intègre le simulateur au sein d'un système de supervision afin d'assurer les fonctions de surveillance et de diagnostic.

6.1. Vers un ordonnancement réactif des procédés

6.1.1. Motivations du couplage simulation/optimisation pour l'ordonnancement

Gérer les problèmes soulevés dans la section 5.4 en cours de simulation apparaît donc difficile. L'idée est donc de coupler la simulation avec un module

d'optimisation s'appuyant sur un modèle mathématique plus apte à résoudre ces situations. Le rôle de ce dernier est donc de fournir les éléments de décisions qui manquent à la simulation pour arriver à son terme et non d'explorer l'ensemble des solutions en vue d'établir un ordonnancement « optimal ».

Cependant, dans un contexte plus général, l'exploitation rationnelle de la flexibilité intrinsèque des procédés batch confère à la fonction *ordonnancement* un rôle crucial. De nombreux travaux dans la littérature traitent ce problème reconnu complexe et différentes approches ont été proposées. Parmi celles-ci, les méthodes d'optimisation basées sur des formulations *MILP (Mixed Integer Linear Programming)* ou *MINLP (Mixed Integer Non Linear Programming)* sont souvent mises en œuvre. Cependant, comme l'indique (Mendez *et al.*, 2006), il subsiste un écart entre théorie et pratique, soit à cause de la taille des systèmes considérés, soit à cause des hypothèses simplificatrices qu'il faut éventuellement introduire pour rendre le problème solvable (l'une et l'autre étant d'ailleurs souvent liées). D'un autre côté, de nombreux facteurs induisent que la recherche d'un optimum mathématique sur des modèles trop simplifiés paraît souvent inutile dans la pratique. En effet, l'optimalité d'une telle solution est souvent perdue à cause de la nature dynamique de ces systèmes industriels (aléas, évolution de la demande, pannes, etc.). Par ailleurs, la mise en œuvre d'une solution est rarement directe dès lors qu'elle est confrontée aux caractéristiques du procédé réel (durées opératoires approximées, hypothèses simplificatrices, etc.).

Lorsque la garantie d'optimalité n'est plus le critère central, certaines approches proposées consistent à établir une solution initiale a priori « sous-optimale » (par rapport à certains critères) que l'on tente ensuite d'améliorer progressivement et ponctuellement par différentes techniques (Méndez et Cerda, 2003), (Roslöf *et al.*, 2001). La solution initiale est obtenue dans un temps fixé ou pour un effort de calcul donné, soit par une méthode exacte, soit par une méta-heuristique. D'autres travaux sont fondés sur le couplage d'un algorithme génétique avec un simulateur à événements discrets (Baudet *et al.*, 1995).

	Ordonnancement basé sur l'optimisation	Ordonnancement basé sur la simulation
Avantages	<ul style="list-style-type: none"> • Exploration exhaustive des solutions candidates • Prise en compte globale des contraintes • Méthodes efficaces de résolution 	<ul style="list-style-type: none"> • Modélisation souvent plus fidèle du procédé réel • Durées opératoires évaluées par des modèles phénoménologiques
Inconvénients	<ul style="list-style-type: none"> • Modélisation souvent basée sur des hypothèses simplificatrice ne permettant plus d'exploiter toute la flexibilité de l'atelier • Durées opératoires fixées et souvent surestimées 	<ul style="list-style-type: none"> • Evaluation d'une solution candidate ⇔ simulation du procédé pour un séquençement et une taille de lot donnés ↳ <i>Exploration difficile et coûteuse en temps CPU</i> • Difficultés à prendre en compte les contraintes de date (nowait, calendrier au conditionnement, politique de nettoyage) car vision partielle de l'horizon temporel

Tableau 2. *Optimisation/Simulation pour l'ordonnancement*

Pour notre part, l'outil d'ordonnancement proposé s'appuie sur ce principe d'amélioration à travers le couplage d'un module d'optimisation basé sur une modélisation de type *MILP* avec notre simulateur dynamique hybride. Le tableau 2 récapitule les principaux points forts et points faibles de ces deux outils que leur association peut compenser.

6.1.2 Processus de décision et structure de l'outil d'ordonnancement proposé

L'établissement d'un plan est basé sur un processus itératif entre le module d'optimisation *ProSched* et le simulateur *ProDHYS* dans lequel l'opérateur est intégré (cf. figure 19).

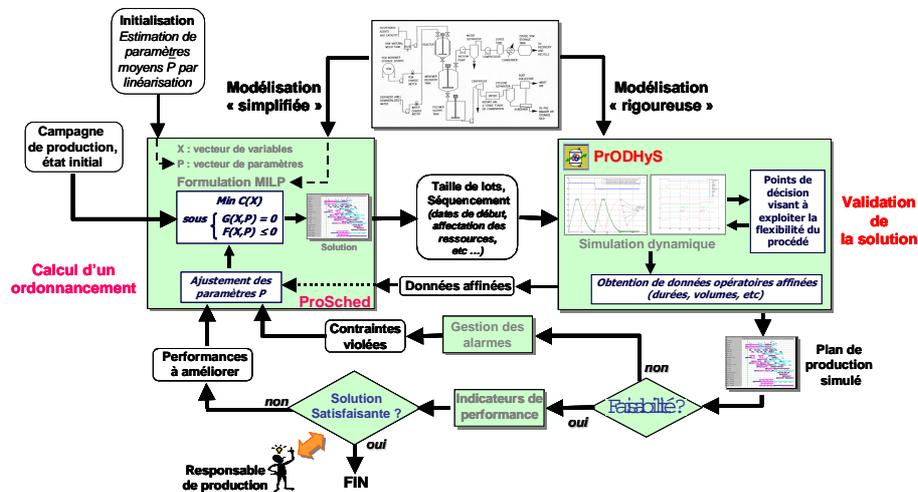


Figure 19. Structure de l'outil d'ordonnancement

Le module d'optimisation établit des plans de production sur la base d'une modélisation *MILP* (résolution avec le solveur *XPRESS-MP*). Ce premier modèle est construit à partir de durées moyennes estimées. Le séquencement et l'affectation des tâches sur les appareils ainsi que le volume des lots sont alors transmis au simulateur qui génère la recette correspondante et simule le fonctionnement dynamique du procédé. La faisabilité du plan est établie (ou pas) avec transmission en retour de durées opératoires affinées et d'indicateurs de performances (temps total d'exécution, taux d'occupation des ressources, etc.). Cette procédure itérative est effectuée jusqu'à ce que l'opérateur estime la solution satisfaisante. Par ailleurs, la reproductibilité du séquencement des lots sur les équipements entre deux phases d'optimisation peut constituer un indicateur de convergence de la méthode. Le principe fondamental de l'approche est donc de supposer que la solution « approchée » (en terme de comportement) établie par le modèle d'optimisation avec un effort de calcul réduit est compensée par la modélisation plus fine du procédé réalisé au niveau simulation, notamment en prenant en compte des modes de

fonctionnement particuliers (chevauchement des transferts, par exemple). Par ailleurs, cette approche devrait globalement rendre plus robuste les plans de production établis et en facilite l'analyse physico-chimique des phénomènes qui s'y déroulent.

Compte tenu de l'ampleur du travail nécessaire pour automatiser cette procédure, ce système est encore en cours de développement. Toutefois, la formulation *MILP* du modèle d'optimisation est décrite dans la section suivante.

6.1.3 Brève présentation du module d'optimisation ProSched

Diverses formulations *MILP* pour le problème d'ordonnancement des procédés batch peuvent être trouvées dans la littérature (Shah *et al.*, 1996), (Floudas et Lin, 2004), (Burkard et Hatzl, 2005), (Shaik *et al.*, 2005), (Méndez *et al.*, 2006). Compte tenu de la procédure à mettre en œuvre, les formulations *MILP* en temps continu apparaissent les mieux adaptées à nos besoins. Une comparaison détaillée de ce type de modèle peut être consultée dans (Shaik *et al.*, 2005). Plus précisément, une formulation du type *Unit-Specific Event-Based* proposée dans (Ierapetritou et Floudas, 1998) a été implémentée. Ce modèle d'optimisation permet :

- d'une part, de calculer simultanément les dates de début (ainsi que leur ordre de passage sur les équipements) et la taille de lots de chaque tâche,
- d'autre part, de prendre en compte des durées de tâche dépendantes de la taille de lot.

La figure 20 montre un schéma de principe d'un ordonnancement selon cette modélisation.

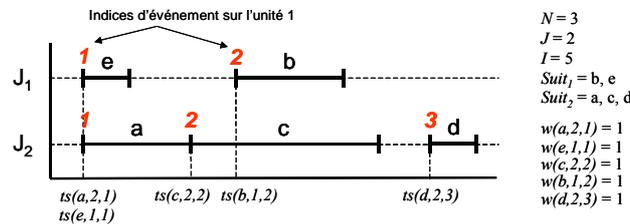


Figure 20. Variables de décision dans la formulation *Unit-Specific Event-Based*

Le tableau 3 répertorie les notations utilisées pour formuler le modèle de ce problème. Ce modèle est établi en s'appuyant sur la représentation *S.T.N.* (*State Task Network*) montrée sur la figure 21. Notons que seules les équations fondamentales du modèle sont reportées ci-après. La fonction objectif correspond à la minimisation du makespan par les contraintes (A.1) et (A.2). La satisfaction de la demande est définie dans (A.3), les contraintes (A.4), (A.5) et (A.6) correspondent respectivement aux contraintes d'allocation, de capacité et de bilan matière. Les contraintes de séquence sont (A.7) et (A.8), celles de stockage sont définies en (A.9) et enfin (A.10) regroupe les contraintes additionnelles accélérant la convergence.

L'ordonnancement obtenu par la résolution de ce modèle permet d'obtenir entre autres, les dates de lancement des lots au niveau de la simulation qui évitent une

rupture d'alimentation de la colonne en régime continu. Notre premier objectif (mené à son terme la simulation en respectant toutes les contraintes de production) est donc atteint (Hétreux *et al.*, 2007). En revanche, la phase de rebouclage entre simulation et module d'optimisation (démarche d'amélioration) n'a pas encore été implémentée actuellement au sein de notre système et constitue le prochain objectif de ces travaux.

<p>Nomenclature</p> <p>Indices : <i>i</i> : tâche <i>j</i> : unité <i>n</i> : évènement représentant le début d'une tâche <i>s</i> : état</p> <p>Ensembles : <i>I</i> : ensemble des tâches <i>J</i> : ensemble des unités <i>suit</i> : matrice telle que $suit_{ij} = 1$ si la tâche <i>i</i> peut être réalisée sur l'unité <i>j</i> <i>N</i> : ensemble des évènements par unité <i>S</i> : ensemble des états</p>	<p>Paramètres : $B_{ij}^{min}, B_{ij}^{max}$: quantité minimale et maximale d'un batch <i>i</i> sur une unité <i>j</i> ST_s^{max} : quantité maximale de stockage d'un état <i>s</i> $\rho_{s,i}$: proportion de <i>s</i> consommée ($\rho_{s,i} < 0$) ou produite ($\rho_{s,i} > 0$) par <i>i</i> $\alpha_{i,j}$: partie fixe de la durée d'une tâche <i>i</i> sur une unité <i>j</i> $\beta_{i,j}$: partie variable de la durée d'une tâche <i>i</i> sur une unité <i>j</i> <i>H</i> : horizon de temps <i>Demand_s</i> : demande en <i>s</i> qui doit être satisfaite en fin de campagne</p> <p>Variables : $w(i,j,n)$: variable binaire qui associe le début de <i>i</i> sur <i>j</i> à l'évènement <i>n</i> $ts(i,j,n)$: date début de <i>i</i> sur <i>j</i> à l'évènement <i>n</i> $b(i,j,n)$: taille du lot de <i>i</i> sur <i>j</i> à l'évènement <i>n</i> $ST(s,n)$: quantité de <i>s</i> à l'évènement <i>n</i> <i>MS</i> : makespan</p>
---	---

Tableau 3. Notations utilisées

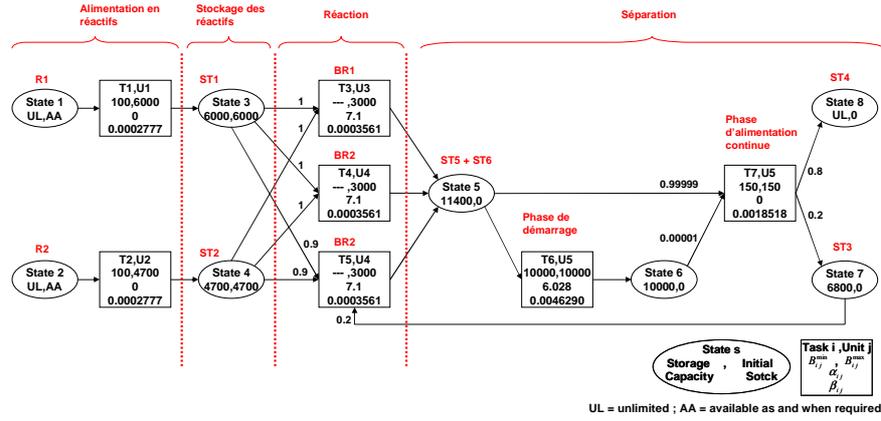


Figure 21. STN (State-Task Network) du procédé considéré

$$\text{Min } MS \quad (A.1)$$

$$ts(i, j, N) + \alpha_{i,j} w(i, j, N) + \beta_{i,j} b(i, j, N) \leq MS \quad \forall i \in I, j \in J \mid suit_{ij} = 1 \quad (A.2)$$

$$ST(s, N) + \sum_{i \in I \mid \rho_{s,i} > 0} \rho_{s,i} \sum_{j \in J \mid suit_{ij} = 1} b(i, j, N) \geq Demand_s \quad \forall s \in S \quad (A.3)$$

$$\sum_{i \in I \mid suit_{ij} = 1} w(i, j, n) \leq 1 \quad \forall j \in J, \forall n \in N \quad (A.4)$$

$$w(i, j, n) B_{ij}^{min} \leq b(i, j, n) \leq w(i, j, n) B_{ij}^{max} \quad \forall i \in I, j \in J \mid suit_{ij} = 1, \forall n \in N \quad (A.5)$$

$$ST(s,n) = ST(s,n-1) + \sum_{i \in I | \rho_{si} > 0} \rho_{si} \sum_{j \in J | \text{suit}_{ij} = 1} b(i,j,n-1) + \sum_{i \in I | \rho_{si} < 0} \rho_{si} \sum_{j \in J | \text{suit}_{ij} = 1} b(i,j,n) \quad \forall s \in S, n \in N \quad (\text{A.6})$$

$$ts(i,j,n+1) \geq ts(i',j,n) + \alpha_{i',j} w(i',j,n) + \beta_{i',j} b(i',j,n) \quad \forall i, i', j | \text{suit}_{ij} = 1, \text{suit}_{i'j} = 1, \forall n < N \quad (\text{A.7})$$

$$ts(i,j,n+1) \geq ts(i',j',n) + \alpha_{i',j'} w(i',j',n) + \beta_{i',j'} b(i',j',n) - H(1 - w(i',j',n)) \quad \forall s, i, i', j, j' | \text{suit}_{ij} = 1, \text{suit}_{i'j'} = 1, i \neq i', j \neq j', \rho_{si} < 0, \rho_{s'i'} > 0, \forall n \in N | n < N \quad (\text{A.8})$$

$$ST(s,n) \leq ST_s^{\max} \quad \forall s \in FIS, \forall n \in N \quad (\text{A.9})$$

$$w(i,j,n) = b(i,j,n) = ts(i,j,n) = 0 \quad \forall i, j | \text{suit}_{ij} = 0, \forall n \in N \quad (\text{A.10})$$

6.2. Vers la supervision des procédés

Pour des raisons de sécurité et de performance, les fonctions de surveillance et de supervision ont un rôle majeur en conduite de procédé. Cependant, certains systèmes industriels rendent complexes ces fonctions en raison du nombre croissant de variables process mises en jeu et de la complexité de leurs relations. Le comportement du procédé est alors difficilement interprétable. Pour cette raison, les travaux de recherche en cours visent à concevoir un système de supervision couplé au simulateur dynamique hybride *PrODHyS*. L'architecture de ce système est montrée sur la figure 22.

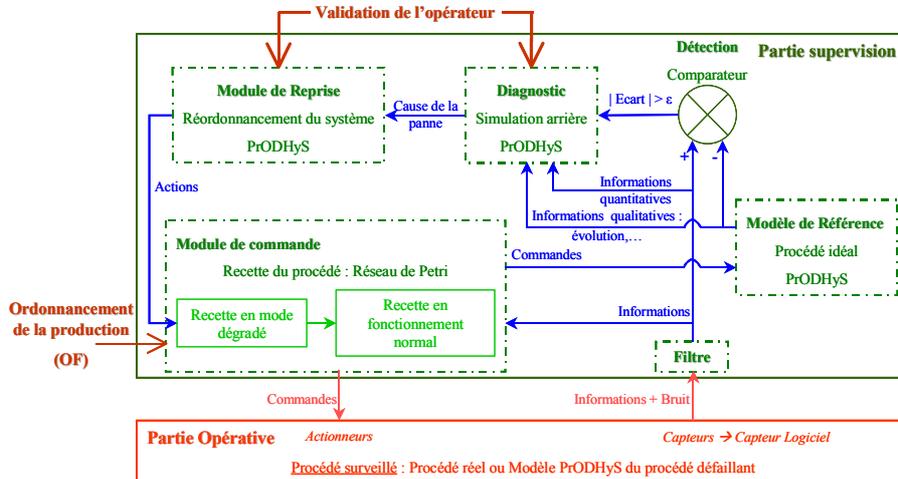


Figure 22. Architecture du système de supervision

Pour cela, un modèle de simulation du procédé réel est construit avec *PrODHyS* afin de constituer le *modèle de référence* du superviseur. Celui-ci est simulé en temps réel et de manière synchrone avec le procédé. Une sorte d'*observateur du*

ystème physique est ainsi établi et permet d'obtenir une estimation de l'état complet du système à tout instant.

Le module de surveillance doit être capable de traiter aussi bien les défauts du système physique (fuites, pertes thermiques, etc.) que ceux des appareils de contrôle/commande (actionneurs, capteurs, etc.). Comme définie dans (De Kleer *et al.* 1987), notre approche est orientée « bon fonctionnement ». Elle repose donc sur la comparaison du comportement attendu, obtenu à l'aide de la simulation du modèle de référence (valeurs des variables d'état) et le comportement réel observé (mesures provenant du procédé). Par ailleurs, une détection par seuil fixe est utilisée. Cette méthode consiste à détecter l'instant pour lequel les grandeurs surveillées sortent d'une zone délimitée par des seuils autour d'une valeur objectif (valeur de la grandeur au sein du modèle de référence). Bien que des méthodes telles que la détection par seuil variable, soient plus rigoureuses et permettent la minimisation du nombre de détections erronées, (Duvilla, 2005) indique que la détection par seuil fixe reste plus robuste aux perturbations. Par ailleurs, elle permet d'obtenir une décision assez rapidement et est facilement implémentable et paramétrable.

Pour assurer un fonctionnement cohérent de cette fonction, il faut d'abord que les mesures soient filtrées afin d'éliminer le bruit. Toutefois, il peut subsister une *incertitude spatiale* et l'écart ε entre mesures et variables du modèle au-delà duquel on considère qu'un défaut est avéré, reste un point délicat à évaluer. Généralement, cette valeur ε est un compromis obtenu à partir d'une série de simulations dans lesquelles sa valeur est ajustée. D'autre part, la simulation du modèle de référence doit évoluer en temps réel et de manière synchrone avec le procédé surveillé. En général, cette caractéristique ne peut pas être garantie à tout instant. En effet, une *incertitude temporelle* existe du fait des paramètres intrinsèques des modèles de simulation décrivant la dynamique du système. Ainsi, certains événements temporels ou d'état peuvent apparaître au niveau du modèle de référence simulé soit en avance, soit en retard par rapport aux observations faites sur le procédé surveillé. Ici encore, se pose le problème de déterminer à partir de quel écart ε peut-on considérer qu'une défaillance est potentiellement détectée. Dans tous les cas, une procédure de recalage du modèle de référence sur le procédé réel doit être réalisée pour pouvoir valider un test de détection.

Lorsque l'occurrence d'un défaut est avérée, la phase de diagnostic est lancée. Ce module s'appuie sur un outil intégrant des connaissances qualitatives du procédé dans lequel sont répertoriées des règles du type « *SI condition ALORS cause* » (méthodes de classification (Dubuisson, 2000)). Toutefois, pour affiner la discrimination et l'interprétation des défaillances constatées, il paraît intéressant d'exploiter les informations fournies par la simulation du modèle de référence, notamment la valeur des variables d'état ainsi que leurs dérivées. En effet, si les mesures provenant du procédé sont pertinentes pour la commande, elles sont souvent insuffisantes pour le diagnostic. Des informations complémentaires concernant ces travaux peuvent être trouvés dans (Olivier *et al.*, 2006a), (Olivier *et al.*, 2006b).

7. Conclusion

La mise en œuvre d'un formalisme à haut niveau d'abstraction associé à des méthodes d'intégration performantes a abouti à la construction d'un simulateur dynamique hybride numériquement robuste. Dans cet article, les potentialités de *PrODHyS* ont été illustrées à travers la modélisation et la simulation d'un procédé semi-continu. Un nouveau module dédié à la modélisation des recettes basée sur la norme *ISA/SP88* a aussi été introduit. Les travaux à venir viseront d'une part, à développer l'approche couplée optimisation/simulation pour la résolution de problème d'ordonnancement de ce type de procédé et d'autre part, de poursuivre les travaux visant à intégrer la simulation au sein d'un système de supervision.

8. Bibliographie

Andersson M., **1994**, Object-Oriented Modelling and Simulation of Hybrid Systems, *Ph.D thesis, Lund Institute of Technology, Sweden.*

Baudet P., Azzaro-Pantel C., Domenech S. et Pibouleau L., **1995**, *A discrete-event simulation approach for scheduling of batch processes*, *Computers and Chemical Engineering*, *19*, S633-S638

Barton P.I., Pantelides C.C., **1994**, *The Modelling of Combined Discrete/Continuous Processes*, *AIChE Journal*, *40*:966-979,

Burkard R.E. et Hatzl J., **2005**, Review, extensions and computational comparison of MILP formulations for scheduling of batch processes, *Computers and Chemical Engineering*, *29*, 1752-1769

Champagnat, R., Esteban, P., Pingaud, H., Valette, R., **1998**, Modelling and simulation of hybrid system through Pr-Tr PN-DAE model., *ADPM'98, Reims*, pp131-137,

Demongodin I., **2001**, Generalised Batches Petri Net: Hybrid Model for High Speed Systems with Variable Delays, *Discrete Event Dynamic Systems: Theory and Applications*, *Vol.11*, n°1/2, p.137-162,

De Kleer J., Williams B. C., **1987**, Diagnosing multiple faults, *Artificial Intelligence*, *Vol. 32*, pp. 97-130,

Duviella E., **2005**, Conduite réactive des systèmes dynamiques étendus à retards variables. Cas des réseaux hydrographiques, *Thèse de Doctorat, ENI Tarbes, France*

Dubuisson B., **2000**. *Diagnostic, intelligence artificielle et reconnaissance de formes*, Hermès, Paris.

Floudas C.A. et Lin X., **2004**, Continuous-time versus discrete-time approaches for scheduling of chemical processes : a review, *Computers and Chemical Engineering*, 28, 2109-2129

Gear C.W., **1971**, *The simultaneous numerical solution of differential-algebraic equations*, IEEE, Transactions on circuit Theory, CT.18 (1), 89-95.

Hétreux G., Théry R., Perret J., LeLann J.M., Joulia X., **2002**, Bibliothèque orientée-objet pour la simulation dynamique des procédés : architecture et mise en oeuvre, *Congrès SIMO, Toulouse*.

Hétreux G., Perret J., LeLann J.M., **2003**, Object hybrid formalism for modelling and simulation of chemical processes, *ADHS'03, Saint-Malo, France*

Hétreux G., Perret J., LeLann J.M., **2004**, Component based approach for simulation of dynamic hybrid systems, *Conference on Conceptual Modeling and Simulation (CMS '04), Genoa, Italy*

Hétreux G., Zaraté P., LeLann J.M., **2005**, Conception d'un atelier de modélisation pour la simulation dynamique hybride des procédés, *Congrès Français de Génie des Procédés (CFGP '05), Toulouse, France*

Hétreux G., Fabre F., Théry R., LeLann J.M., **2007**, Approche combinée Optimisation/Simulation pour l'ordonnement des procédés discontinus, *Congrès Français de Génie des Procédés (CFGP '07), Saint-Etienne, France*

Ierapetritou M.G., Floudas C.A., **1998**, *Effective continuous-time formulation for short-term scheduling: 1. Multipurpose batch processes*, Industrial and Engineering Chemistry Research, 37, 4341

Jourda, L., Joulia, X., Koehret, B., **1996**, *Introducing ATOM, the Applied Thermodynamic Object-oriented Model*, Computer & Chemical Engineering, 20A, S157-S164.

Méndez C.A. et Cerdá J., **2003**, Dynamic scheduling in multiproduct batch plants, *Computers and Chemical Engineering*, 27, 1247-1259

Mendez C.A., Cerdá J, Grossmann I.E. Harjunkoski I, Fahl M. , **2006**, State of the art review of optimisation methods for short term scheduling of batch processes, *Computer and Chemical Engineering* , Vol. 30 pp. 913-946

Moyse, A., **2000**, Odysseo : plate-forme orientée-objet pour la simulation dynamique des procédés, Thèse de Doctorat, INP de Toulouse, France

Nilsson B, **1993**, Dynamic modeling of chemical processes using Omola , *IchEmE Symposium Series*, Vol. 133, pp. 103-110

Olivier N., Hétreux G., LeLann J.M., **2006a**, Use of an object oriented dynamic hybrid simulator for the monitoring of industrial processes, *IFAC Conference ADHS'2006, Alghero, Sardaigne, Italie*

Olivier N., Hétreux G., LeLann J.M., LeLann M.V., **2006b**, Vers une approche de diagnostic des systèmes dynamiques hybrides, *SIMO'2006, Toulouse, France*

Perret J., Hétreux G., LeLann J.M., **2004.**, Integration of an object formalism within a hybrid dynamic simulation environment, *Control Engineering Practice (Elsevier), Vol. 12/10, pp. 1211-1223*

Perret J., **2003**, Intégration des Réseaux de Petri Différentiels à Objets dans une plateforme de simulation dynamique hybride : application aux procédés industriels, *Thèse de Doctorat, INP, Toulouse (France).*,

Piela P.C., **1989**, ASCEND : an object-oriented computer environment for modelling and analysis, *PhD Thesis, Carnegie-Mellon University, USA*

Roslöf J., Harjunkoski I, björkqvist J., Karlsson S. and Westerlund T, **2001**, An MILP-based reordering algorithm for complex industrial scheduling and rescheduling, *Computers and Chemical Engineering, 25, 821-828*

Sargousse A, **1999**, Noyau numérique orientée-objet dédié à la simulation des systèmes dynamiques hybrides. *Thèse de Doctorat, INPT, France*

Shaik M.A., Janak S.L. et Floudas C.A., **2005**, *Continuous-Time Models for Short-Term Scheduling in Multipurpose Batch Plants : a Comparative Study*, Chemical Engineering, Princeton University, Princeton, NJ 08540

Shah N., Liberis L., Izumoto E., Henson R., **1996**, Integrated batch plant design: a polymer plant case study, *Computer and Chemical Engineering, Vol. 20, Suppl., pp. S1233-S1238*,

Stephanopoulos G., Henning G., Leone H., **1990**, MODEL.LA : a modelling language for process engineering. I- the formal framework, *Computers and Chemical Engineering, 14, 8, pp. 813-846*,

Théry R., Hétreux G., Meyer X.M., Joulia X., **2002.**, Environnement orienté objet pour la simulation dynamique d'un procédé de distillation réactive, *Congrès SIMO, 24-25 Octobre, Toulouse*

Zaytoon J., **2001**, *Systèmes dynamiques hybrides*. HERMES Sciences publications,