

Formalisme hybride orienté objet pour la simulation dynamique des procédés chimiques

J.Perret, G. Hétreux, J.M. Le Lann

Laboratoire de Génie Chimique (LGC - UMR 5503), département PSE/Génie Industriel,
5 Rue Paulin Talabot, 31106 Toulouse Cedex 1, France

E-mail: Jocelyne.Perret@ensiacet.fr, Gilles.Hetreux@ensiacet.fr,
JeanMarc.LeLann@ensiacet.fr.

MOTS CLES

simulation dynamique hybride, formalisme *RdPDO*, approche orientée objet, procédés

L'ESSENTIEL

- *PrODHyS* est un environnement orienté objet pour la simulation dynamique hybride des procédés chimiques.
- Il fournit des composants généraux et réutilisables permettant la construction d'éléments plus spécifiques.
- Le formalisme développé pour modéliser les systèmes hybrides s'appuie sur le modèle « Réseau de Petri Différentiel à Objets » (*RdPDO*).

SYNOPSIS

- *PrODHyS* is an object oriented environment for the dynamic simulation of chemical engineering processes.
- It provides standardised and reusable components designed for the built of more specific elements.
- The formalism developed in order to model hybrid systems is based on the model « Object Differential Petri Net » (*ODPN*).

L'ACCROCHE

PrODHyS est un environnement de simulation dynamique hybride dont le formalisme et la structuration logicielle repose sur les techniques les plus récentes en matière de Génie Logiciel (UML, C++, etc.) ainsi que sur la philosophie d'interfaces coopératives Orientées-Objet (notion de composants O-O).

INTRODUCTION

Aujourd'hui, la simulation dynamique constitue un outil d'études privilégié lors des phases d'analyse et de conception des systèmes industriels. L'apparition de logiciels commerciaux tels que *ProSim Batch* (www.prosim.net) ou *Aspen Dynamics* (www.aspentech.com) en est la preuve. Parallèlement, ces simulateurs ont largement profité de l'évolution rapide de la puissance des calculateurs ainsi que d'avancées considérables dans les stratégies de résolution notamment de grands systèmes d'équations algébro-différentielles. Dans ce contexte, les travaux de recherche entrepris depuis plus de quinze ans au sein du *Laboratoire de Génie Chimique (LGC)* ont conduit au développement progressif d'une plate-forme de simulation nommée *PrODHyS (Process Object Dynamic Hybrid Simulator)*.

Par ailleurs, l'évolution des procédés de fabrication vers des processus batch confronte les modélisateurs avec des modes de fonctionnement souvent difficiles à gérer avec des modèles purement continus ou purement discrets. Pour ces raisons, les systèmes dynamiques hybrides constituent désormais un intérêt certain (Zaytoon, 2001) et de nombreux outils de simulation ont été développés pour les représenter. Nous pouvons citer par exemple *gPROMS* (Barton et Pantelides, 1994), *shift* (Deshpande et al., 1998), *Omsim* (Andersson, 1994), *Chi* (Fábián et al., 1998),

BaSiP (Wöllhaf et al., 1996), *ABACUSS II* (www.yoric.mit.edu/abacuss2/abacuss2.html). Parallèlement, de nombreux modèles décrivant simultanément les aspects discrets et continus ainsi que leurs interactions ont été définis ou étendus. Nous pouvons citer par exemple les *automates hybrides* (Alur et al., 1995), les *réseaux de Petri prédicats-transitions différentiels* (Champagnat et al., 1998), les *réseaux de Petri hybrides* (Le Bail et al., 1991), les *statecharts hybrides* (Kesten et Pnuéli, 1992). De la même manière, *PrODHyS* a adopté une modélisation hybride des procédés en tenant compte simultanément des aspects continus, décrits par des équations différentielles algébriques (EDA) et des aspects discrets, décrits par des réseaux de Petri. *PrODHyS* se caractérise aussi par une approche orientée objet des procédés, approche qui s'impose actuellement, tant sur le plan conception que développement, comme une réponse adaptée aux besoins d'extensibilité, de réutilisabilité et de qualité logicielle. Ainsi, le noyau de simulation hybride a été conçu autour du formalisme *Réseau de Petri Différentiel à Objets (RdPDO)* qui exploite la complémentarité entre l'approche objet et le formalisme hybride.

Le but de cet article est de présenter le formalisme employé pour décrire et gérer la dynamique hybride des procédés. Une présentation préliminaire de la plate-forme est dans un premier temps réalisée.

1. L'ENVIRONNEMENT DE SIMULATION

1.1. Structure générale de la plate-forme

Le formalisme développé au sein de l'environnement *PrODHyS* ainsi que la structuration logiciel repose sur les techniques les plus récentes en matière de Génie Logiciel (UML, C++, etc.) ainsi que sur la philosophie d'interfaces coopératives Orientées-Objet (notion de composants O-O) inspirée des recommandations du projet européen CAPE-OPEN et Global Cape Open. La technologie objet permet une modélisation modulaire et hiérarchique des procédés. Ainsi, la dérivation des éléments généraux fournis par *PrODHyS* facilite la construction d'entités spécifiques plus complexes grâce à des mécanismes tels que la composition, l'héritage et le polymorphisme.

Signalons que cette plate-forme résulte de nombreux travaux réalisés depuis plusieurs années au sein du laboratoire (Moysse, 2000), *DISCo* (Sargousse, 1999), (Le Lann, 1999), (Jourda *et al.*, 1996). Actuellement, *PrODHyS* est constitué de plus de mille classes structurées autour de cinq modules principaux. Le schéma de la figure 1 représente la structure globale de la plate-forme.

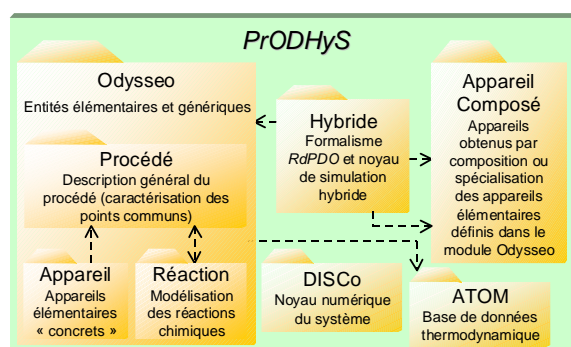


Fig. 1. Structure de la plate-forme *PrODHyS*

Soulignons simplement que le module *Hybride* contient l'ensemble des classes nécessaires à la description du formalisme et du noyau de simulation hybride.

1.2. Formalisme hybride considéré

Le formalisme *Réseau de Petri à Objets* repose sur une approche mixte combinant des modèles continus et discrets dans une même structure, l'aspect hybride étant pris en compte dans l'interface entre les deux parties. Ce modèle se caractérise essentiellement par deux aspects :

1.2.1. Les aspects dynamiques et comportementaux

Un *RdPDO* est caractérisé par un ensemble de places dont certaines sont dites *discrètes* et d'autres *différentielles*. Une place discrète représente un état discret (la disponibilité d'une ressource par exemple) alors qu'une place différentielle désigne un comportement continu caractérisé par un ensemble

de variables d'état continues et dont la dynamique est régie par un système algèbro-différentiel. Un jeton mis dans une place déclenche la résolution du système algèbro-différentiel correspondant et provoque ainsi l'évolution continue au cours du temps des variables d'état. Ceci constitue la partie *dynamique* du modèle. En ce qui concerne la partie *comportementale*, le solveur surveille parallèlement les fonctions événements de chaque transition située en aval de la place marquée. Quand un événement est satisfait, la transition associée est franchie. Un nouveau marquage est mis en place et active le système d'équations correspondant pour lequel, les valeurs initiales des variables et de leurs dérivées ont été déterminées par les fonctions de jonction de la (ou des) transition(s) située(s) en amont, ainsi que par une procédure automatique intégrée au solveur numérique permettant le calcul de conditions initiales cohérentes.

1.2.2. Les aspects statiques et structurels

L'intégration des concepts objets (Sibertin-Blanc, 1985) permet de décrire les aspects statiques et structurels du système. Chaque jeton est désormais défini comme une classe d'objet possédant un ensemble d'attributs, dont les variables d'états continues, et un ensemble de méthodes qui traitent ces données. Ainsi, les conditions associées aux transitions utilisent d'une part les méthodes et attributs portés par les jetons sensibilisant la transition et d'autre part les méthodes et attributs de l'objet qui intègre ce réseau de Petri. Les actions, quant à elles, exécutent les méthodes offertes par les jetons franchissant la transition ou appartenant à l'objet incluant ce réseau ; elles peuvent également créer et détruire des jetons. Le rôle de ces actions ne consiste pas simplement à déterminer les valeurs initiales des variables d'état continues et de leur dérivées, mais également à modifier l'état de l'objet référencé par le jeton ou l'état de l'objet qui intègre ce réseau. Enfin, les arcs d'entrée et de sortie sont explicitement typés. Ils portent une inscription spécifiant le type d'objet autorisé à transiter sur l'arc. Ce mécanisme, avec individualisation et typage du jeton, rend le réseau plus compact, sans perte d'information.

L'exploitation de l'approche objet au sein des *RdPDO* s'effectue sous deux formes. La première est d'*introduire les objets dans les réseaux de Petri*. La philosophie sous-jacente est de modéliser un sous-système par un réseau de Petri unique qui manipule des jetons individualisés et enrichis d'entités structurées porteuses d'information. Le réseau modélise ainsi la structure de contrôle du système alors que les jetons représentent la structure de données associée. La seconde forme est d'*introduire les réseaux de Petri dans les objets* afin d'en décrire le comportement interne. Le marquage du réseau indique l'état courant de l'objet, le franchissement d'une transition se traduit par l'exécution d'une de

ses méthodes et la structure globale du réseau spécifie les séquences légales d'exécution des méthodes.

Soulignons enfin que le formalisme *RdPDO* constitue une extension plus aboutie du formalisme *RdP prédicats-transitions différentiel* défini par Champagnat (Champagnat et al., 1998).

1.2.3. Représentation

La représentation adoptée pour décrire le modèle *RdPDO* est illustrée sur la figure 2.

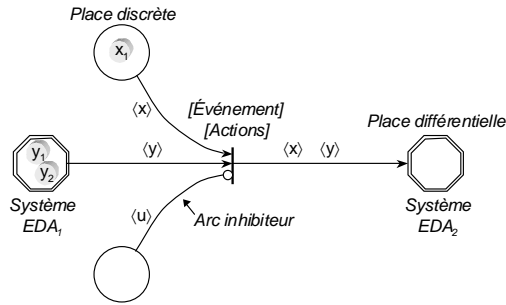


Fig. 2. Représentation des *RdPDO*

1.3. Le noyau de simulation

Afin d'implémenter le formalisme *RdPDO*, le noyau du simulateur se décompose en trois modules : le *solveur discret*, le *solveur continu* et un *gestionnaire de simulation* qui gère les interactions entre les deux solveurs. Le cycle de fonctionnement du simulateur est alors le suivant : **1.** le gestionnaire de simulation donne la main au solveur discret **2.** le modèle discret est exécuté jusqu'à ce qu'une configuration stable soit atteinte, c'est à dire jusqu'à ce que plus aucune transition ne puisse être franchie **3.** le gestionnaire de simulation concatène alors les systèmes EDA de chaque place différentielle marquée ainsi que les fonctions de sensibilisation de chaque transition située en aval des places marquées **4.** l'intégration du système algèbro-différentiel global résultant est lancée au niveau du solveur continu, provoquant ainsi l'évolution des variables continues au cours du temps ; celle-ci est stoppée dès l'occurrence d'un événement **5.** le gestionnaire de simulation enregistre les variables et redonne la main au solveur discret qui exécute les actions associées aux transitions franchissables et établit le nouveau marquage (\Rightarrow retour en **1**).

2. MODELISATION OBJET D'UN PROCÉDE

La problématique de la simulation hybride réside principalement dans la gestion des séquences légales entre les différentes configurations possibles du système EDA en fonction de l'état effectif du système. Lorsque le procédé devient complexe, le nombre d'états atteignables peut rapidement s'accroître. La structuration objet de *PrODHyS* permet de pallier cette difficulté. En effet, (Hétreux

et al., 2002) ont montré qu'il était possible de construire la plupart des appareils complexes par composition ou spécialisation d'un petit nombre d'appareils élémentaires. Toute la difficulté réside dans la caractérisation de ces appareils élémentaires qui ne sont pas nécessairement des appareils élémentaires mais plutôt des entités autonomes abstraites et génériques ayant une fonctionnalité simple et prédéfinie et communiquant via une interface isomorphe. Les sections suivantes spécifient les axes de modélisation d'un appareil. Un exemple d'application est dans un premier temps proposé afin d'illustrer les concepts mis en œuvre.

2.1. Exemple illustratif

L'exemple proposé est basé sur l'évaporation de Rayleigh. Le procédé utilisé est constitué d'une cuve ouverte comportant un détecteur de niveau et un système thermique permettant de chauffer le mélange initialement liquide. Selon les conditions opératoires, une phase vapeur de composition y peut apparaître et s'échapper dans l'atmosphère avec un débit v . Le modèle mathématique de ce système à l'équilibre thermodynamique et dans son état maximal (c'est à dire, liquide/vapeur) est représenté sur la figure 3.

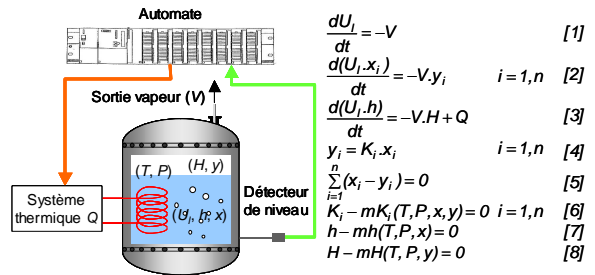


Fig. 3. Exemple illustratif : évaporation de Rayleigh

Les trois premières équations (1), (2) et (3) sont respectivement le bilan matière global, les bilans matière partiels et le bilan d'énergie. Les équations (4) et (5) sont les relations traduisant l'équilibre liquide/vapeur. Enfin, les équations (6), (7) et (8) sont les modèles respectifs des constantes d'équilibre liquide/vapeur K_i , de l'enthalpie liquide h et de l'enthalpie vapeur H au sein de la cuve.

2.2. Modélisation de la matière

L'utilisation d'un modèle monolithique pour la représentation conjointe du comportement de la cuve et du comportement de la matière dans la cuve peut rapidement conduire à un grand nombre de places et transitions. Or, dans *PrODHyS*, la matière est faiblement couplée avec l'appareil qui la contient. Ainsi, afin de diminuer la complexité du modèle hybride, celui-ci est décomposé. Plutôt que de confondre les changements d'états de la matière avec ceux de la cuve, ils sont décrits séparément par des réseaux de Petri spécifiques. Le comportement de la matière reste cependant intégré à celui de la cuve grâce à l'utilisation d'un jeton Matière (de type $\langle m \rangle$)

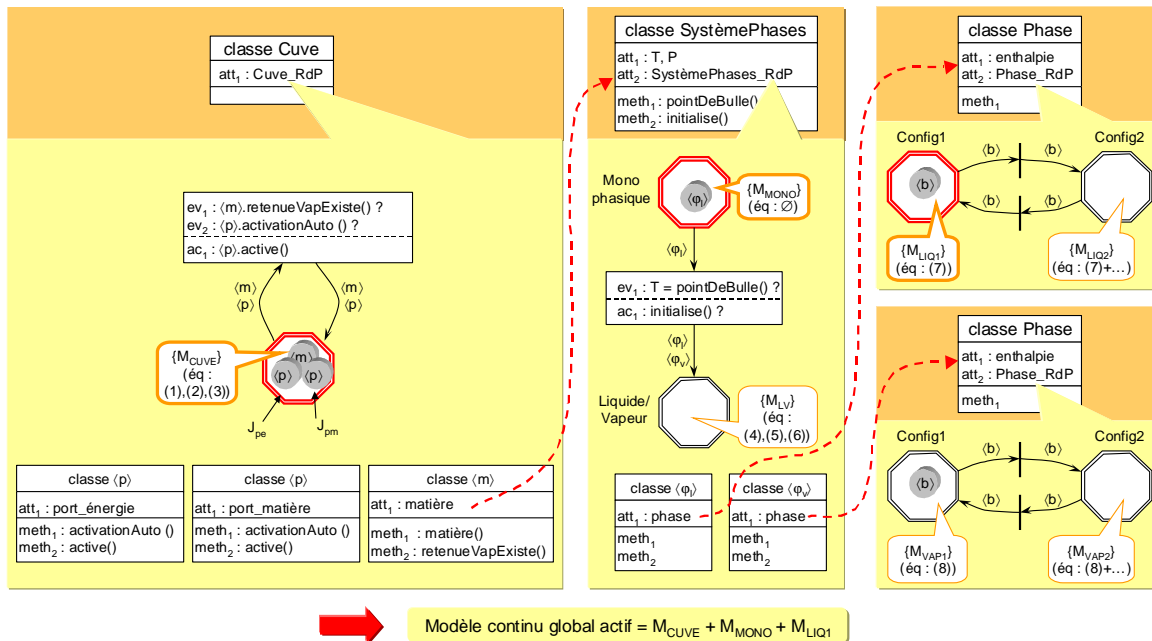


Fig. 4. Modèle associé à l'exemple

portant un objet Matière et se déplaçant sur le réseau de la cuve. De la même façon, toujours dans un souci d'accroître la modularité et par conséquent la réutilisabilité du modèle, le comportement de la matière peut être décomposé. Une partie se rapporte explicitement au système de phases global ; une autre est spécifique aux phases elles-mêmes. Le comportement de la phase reste cependant intégré à celui du système de phases grâce à l'utilisation d'un autre jeton typé qui se déplace sur le réseau de Petri du système de phases ; il s'agit du jeton Phase qui porte un objet Phase. Selon la nature de la phase transportée, le jeton Phase peut être spécialisé en un jeton PhaseLiquide (de type $\langle \varphi_l \rangle$) et un jeton PhaseVapeur (de type $\langle \varphi_v \rangle$). Le modèle global associé à l'exemple de l'évaporation de Rayleigh est alors décrit sur la figure 4. Celle-ci propose des fragments de modèles de chaque sous-système. Les deux réseaux de Petri de droite modélisent le comportement des phases susceptibles d'être présentes dans la cuve. Deux configurations distinctes (que l'on ne détaillera pas) les caractérisent. De façon générale, les équations associées aux places différentielles concernent les modèles permettant de calculer les propriétés physiques de la phase (enthalpie par exemple) ou les relations éventuelles permettant de déterminer la rétention. Le réseau de Petri du milieu modélise le comportement du système de phases et permet de représenter le passage de la matière de l'état monophasique à l'état liquide/vapeur lorsque la température de bulle est atteinte. Le marquage de la place monophasique spécifie la nature de la phase présente dans la cuve (jeton PhaseLiquide dans l'exemple). Le système d'équations associé à l'état monophasique est vide ; celui associé à l'état Liquide/Vapeur comporte les équations d'équilibre entre phases ((4) et (5)) et les modèles des constantes d'équilibre (6). A gauche, le réseau de Petri de la

cuve modélise l'activation automatique du port matière de sortie lorsque la phase vapeur apparaît. Ainsi, le modèle continu global du système résulte de la concaténation du système EDA M_{CUVE} associé à la place différentielle marquée du réseau de Petri de la cuve, du système EDA M_{MONO} associé à la place différentielle marquée du réseau de Petri du système de phases (car le jeton Matière marque la place différentielle du RdP de la cuve) et enfin du système EDA M_{LIQ1} associé à la place différentielle marquée du RdP de la phase liquide (seul le jeton PhaseLiquide marque la place du RdP du système de phases).

2.3. Modélisation de l'appareil élémentaire

Dans *ProDHyS*, un appareil élémentaire est défini selon trois axes principaux : sa *topologie*, son *modèle* et sa *configuration*.

2.3.1. Topologie de l'appareil

De manière générale, un appareil est défini comme une enceinte qui peut échanger de la matière, de l'énergie ou de l'information. Afin de formaliser ces échanges, un élément d'interface nommé *port* a été introduit (ceci rejoint la notion d'encapsulation et d'accesses de l'approche objet).

L'établissement d'un échange entre la partie interne de l'appareil et l'extérieur (environnement ou autre appareil) correspond dans *ProDHyS* à un partage de données (cf. figure 5). Toute donnée associée à un *port de sortie* est rendue potentiellement accessible. Pour accéder à cette donnée, il faut qu'un *port d'entrée* lui soit connecté. Ainsi, un port de sortie contient la donnée échangée, un port d'entrée ne contient qu'une référence sur cette donnée. L'intérêt d'un tel mécanisme est d'éviter l'écriture d'équations de « connexion » du type $F_{tout} = F_{2in}$.

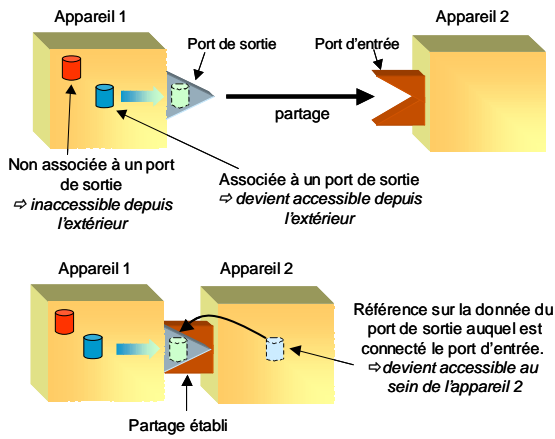


Fig. 5. Représentation des ports

Compte tenu de la nature des échanges, deux types de port existent : les *ports de communication* (échange d'information) et les *ports de transport* (échange physique de matière ou d'énergie). Dans ce dernier cas, le transfert est caractérisé par un *flux* et un *potentiel*.

2.3.2. Modélisation de l'appareil

De manière générale, la mise en place du modèle selon le formalisme *RdPDO* s'effectue en deux étapes : la première consiste à définir l'ensemble des états possibles du procédé ainsi que les transitions d'état et les événements intrinsèques le faisant évoluer ; la seconde consiste à établir les modèles mathématiques (un par place différentielle) qui décrivent le comportement continu de l'appareil. Dans le cas particulier de l'appareil élémentaire, seules les équations bilans (1), (2) et (3) sont prises en compte. Dans celles-ci, les variables liées aux flux (donc fournies par les ports de transport) sont les seules variables propres à l'appareil. Or, par construction, un appareil peut posséder un nombre varié de ports de types différents. De plus, la valeur de ces flux dépend de l'état courant de l'appareil. Par exemple, la simulation du débordement d'un bac nécessite un port de sortie liquide mais dont le flux n'existe que si le niveau maximum est atteint. De même, une variable de débit vapeur n'a d'intérêt que lorsque la matière contenue dans l'appareil est diphasique. Ces types de situation peuvent se gérer de deux manières : soit écrire un système EDA en ajoutant une équation du type $F = 0$, où F est le débit non pris en compte ; soit enlever cette variable du système EDA global. Dans *ProDHYS*, cette dernière solution est adoptée. Pour la mettre en œuvre, les ports sont associés à des jetons typés nommés jetons Port (de type $\langle p \rangle$) et la prise en compte du flux correspondant est gérée au niveau d'un réseau de Petri spécifique au port. Le réseau de Petri d'un appareil possède alors autant de jetons Port que l'appareil possède de ports. La figure 6 représente la modélisation adoptée pour représenter le système précédent et notamment ses entrées/sorties. La cuve est caractérisée par deux ports : un port d'énergie

pour la chauffe de la cuve et un port de matière pour la sortie vapeur. Les jetons Port associés (J_{pe} et J_{pm}) marquent la place différentielle du RdP de la cuve. Quant au marquage courant de leur propre réseau, il indique d'une part que la chauffe est en fonctionnement (place marquée du RdP du port d'énergie : Actif) et d'autre part, que la matière est en phase liquide ; en effet, il n'existe pas encore de dégagement vapeur (place marquée du RdP du port de matière : Inactif).

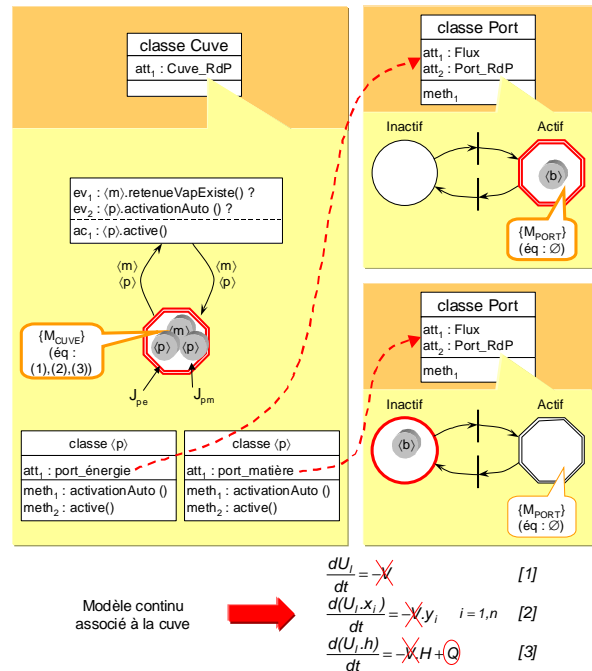


Fig. 6. Modélisation des entrées/sorties

La variable de flux associée à un port n'est prise en compte dans la génération du système EDA global que si la place marquée du port est la place différentielle. Signalons par ailleurs qu'un des intérêts d'utiliser les réseaux de Petri pour modéliser le comportement des ports est de pouvoir, en plus, associer des équations spécifiques à la place différentielle (pour le calcul de la pression hydraulique lorsque les phénomènes hydrauliques sont pris en compte par exemple).

2.3.3. Configuration de l'appareil

L'application d'une configuration finalise la construction d'un appareil. Par exemple, elle permet de spécifier la valeur des paramètres constitutifs et géométriques (diamètre d'une cuve, volume, débit maximum, etc.), le choix de certains modèles (modèle de rétention, de pression, etc.), le marquage initial de l'appareil, etc.

3. STRUCTURATION OBJET DU MODELE

Le paragraphe 1 a souligné que la plate-forme *ProDHYS* avait pour objectif de fournir des composants généraux permettant de développer des entités complexes plus spécifiques en utilisant les

mécanismes de dérivation propre à l'approche objet. Ces principes ont largement été utilisés à travers la description de la topologie et de la configuration des appareils (Moysse, 2000). Ce qui suit met en évidence l'intégration de ces concepts à travers la mise en place des modèles hybrides.

3.1. Specialisation

Dans l'environnement *PrODHyS*, la complexité du comportement d'un équipement s'inscrit à plusieurs niveaux. Au niveau le plus bas, le réseau de Petri regroupe les caractéristiques communes des différents réseaux. Au fur et à mesure que le modèle s'affine, des éléments (places, transitions, événements, etc.) lui sont ajoutés.

3.2. Composition

Un appareil composé est défini comme un appareil constitué de plusieurs appareils élémentaires ou composés. Le comportement d'un appareil composé est responsable du comportement des appareils qui le composent. Par exemple, la cuve de stockage représentée sur la figure 3 possède un système thermique. Dans *PrODHyS*, cet ensemble constitue un appareil composé qui peut être utilisé directement sans avoir à se soucier des connections qui les relient.

3.2. Encapsulation

Afin de faciliter l'utilisation des appareils déjà développés et de contrôler l'accès à leur modèle, certains éléments, en termes de places et transitions, sont rendus inaccessibles à l'utilisateur. En effet, des zones privées et publiques ont été définies et n'autorisent l'accès qu'à certaines places (notamment les places de commande). Le mécanisme employé ici repose sur le principe d'encapsulation.

CONCLUSION

La modélisation de la plupart des procédés industriels nécessite la prise en compte des phénomènes hybrides. L'utilisation d'un modèle de haut niveau associé avec des méthodes d'intégration numériques puissantes permettent de construire un simulateur dynamique hybride robuste. Conçu selon une approche orientée objet, *PrODHyS* fournit une bibliothèque de composants logiciels autonomes qui peuvent être composés ou spécialisés afin de développer des appareils spécifiques. Pour un développeur, l'exploitation de ces composants élémentaires facilite la conception et la création d'un nouvel appareil. Pour un utilisateur, l'exploitation des appareils prédéfinis offerts comme des « boîtes noires » facilite la mise en place de scénarios. Ces éléments font de *PrODHyS* un outil opérationnel et évolutif.

REFERENCE

- Alur R., Courcoubetis C., Halbwachs N., Henzinger T.A., Ho P.H., Nicollin X., A. Olivero, Sifakis J. and Yovine S. (1995). The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, Vol.138, p.3-34.
- Andersson M. (1994). Object-Oriented Modelling and Simulation of Hybrid Systems. Ph.D thesis, Lund Institute of Technology, Sweden.
- Barton P.I. and Pantelides C.C. (1994). The Modelling of Combined Discrete/Continuous Processes. *AIChE Journal*, 40:966-979.
- Champagnat R., Esteban P., Pingaud H. and Valette R. (1998). Modeling and simulation of a hybrid system through Pr-Tr PN-DAE model. In: *ADPM'98*, p.131-137. Reims.
- Deshpande A., Gollu A. and Semenzato L. (1998). The shift programming language for dynamic networks of hybrid systems. *IEEE Trans. Automatic Control special issue on Hybrid Systems*.
- Fábián G., van Beek D.A. and Rooda J.E. (1998). Integration of the Discrete and the Continuous Behaviour in the Hybrid Chi Simulator, *European Simulation Multiconference*, Manchester.
- Hétreux G., Thery R., Perret J., Le Lann J.M. and Joulia X. (2002). Bibliothèque orientée-objet pour la simulation dynamique des procédés : architecture et mise en œuvre, *Congrès SIMO'2002*, Toulouse, France.
- Jourda L., Joulia X. and Koehret B. (1996). Introducing ATOM, the Applied Thermodynamic Object-Oriented Model. In: *Computer & Chemical Engineering*, 20A, S157-S164.
- Kesten Y. and Pnueli A. (1992) Timed and hybrid statecharts and their textual representation, *Lecture Notes in Computer Science*, Vol.N°571.
- Le Bail J., Alla H. and David R. (1991). Hybrid Petri Nets, *Proceedings of the European Control Conference*, p.1472-1477, Grenoble.
- Le Lann J.M. (1999). Des mathématiques à la simulation dynamique robuste des procédés : le traitement algèbro-différentiel des équations EDA. Habilitation, INP, Toulouse, France.
- Moysse A. (2000). Odysseo, plate-forme orientée-objet pour la simulation dynamique des procédés. PhD thesis, INPT, France.
- Sargousse A. (1999). Noyau numérique orientée-objet dédié à la simulation des systèmes dynamiques hybrides. PhD thesis, INPT, France.
- Sibertin-Blanc C. (1985). High-level Petri nets with Data structure. *6th European workshop on Petri nets and applications*, Espoo (Finland).
- Wöllhaf K., Fritz M., Schulz C. and Engell S. (1996). BaSiP – Batch Process Simulation With Dynamically Reconfigured Process Dynamics. *Supplement to Computers and Chemical Engineering*, 20(972), p.1281-1286.
- Zaytoon J. (2001). *Systèmes dynamiques hybrides*. HERMES Sciences publications.