

Environnement G-DEVS/HLA pour la simulation distribuée de systèmes de production multiprocessus

Gregory Zacharewicz, Patrick Pujo, Claudia Frydman, Norbert Giambiasi

▶ **To cite this version:**

Gregory Zacharewicz, Patrick Pujo, Claudia Frydman, Norbert Giambiasi. Environnement G-DEVS/HLA pour la simulation distribuée de systèmes de production multiprocessus. Journal of Decision Systems, 2009, 18 (3), pp.375-402. 10.3166/JDS.18.375-402 . hal-00411815

HAL Id: hal-00411815
https://hal.science/hal-00411815

Submitted on 8 Jun 2016

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.

Environnement G-DEVS / HLA pour la simulation distribuée de systèmes de production multi processus

Grégory Zacharewicz[*] — Patrick Pujo[**] — Claudia Frydman[**] — Norbert Giambiasi[**]

[*] *Laboratoire IMS UMR CNRS 5218, Groupe Productique (GRAI),*
Université de Bordeaux, 351 Cours de la Libération, 33405 Talence cedex, France
gregory.zacharewicz@ims-bordeaux.fr

[**] *Laboratoire LSIS UMR CNRS 6168, Université Paul Cézanne,*
Avenue Escadrille Normandie Niémen, 13397 Marseille cedex 20, France
{patrick.pujo, claudia.frydman, norbert.giambiasi}@lsis.org

RÉSUMÉ. *L'objectif de cet article est de présenter un environnement de simulation pour systèmes de production intégrant des processus de pilotage et des processus opérationnels. A partir d'une spécification des processus opérationnels selon la norme JIS Z 8206 et des processus de pilotage sous forme de Workflows, nous générons des modèles G-DEVS simulables dans un contexte distribué. Après avoir introduit la méthodologie de transformation d'un processus en modèles G-DEVS, nous présenterons une mise en œuvre de l'ensemble de ces modèles, via une architecture HLA permettant l'interconnexion et l'interopérabilité entre les composants du modèle. Nous illustrons l'utilisation de cet environnement avec un exemple pris dans le domaine de la microélectronique et nous conclurons par l'application de cette approche à l'étude des synchronisations entre lignes de production et à la mise en œuvre de couplages simulation-réalité.*

ABSTRACT. *This paper presents a simulation environment for manufacturing systems integrating control processes and operational processes. From the operational processes specification according to the JIS Z 8206 norm and from the control processes as Workflows standards, distributed G-DEVS models are generated. At first, the transformation of a process in G-DEVS models is described. Then, the implementation of a global simulation of the set of these models via an HLA architecture allowing interconnection and interoperability between the model components is presented. An illustration of the use of this environment is given in the microelectronic field. We conclude by the application of this approach to the study of the synchronizations between production lines and to the implementation of coupling between simulation and reality.*

MOTS-CLÉS : *DEVS, G-DEVS, simulation distribuée, HLA, XML, systèmes de production multi processus, processus de pilotage, Workflow, JIS Z 8206.*

KEYWORDS: *DEVS, G-DEVS, distributed simulation, HLA, XML, multi-process manufacturing systems, control process, Workflow, JIS Z 8206.*

1. Introduction

Les systèmes de production manufacturière actuels tendent vers toujours plus de complexité. Ceci peut s'expliquer par la conjonction de différents facteurs indépendants, mais favorisant cette tendance. Premièrement, le monde industriel est soumis à une pression économique toujours plus importante, ce qui se traduit par un souci toujours fort de productivité et de rationalité. Ensuite, les produits manufacturés sont d'une part de plus en plus sophistiqués, donc compliqués à produire, et d'autre part de plus en plus fiables, pour répondre aux exigences des consommateurs. Par ailleurs, la durée de vie d'un produit sur le marché est de plus en plus courte, soit du fait de l'évolution technologique, soit du fait du besoin de renouvellement des produits proposés aux consommateurs. Parfois, et cela est souvent le cas dans l'industrie du semi conducteur, la durée d'existence d'un produit sur le marché est de l'ordre de sa durée de mise au point et d'industrialisation. Ceci nécessite une plus grande dynamique dans le fonctionnement des équipements de production, avec un pilotage qui doit intégrer différentes dimensions parfois contradictoires. Les investissements de production doivent alors être conçus en conséquence, et les entreprises, qui n'ont pas le droit à l'erreur, se dotent de systèmes d'information et de décision performants, mais aussi complexes et sensibles que leurs systèmes opérationnels de production. Aussi, pour simplifier leur fonctionnement interne et l'optimiser, beaucoup d'entreprises se recentrent sur leur métier de base, en optimisant leurs propres processus qui deviennent hyper spécialisés et en externalisant ou en sous traitant les activités autres, qu'elles soient en amont, en aval ou de soutien. Ceci se traduit souvent par un parallélisme entre production et industrialisation, avec de manière implicite par une augmentation du nombre d'acteurs en interactions dynamiques. Cela accroit la complexité globale du 'système' qui permet l'obtention du produit final.

Dans ce contexte que nous allons un peu plus détailler dans un premier temps, nous proposons ici un environnement de simulation multi formalisme qui permet l'édition des processus décisionnels et opératoires *via* des formalismes habituels de description, puis leur transposition dans le formalisme G-DEVS et la constitution d'un système de simulation distribuée.

Pour cela, nous exposerons les techniques permettant la transposition de formalismes. Nous donnerons quelques exemples concrets issus de l'industrie de la micro électronique, où les processus tant décisionnels qu'opérationnels sont relativement complexes.

Nous conclurons par une revue de quelques perspectives d'applications aux lignes de production alimentées en juste à temps par des lignes secondaires, avec la possibilité de combiner partiellement monde réel et monde des modèles, grâce aux propriétés d'interopérabilité de la technologie choisie.

2. Contexte de l'étude présentée

Les lignes de production sont au cœur de ces processus opérationnels spécialisés. Leur étude était plutôt passée de mode avec l'apparition des systèmes de production flexibles, mais elle connaît un regain d'intérêt, avec comme thèmes scientifiques des réflexions approfondies sur leur dynamique, leur pilotage, leur coordination, leur synchronisation ou encore leur interopérabilité (Zacharewicz, 2009). Ces problèmes sont d'autant plus difficiles à appréhender que les modèles mathématiques de comportement des processus ne sont toujours faciles à expliciter. En cela, la simulation apporte une possibilité d'étude et d'analyse qui se rapproche de l'expérimentation et du prototypage sans en avoir les inconvénients.

Comment aborder une réflexion ou une étude sur un système de production multi processus ou multi acteurs ? Déjà, l'étude d'une seule ligne de production est d'une grande difficulté. (Abdul-Kader, 2003) présente une méthode d'optimisation expérimentale *via* simulation d'une chaine de production constituée de m machines et de m-1 stocks d'encours inter-machines. Il montre le lien entre le temps de cycle et la manière de piloter la chaîne et l'importance de l'analyse de la production pour pouvoir optimiser la performance. Dans le même ordre d'idées, (Pujo *et al*, 2006) s'intéressent au déménagement progressif d'une ligne de production, qui devient ainsi temporairement deux lignes de production en interactions extrêmement fortes, l'une sur l'ancien site, et l'autre sur le nouveau. Là encore, la simulation est l'outil indispensable voire incontournable pour pouvoir prendre les bonnes décisions.

Dès que le nombre de processus augmente, les méthodes analytiques ne sont plus adaptées. Par ailleurs, il n'est plus possible d'étudier un processus ou une ligne de production indépendamment de ses interactions avec d'autres processus, car son fonctionnement dépend de ces autres processus. Aussi, une généralisation de l'étude par simulation de systèmes de production multi processus est nécessaire.

L'approche processus a tendance à s'étendre dans le monde industriel, particulièrement depuis la révision des normes qualité ISO 9001:2000. Ceci conduit les entreprises à une définition systématique des activités et tâches nécessaires à l'obtention du résultat désiré. Ce mouvement est extrêmement intéressant dans le sens où il permet de mieux appréhender le fonctionnement des systèmes de production. Il se traduit lui-même par la distinction entre processus opérationnel et processus décisionnel, ce qui permet de mieux appréhender les mécanismes comportementaux et d'en produire des modèles conformes, mais qui en contre partie augmente la difficulté de mise en œuvre de la simulation.

Toutefois, cette analyse par l'approche ISO 9001:2000 reste limitée, car les descriptions des processus qui en résultent produisent des modèles statiques qui ne laissent que peu de place à d'une part l'interprétation du comportement dynamique pour l'analyse de la productivité et à d'autre part l'identification d'interactions néfastes au bon fonctionnement de l'ensemble du système de production. Ces deux aspects sont pourtant fondamentaux. L'analyse du potentiel de chaque activité ne

peut être évaluée que du point de vue dynamique, tant sur le plan des liaisons avec les activités antérieures et postérieures que sur celui des liens avec les fonctions de pilotage. Cette analyse devient prépondérante lorsqu'elle concerne des activités clefs, telles que des activités de décision, capables d'influer sur la performance globale du système de production. Par ailleurs, il s'agit de détecter des situations contextuelles où le comportement global du système n'est pas satisfaisant. Dans les deux cas, la composante dynamique du comportement ne peut être appréhendée de manière calculatoire et nécessite une approche par simulation à évènements discrets.

En effet, la simulation à évènements discrets des systèmes, une fois leur modélisation validée, permet l'évaluation de performances, la recherche de configurations avantageuses, l'identification de situations critiques, l'analyse de l'aptitude à la réactivité ou encore la validation de règles de décision applicables lors du pilotage. Cet outil extrêmement performant fait d'ailleurs l'objet d'une offre conséquente de logiciels industriels de simulation, dont certains sont d'ailleurs extrêmement aboutis. Toutefois, il reste un certain nombre de points qui en limite l'usage. Nous citons à titre d'exemples : les barrières d'interopérabilité entre logiciels de simulation différents, l'absence de base formelle pour la description des modèles contribuant à des ambigüités, le problème de déterminisme et de causalité ou encore les difficultés de connexion avec le monde réel. Ces problèmes deviennent critiques dans le cas où il faut procéder à des analyses dynamiques de systèmes complexes multi processus relevant tant des aspects physiques que des aspects informationnels et décisionnels et intégrant des comportements asynchrones et des points de synchronisation.

En Productique, dans des domaines comme l'automobile ou l'aéronautique, ce cas est relativement fréquent dès que l'étude de simulation porte conjointement sur le système opérationnel de production multi flux et sur le système de pilotage qui lui est associé. Dans le domaine encore plus spécifique de la microélectronique, il faut être capable d'appréhender très rapidement et à différents niveaux les problèmes de goulot d'étranglement, de priorités, de répartition de tâches vers l'amont ou l'aval, tout en respectant des contraintes temporelles fortes (dates de livraison, durées de cycle…) (Ramamurthi *et al*, 2005). Par ailleurs, (Rose, 2000) a montré que des modèles trop simplifiés ne permettent pas d'appréhender tous les problèmes. Or, plus un modèle est complexe, plus sa formalisation analytique pose de problèmes pour l'obtention de solutions. La simulation est alors l'outil idéal pour l'aide à la mise au point des règles de décision du système de pilotage (Miyashita *et al,* 2004).

Nous proposons d'utiliser G-DEVS (Giambiasi *et al*, 2000) comme formalisme unificateur de modélisation pour simuler et donc analyser le fonctionnement simultané de processus distribués de natures différentes. Nous présentons en conséquence dans cet article une solution, fondée sur un environnement G-DEVS / HLA (Zacharewicz *et al*, 2006), permettant de décrire des processus opérationnels, décisionnels et informationnels, de les transformer en modèles G-DEVS, et enfin de les exécuter dans une simulation distribuée ouverte à l'interopérabilité avec d'autres logiciels par la compatibilité HLA.

Après avoir décrit les aptitudes de cet environnement à l'édition de processus *via* des formalismes usuels de description, nous exposerons les techniques permettant la transposition, visant à l'interopérabilité, de ces formalismes en modèles G-DEVS et la constitution d'un système de simulation distribué.

Nous donnerons quelques exemples concrets issus de l'industrie de la micro électronique, où les processus tant décisionnels qu'opérationnels sont relativement complexes et nous conclurons par une revue de quelques perspectives d'applications aux lignes de production alimentées en juste à temps par des lignes secondaires, avec la possibilité de combiner partiellement monde réel et monde des modèles, grâce aux propriétés d'interopérabilité de la technologie choisie.

3. Modélisation de processus

Nous avons choisi de privilégier la cohabitation des deux types de modèles de processus que nous rencontrons systématiquement dans tous les systèmes manufacturiers, à savoir l'un orienté flux physiques, pour lequel nous avons choisi la norme JIS Z 8206, et l'autre orienté flux informationnel, pour lequel nous nous appuyons sur une représentation de type Workflow.

3.1. *Processus opérationnels*

Les processus manufacturiers font l'objet de modélisation depuis les débuts de l'ère industrielle. Les premières représentations intéressantes datent des années vingt, avec les travaux des époux Gilbreth, qui proposaient un ensemble de 26 symboles graphiques correspondant aux différents équipements types de l'époque (Gilbreth and Gilbreth, 1921).

Cette première représentation a fait l'objet en 1947 d'une standardisation, avec un vocabulaire ramené à 5 symboles correspondant aux 5 grandes classes d'activités rencontrées dans un système manufacturier :

- ○ : le cercle symbolise une opération, c'est-à-dire l'apport d'une valeur sur le produit (transformation, assemblage…),
- ⇨ : la flèche symbolise un transfert ou un déplacement du produit,
- ▽ : le triangle isocèle symbolise un stockage de produits,
- ⊃ : ce symbole représente une attente, c'est-à-dire un stockage temporaire, le plus souvent au pied d'un équipement,
- □ : le carré symbolise les activités de contrôle et d'inspection.

Cette représentation, reprise par la norme japonaise JIS Z 8206, nous apparait comme le meilleur standard pour représenter les processus opérationnels de part sa reconnaissance des acteurs du domaine. De ce postulat, un éditeur graphique a été développé au LSIS : LSIS_PME (Production Model Editor) (figure 1), il permet de décrire des processus et de les sauvegarder au format XML.

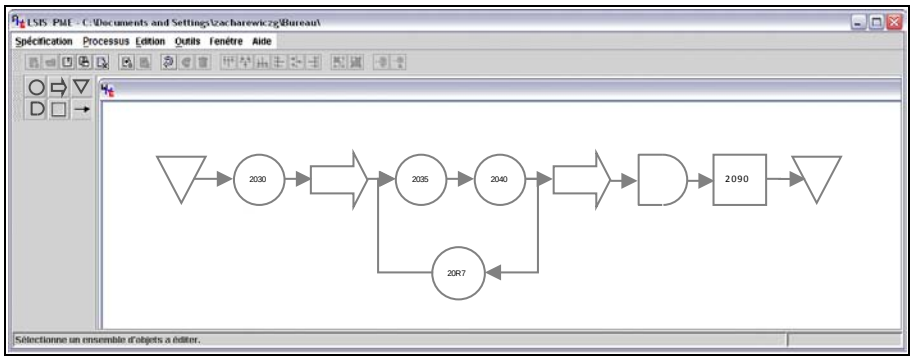


Figure 1. *Description d'un processus opérationnel norme JIS Z 8206 via LSIS-PME*

3.2. *Processus décisionnels*

Pour les processus décisionnels, nous nous sommes référés à la représentation Workflow (WfMC, 1999) dont l'utilisation est manifeste. Nous avons là encore privilégié une représentation graphique voulue universelle, et nous avons élaboré un éditeur graphique de Workflow LSIS_WME (Workflow Model Editor) (Zacharewicz *et al*, 2008) respectant une représentation graphique simplifiée des concepts énoncés par la Workflow Management Coalition (WfMC, 1999) (figure 2).

Le langage graphique est alors extrêmement simple (Zacharewicz *et al*, 2006) :

- ○ : le cercle symbolise soit les opérations initiales et finales, soit des routages convergents ou divergents, permettant de définir tous les opérateurs logiques (c'est l'indication interne qui les différencie),
- □ : le carré symbolise une opération atomique ou composite sur un item (un produit).

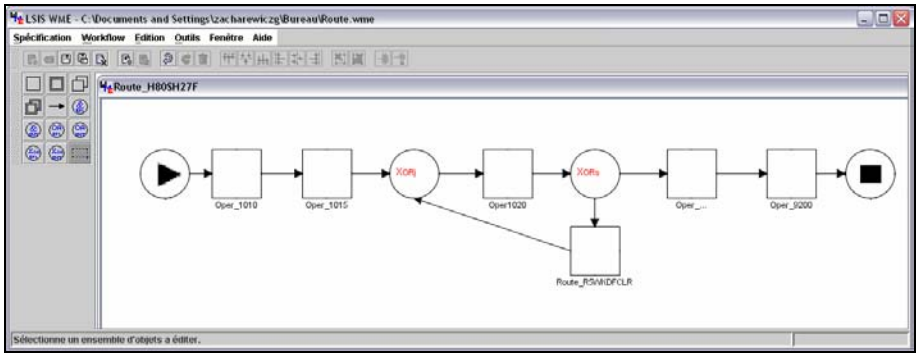


Figure 2. *Description d'un processus informationnel en Workflow via LSIS-WME*

Là encore, le modèle graphique du Workflow est transformé dans un format textuel, défini comme un langage pivot et sauvegardé au format XML.

Bien entendu, d'autres modes de description de processus opérationnels et décisionnels pourraient être implémentés selon le même principe : il existe de nombreuses possibilités de représentation des processus.

3.3. *Différents niveaux de processus décisionnels*

Généralement, la progression des produits dans un processus opérationnel doit faire l'objet d'un suivi. Nous pouvons donc en conclure qu'il existe en parallèle de chaque processus opérationnel un processus informationnel se déroulant simultanément et assurant les fonctions de conduite rapprochée. Il s'agit du suivi de la production et de l'archivage des informations de traçabilité, mais également du lancement des ordres de fabrication, c'est-à-dire du déclenchement des actions sur le processus opérationnels.

Ce premier niveau décisionnel élémentaire est complété par une couche décisionnelle de type pilotage d'atelier (*Control/Production*) et par une couche de contrôle qualité (*Process Control*). L'ensemble de ces fonctions forme le noyau d'un Système de Pilotage de Production *Manufacturing Execution System* (MES).

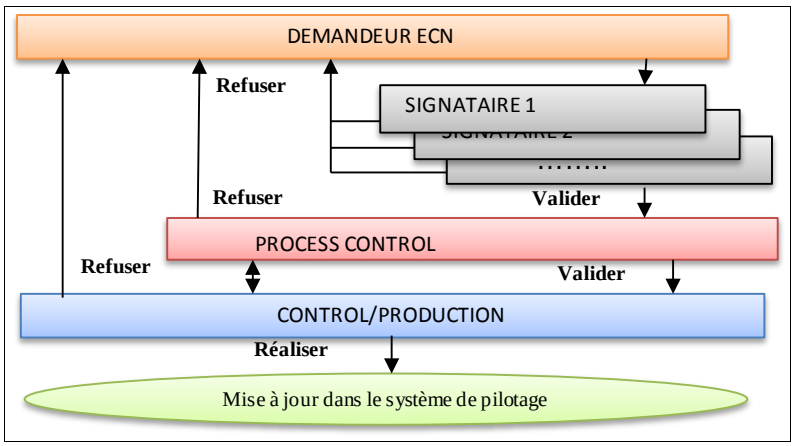


Figure 3. *Différents niveaux de pilotage*

Ces fonctions de pilotage ne sont pas suffisantes, en particulier lorsque le procédé est en limite de capabilité et qu'il faut fréquemment prendre des décisions de recalibrage des machines et de rectification des limites sur les cartes de contrôle. C'est le cas dans l'industrie micro électronique, où apparaît la notion de pilotage du pilotage : nous y rencontrons des Workflows informationnels et décisionnels visant à la requalification du procédé en fonction d'éléments ou d'informations en provenance de la production.

Considérant la nature stratégique des décisions à prendre, un ensemble de vérifications et de précautions s'avère indispensable. Cela se traduit par le recours à des Workflows de haut niveau décisionnel, appelés ECN (*Engineering Change Notification*), gérant les demandes de modification du paramétrage du processus opérationnel et les validant par approbation par un ensemble d'experts. Le schéma de la figure 3 illustre la relation entre les différents niveaux de pilotage. Nous proposons d'élaborer un système de Modélisation et Simulation permettant d'assister les prises de décisions stratégiques des experts.

4. Formalisation d'un processus dans le formalisme DEVS

Divers formalismes rivalisent pour modéliser des systèmes complexes à évènements discrets : citons les Statecharts, les réseaux de Petri ou DEVS (Zeigler *et al,* 2000). Nous utilisons ce dernier pour les raisons suivantes. Il permet de modéliser facilement des systèmes complexes. De plus, il intègre intrinsèquement le temps, permet de manipuler conjointement les concepts d'état et d'évènement et propose une sémantique opératoire claire et indépendante des modèles.

4.1. *Rappel sur DEVS, G-DEVS*

4.1.1. *Discrete Event System Specification (DEVS)*

(Zeigler, 1976) définit une spécification formelle de modèles à événements discrets : DEVS (*Discrete Event System Specification*). Il introduit notamment la possibilité d'évolution autonome du modèle grâce à la durée de vie des états, associée à une fonction de transition interne. DEVS se définit comme un formalisme abstrait universel indépendant de l'implémentation. Le concept de modèles basiques-couplés, introduit ultérieurement (Zeigler *et al*, 2000), fournit un moyen de construire des modèles composés, en réutilisant des descriptions stockées.

Un modèle atomique représente un système comportant des entrées et des sorties continues par morceaux et contient un modèle à événements discrets de type DEVS définissant le comportement du modèle. Les signaux d'entrée, de sortie et les états sont abstraits par une forme constante par morceaux où les seuils sont considérés comme des événements discrets. Formellement, un modèle atomique M est spécifié par un septuplet :

$M = < X, S, Y, \delta_{int} , \delta_{ext} , \lambda, D>$, où

- $\mathbf{X}$: ensemble des types des événements externes,
- $\mathbf{S}$: ensemble des états séquentiels,
- $\mathbf{Y}$: ensemble des types d'événements de sortie
- $\delta_{int} : \mathbf{S} \rightarrow \mathbf{S}$ fonction de transition interne définissant les changements d'état dus à des événements internes,
- $\delta_{ext} : \mathbf{ST} \times \mathbf{X} \rightarrow \mathbf{S}$ fonction de transition externe définissant les changements d'état dus à des événements externes,

L'ensemble **ST** des états totaux du système est ST = {(s, e) s ∈ S, o ≤ e ≤ D (s)}, où **e** représente le temps écoulé dans l'état s,
λ : **S** → **Y** fonction de sortie,
δ : **S** → **R+** ∪ ∞ fonction durée de vie des états.

Les quatre éléments dans le septuplé nommés δ_{int}, δ_{ext}, λ et δ sont les fonctions caractéristiques, S est l'ensemble des variables d'état et X, Y sont respectivement les ensembles d'événements d'entrée et de sortie.

4.1.2. *Représentation Graphique de Modèles (DEVS)*

Dans l'objectif d'une rationalisation et d'une composition modulaire hiérarchique, (Song and Kim, 1994) ont proposé une représentation graphique normalisée des modèles atomiques, intégrés dans un bloc comportant des ports d'entrées / sorties. La figure 4 introduit un exemple de DEVS atomique, avec :

- Un port d'entrée 'A', qui peut recevoir la valeur : a,
- Un port de sortie 'B', qui peut émettre les valeurs : b1 ou b2,
- La variable d'état 'Phase' qui prend les valeurs : Init, Reception, Queue
- Une variable d'état 'i' qui prend des valeurs numériques entières.

L'état des modèles graphiques DEVS est au minimum défini par la variable d'état « *Phase* » (ses valeurs étant représentées encerclées). Une durée de vie peut être associée à chaque valeur de phase (chiffres à l'intérieur des cercles). Les arcs pleins représentent les transitions externes δ_{ext}; par exemple, le code "**A ? a**" sur un arc de ce type indique que l'état du modèle transitera en recevant un évènement d'entrée de valeur "**a**" sur le port d'entrée "**A**". Les arcs en pointillés représentent les transitions internes δ_{int} : s'il s'écoule une durée de vie **δ** dans une phase source de ce type d'arc, le code "**B ! b1**" montre que l'état du modèle transitera et émettra, par la fonction de sortie **λ,** un évènement de sortie de valeur "**b1**" sur le port de sortie "**B**" (Figure 4). Notons que des conditions sur la valeur des variables d'état peuvent venir compléter la définition d'une transition (e.g. la transition **B ! b2** et associée d'une condition sur la variable d'état **i**, « **if i == C** », C étant une valeur paramètre constante).

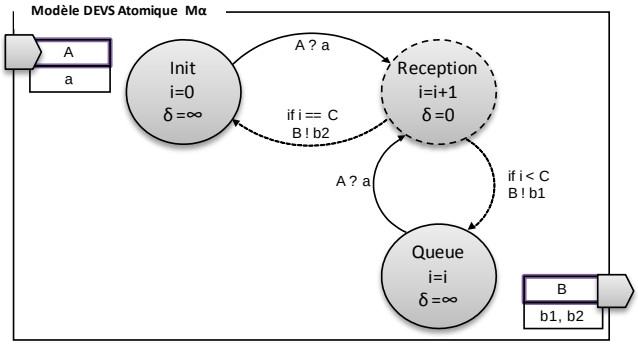


Figure 4. *Représentation Graphique d'un Modèle DEVS Atomique*

4.1.3. *Modèle DEVS couplé*

Un modèle couplé (figure 5) est un modèle structurel. Cette structure est décrite par l'interconnexion de modèles de base. Chaque modèle de base du modèle couplé interagit avec d'autres modèles pour produire un comportement global.

Les modèles de base sont soit des modèles atomiques soit d'autres modèles couplés figurant dans une bibliothèque, le couplage de ces modèles se réalise de façon hiérarchique comme indiqué Figure 6 a.

Un exemple de représentation graphique d'un modèle couplé MCχ. (figure 5) montre un modèle constitué d'un modèle couple MC_B et de deux modèles atomiques MA_A et MA_C. Les couplages inter-modèles (EIC : couplages d'entrées externes, IC : couplages internes et EOC : couplages de sorties externes) véhiculent des évènements externes, internes et de sorties.

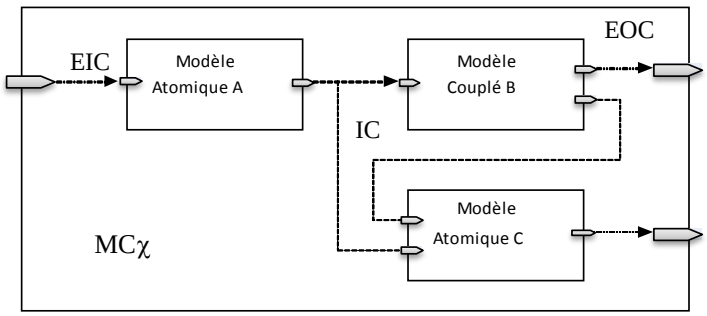


Figure 5. *Représentation graphique d'un modèle DEVS couplé*

4.1.4. *Simulateur DEVS*

Parallèlement à l'élaboration des différents modèles DEVS présentés précédemment, B.P. Zeigler a développé le concept de simulateur abstrait (Zeigler *et al,* 2000). L'architecture de simulation est dérivée de la structure hiérarchique des modèles couplés DEVS. Un simulateur abstrait représente une description algorithmique permettant de mettre en œuvre les fonctions du modèle, afin de générer son comportement. Un tel simulateur est obtenu en faisant correspondre à chaque élément du modèle un composant du simulateur. La construction d'un simulateur indépendant du modèle permet une séparation, au niveau réalisation, des parties modélisation et simulation.

Pour effectuer une simulation, une hiérarchie de processeurs, équivalente à la hiérarchie des modèles, est construite. A chaque composant du modèle est associé un processeur de la structure hiérarchique du simulateur (Figure 6 a). Chaque processeur participe à la simulation en exécutant les fonctions qui expriment la dynamique du modèle. Les processeurs sont :

- le **Simulateur** qui assure la simulation des modèles atomiques en utilisant les fonctions définies par DEVS,
- le **Coordinateur** qui assure le routage des messages entre les modèles couplés en fonction des définitions de couplage,
- le **Coordinateur Racine** qui assure la gestion globale de la simulation. Il ordonne le début et la fin de la simulation et gère l'horloge globale.

La simulation s'effectue par l'échange de messages spécifiques (Zeigler *et al,* 2000) entre les différents processeurs par les liens décrit dans la Figure 6 a).

4.1.5. *G-DEVS*

G-DEVS (Generalized Discrete EVent Specication) (Giambiasi *et al,* 1995 ; Giambiasi *et al,* 2000) est une extension du formalisme DEVS (Zeigler, 1976 ; Zeigler, 1984) qui permet d'intégrer en lieu et place d'entrées/sorties constantes par morceau des entrées/sorties polynomiales par morceau. Les valeurs des évènements d'entrées/sorties ne sont plus uniques, mais la définition du modèle et son évolution demeurent dirigées par l'apparition discrète de ces nouveaux événements : les trajectoires d'entrées/sorties sont définies polynomiales par morceau, chaque polynôme d'ordre N n'est décrit que par ses N coefficients et le passage d'un morceau à un autre demeure un évènement discret. Similairement au formalisme DEVS classique, nous retrouvons les notions de modèles atomiques, de modèles couplés et les représentations graphiques associées. Le couplage entre modèles reste réalisé selon une approche hiérarchique.

Le concept de simulateur abstrait introduit par (Zeigler *et al,* 2000) pour définir la sémantique de la simulation est réutilisé pour les modèles G-DEVS. L'architecture du simulateur est pareillement dérivée de la structure hiérarchique du modèle. La spécificité d'une simulation G-DEVS réside donc dans la définition d'un événement, elle est caractérisée par une liste de valeurs de coefficients par opposition à une valeur constante et unique dans DEVS.

4.2. G-DEVS/HLA

HLA (*High Level Architecture*) est une spécification d'architecture logicielle créée pour le *Department of Defense* (DoD) américain, *via* son département 'Modélisation & Simulation' (DMSO) pour pouvoir exécuter une simulation globale composée de simulations distribuées et hétérogènes. Dans HLA, chaque simulation participante est appelée fédéré, chaque fédéré dialogue avec les autres fédérés dans une fédération HLA. HLA est normalisé IEEE. Par exemple, la norme IEEE (M&S) P1516.2-2000 décrit comment s'effectuent les communications entre modèles et la mise en application *via* un « coordinateur central » : le RTI (Run Time Infrastructure) (e.g. figure 6b). Les fédérés interagissent en utilisant des services proposés par le RTI. Notamment, ils peuvent 'publier' pour informer de leur intention de fournir de l'information et 'souscrre' pour récupérer de l'information provenant d'autres fédérés. Ces informations ont des structures classiques en

Programmation Orientée Objet. Deux types d'objets échangés dans HLA peuvent être singularisés : il s'agit des classes d'Objet, persistants pendant la durée de la simulation, et les classes d'Interaction, qui concernent les informations transmises non persistantes entre deux fédérés donnés (IEEE, 2000). Enfin, pour respecter les relations de causalité temporelles lors de la simulation, HLA propose des mécanismes de gestion et d'avancement du temps, avec les algorithmes de synchronisation de simulation distribuée (optimistes et pessimistes) (Fujimoto 1998).

Pour créer des modèles G-DEVS compatibles HLA, nous avons étendu le logiciel de M&S graphique LSIS_DME (pour DEVS Model Editor) (Zacharewicz *et al,* 2006) afin de permettre une simulation en mode distribué. La structure d'un modèle G-DEVS peut être maintenant décomposée en différents modèles qui seront implémentés sur des ordinateurs différents connectés à internet. Chaque modèle est nommé fédérés HLA (figure 6b), l'ensemble constituant une fédération HLA (c'est-à-dire un modèle couplé G-DEVS distribué) au sens décrit au paragraphe précédent.

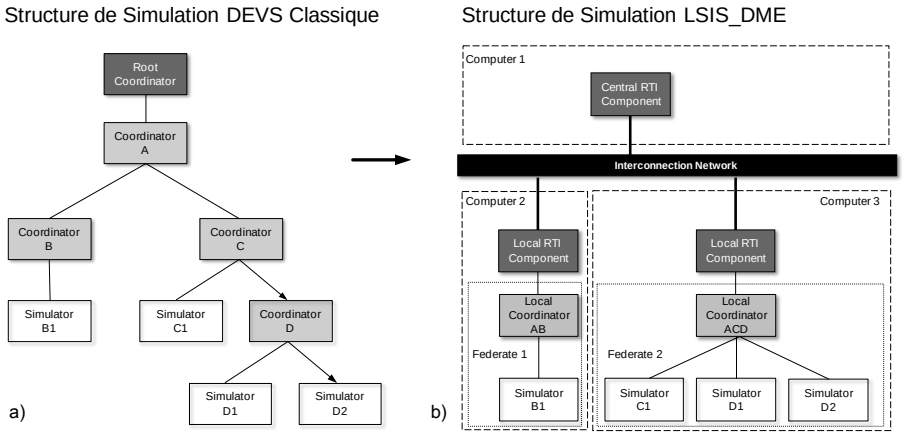


Figure 6. *Structure de simulation distribuée G-DEVS / HLA*

Cet outil de simulation, associé avec LSIS_PME & LSIS_WME qui sont totalement interopérables (au sens de l'interopérabilité unifiée et fédérée (Zacharewicz *et al*, 2009)), nous fournit un environnement de simulation pour systèmes de production multi processus. Cet outil est ouvert à l'interopérabilité avec d'autres composants logiciels hétérogènes. En particulier certaines applications développées par l'entreprise, difficile à intégrer au modèle, peuvent être ajoutées et intégrées à la simulation distribuée par leur passage à la compatibilité HLA (Zacharewicz *et al*, 2009). Les processus de collaboration avec des sous traitants et des partenaires projets peuvent aussi être considérés comme entrées/sorties du modèle (modèles boites noires) permettant ainsi de traiter certains problèmes de confidentialité rencontrés dans la réalisation de l'interopérabilité de 3ème génération.

4.3. *Transformation de processus en modèles G-DEVS couplés*

La figure 7 décrit la chaîne méthodologique de transformations nécessaires pour passer des descriptions orientées 'processus' (Workflow et JIS Z 8206) en modèles G-DEVS. Ces transformations sont fondamentales pour deux raisons. D'une part, G-DEVS est dans notre cas particulièrement adapté, car nous pouvons attacher à chaque évènement un vecteur de valeurs qui permet de répondre au caractère complexe de la description des produits en cours de production. D'autre part, les descriptions orientées 'processus' ne sont pas réputées pour être fondées sur des concepts formels forts, le formalisme G-DEVS permet la vérification de propriétés sémantiques et la validation des modèles par simulation.

Dans cette démarche de formalisation, nous proposons, dans une première étape, l'utilisation d'un langage formel de description d'un processus, compatible avec une représentation XML. Nous avons créé ce langage en référence à la représentation XML des Workflows proposées par la WfMC et aux modèles de processus, il reprend et complète les éléments graphiques introduits en §3.1 et 3.2. De plus, la notion implicite de ressource dans les modèles graphiques opérationnels et de Workflow est ajoutée. Nous partons en effet de la nécessité de modéliser l'allocation des ressources sur les tâches et de traiter cette action dans la simulation des modèles.

Grâce à cela, la validation syntaxique du modèle est obtenue en référence à une DTD de Workflow (*Document Type Definition*) élaboré par la WfMC (WfMC, 2005). Par extension, nous avons défini une DTD pour les processus opérationnels. Un processus P est alors défini par une structure algébrique telle que :

P = (Nom, RESS, A, Ct, ARC, IT, ST), où :

Nom, le nom du processus,
RESS, l'ensemble des ressources du processus,
A, l'ensemble des tâches du processus,
Ct, l'ensemble des contrôleurs du processus,
ARC, l'ensemble des arcs représentant le flux d'item du processus,
IT, l'ensemble des items (produits semi finis) du processus,
ST, l'ensemble des stocks du processus.

Ce formalisme rationalise les modèles mais il donne une description uniquement statique du système. Dans l'objectif d'observer l'évolution dynamique du système, ces processus doivent être transformés en modèles simulables (i.e. G-DEVS) puis chargés et exécutés dans LSIS_DME. La démarche de transformation est automatisée par un traitement XSL des modèles XML de processus qui génère un fichier XML G-DEVS. Ce XML instancie des modèles issus d'une bibliothèque G-DEVS qui contient les éléments génériques d'un processus (i.e. les éléments de P). Ils sont ensuite connectés et intégrés au modèle couplé G-DEVS du processus global. Un algorithme de transformation a été proposé par (Zacharewicz, 2008) pour le passage Workflow vers G-DEVS. Nous l'avons complété pour l'utiliser également pour le passage d'un modèle « processus opérationnel » en G-DEVS.

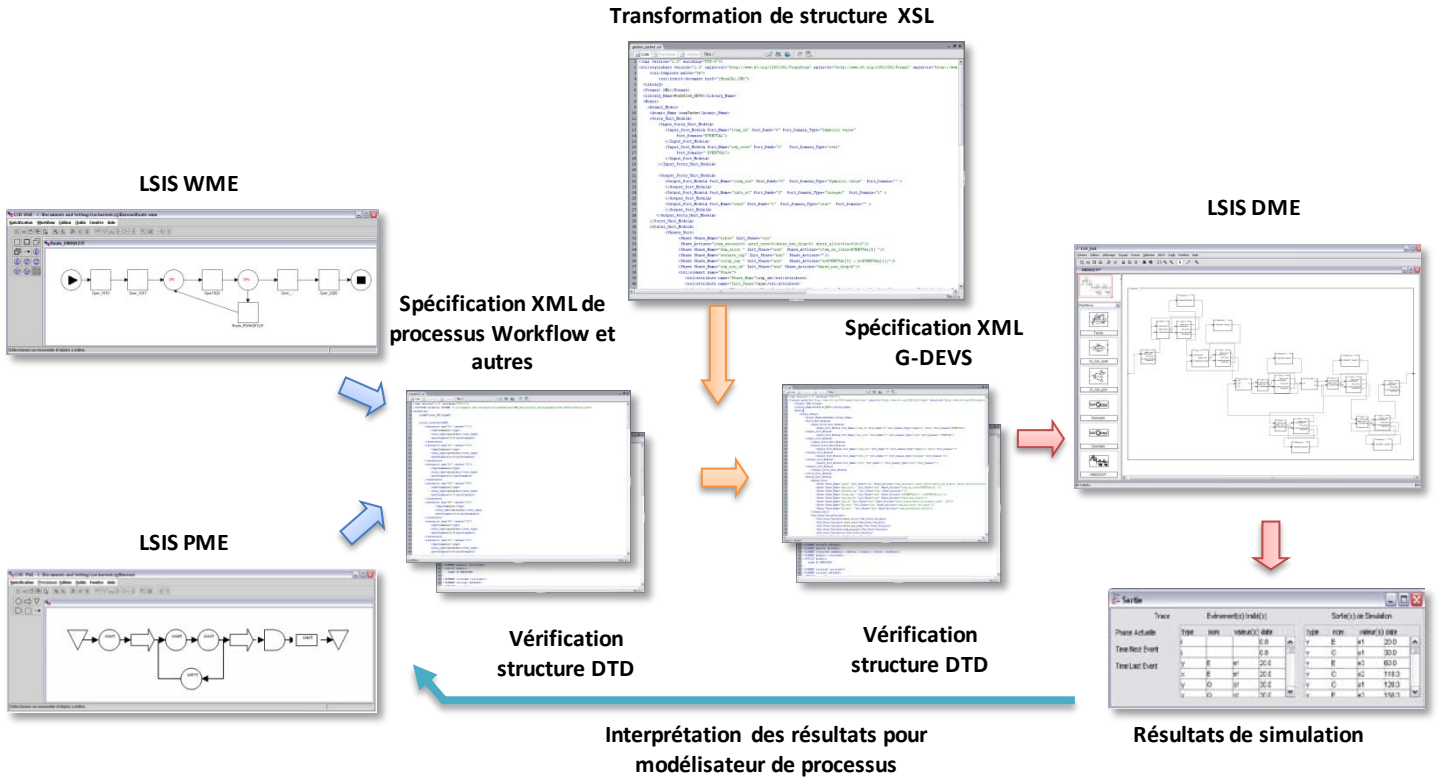


Figure 7. *Transformation d'un modèle de processus en modèle G-DEVS couplé*

Chaque modèle atomique de la bibliothèque Workflow fait donc l'objet d'une transcription vers une description formelle en modèles G-DEVS atomiques. Nous allons à présent exposer deux exemples concrets de processus, l'un au niveau le plus élevé des demandes de modification procédé (§5.1) et l'autre au niveau d'une étape du processus opérationnel de production des Wafers (§5.2).

5. Exemples de processus dans le domaine de la production microélectronique

Les microcircuits sont élaborés par lots sur des wafers. Certains wafers peuvent supporter des circuits différents, pour les petites séries par exemple. Dans tous les cas, les équipements traitent les wafers par lot. Ces wafers suivent donc des gammes de transformations appelées *'route*' dans le secteur de la microélectronique. Un produit peut avoir plusieurs routes possibles, une principale et d'autres '*alternatives*'. Chaque route est décomposée en '*opérations'*, décomposées à leur tour selon un '*script'*. Cela représente en fait des gammes cycliques, dans le sens où les produits passent plusieurs fois sur les mêmes équipements, le procédé étant cyclique. Chaque opération est décrite par une suite de '*steps'* (entre 10 et 30 *steps* par opération), correspondant chacun à un '*event'*, c'est-à-dire à une configuration ou à une tâche de l'équipement correspondant (figure 8).

5.1. *Workflow d'une demande de modification ECN*

Le Workflow que nous allons étudier maintenant concerne le traitement d'une ECN (Engineering Change Notification). Il s'agit de déployer sur le processus opérationnel un changement de paramètres pour améliorer les performances de production ou corriger un défaut identifié. Cela peut consister à modifier des paramètres d'équipements, en fonction de l'utilisation précédente. Par exemple, un autoclave de dépôt de matière sur les wafers voit une évolution de la surface interne de son enceinte, sur laquelle les dépôts s'effectuent également à chaque opération sur des wafers. Dans le temps, cela nécessite des ajustements de procédé : température, durée…

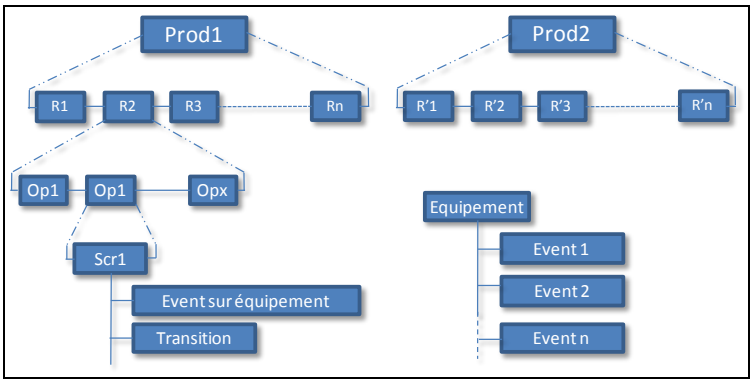


Figure 8. *Décomposition hiérarchique d'une route STMicroElectronics*

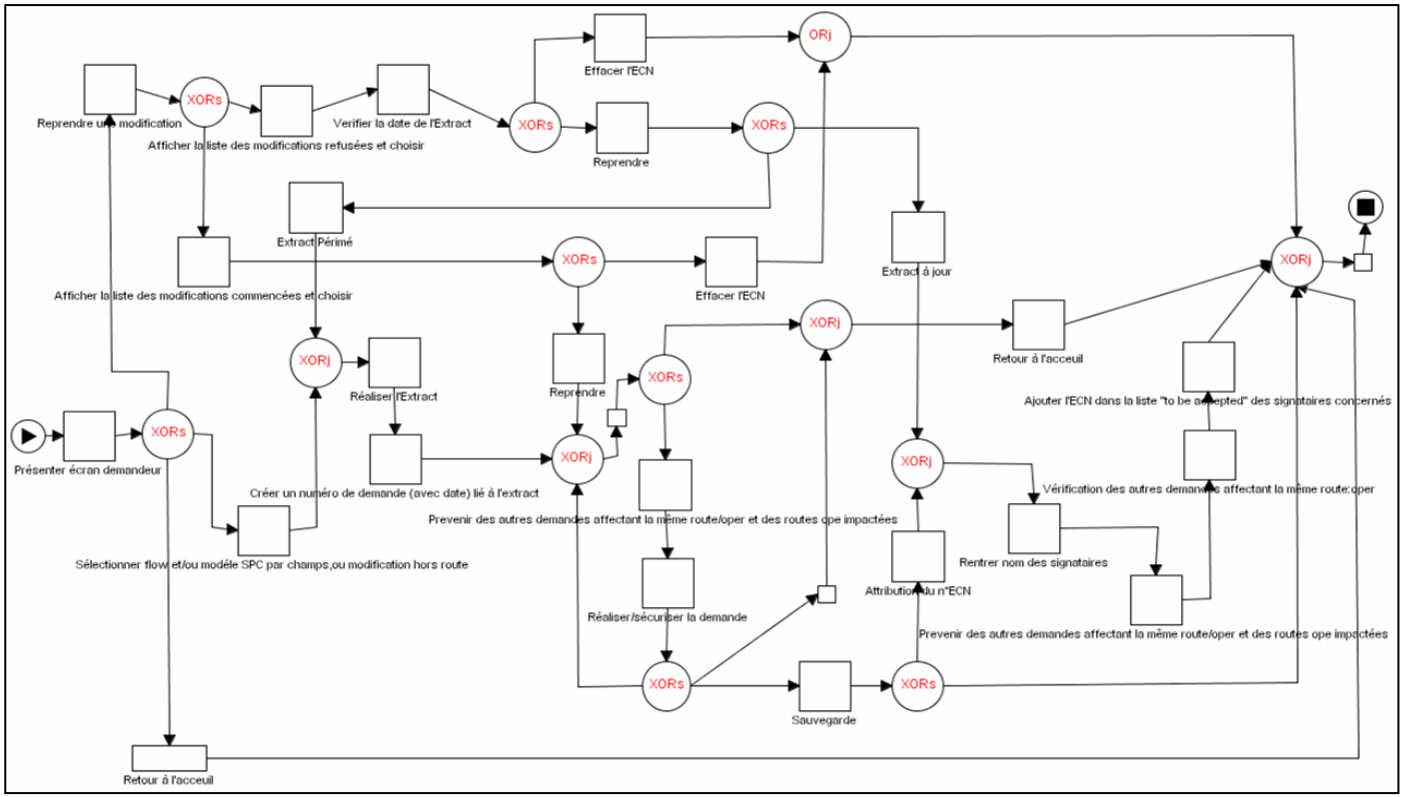


Figure 9. *Workflow général d'un ECN*

Un autre exemple concerne la modification des limites de contrôle d'une caractéristique automatiquement instrumentée sur un équipement donné. Cette modification peut être sans conséquence sur une grande proportion des produits et induire des conséquences néfastes et infondées sur quelques références. Le déploiement de telles modifications ne peut donc être déconcerté, mais doit répondre à un ensemble de contraintes de sécurisation, au premier rang desquelles figure un circuit de validation de la demande de changement auprès d'un ensemble complet d'experts du système de production, chacun étant spécialisé sur une vue ou un aspect scientifique du procédé de production des microcircuits. Le recours à la plateforme de M&S permet donc d'assister les experts dans la prise de décision par l'exploitation de résultats de scenarios de simulation.

Le schéma général de la figure 9 présente un modèle de vision globale de l'enchaînement du processus ECN de validation, développé dans LSIS_WME. Le processus démarre par une demande de modification ECN (Engineering Change Notification) et va jusqu'à la mise en place de cet ECN dans le MES. Parmi ces tâches, 4 représentent en fait le déclenchement de quatre autres Workflows. Nous soulignons donc la complexité de ce niveau décisionnel (des milliers de produits en cours de production, et des dizaines d'ECN en cours de validation) et l'intérêt du recours à la simulation pour évaluer le bon fonctionnement du dispositif.

La figure 10 présente, dans un objectif pédagogique, la transformation générée à partir d'une route Workflow simple également issue du domaine microélectronique. Elle permet d'identifier facilement les modèles G-DEVS instanciés à partir d'une route Workflow. Nous pouvons remarquer notamment qu'un composant Workflow « Operation » nécessite plusieurs modèles G-DEVS.

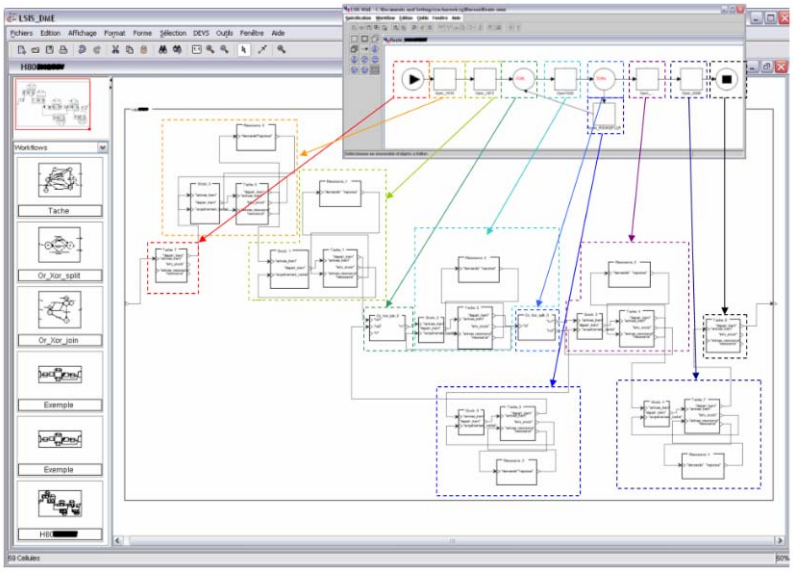


Figure 10. M*odèle graphique Workflow et modèle couplé G-DEVS équivalent*

Plus en détails, le schéma de la figure 11 présente un groupe de modèles atomiques associés à l'élément Workflow « opération » (ici OPER 1010), puis le détail d'un modèle générique G-DEVS de « ressource » employé dans une opération Workflow.

Ce groupe met en évidence l'apparition de modèles G-DEVS de « ressource » et de « stock » nécessaires à la réalisation d'une simulation réaliste, en effet ces éléments sont implicites au niveau Workflow mais doivent être gérés en simulation.

Le comportement du modèle atomique de « ressource » G-DEVS peut être résumé ainsi : Le modèle de ressource est initialement en attente de demande d'allocation. Lorsqu'une demande est reçue dans cet état, la ressource répond qu'elle est disponible. Suite à cela, la tâche qui souhaite l'utiliser lui communique la durée d'allocation. La ressource est alors occupée pour une durée donnée, et ne peut donc plus, dans cet état, être allouée à une autre tâche. En conséquence, les autres tâches effectuant une demande d'allocation recevront une réponse négative à leur requête. Lorsque la durée d'allocation est écoulée, l'état du modèle redevient libre et donc disponible à l'allocation par une tâche (l'allocation est simple dans ce cas mais les allocations multiples sont possibles).

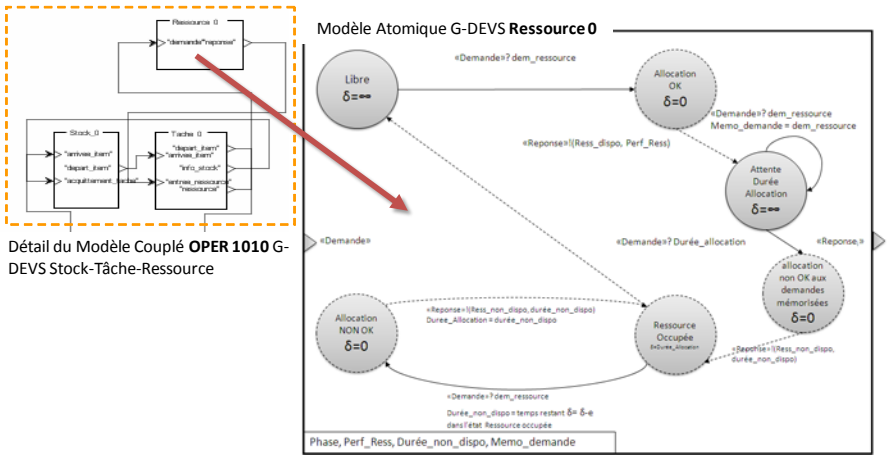


Figure 11. *Modèle atomique d'un modèle couplé de tâche*

5.2. N*iveau du Processus opérationnel : description d'une route*

Un modèle processus opérationnel est élaboré dans LSIS_PME en parallèle de chaque modèle de Workflow, transformé selon le principe décrit figure 7. Pour chaque composant basique (i.e. §3.1), un ensemble de modèles G-DEVS sont instanciés. Nous nous focalisons ici sur un modèle « event » utilisé dans une « opération » du modèle opérationnel. Nous mettons en évidence, à cette occasion, la différence entre les modèles G-DEVS « tâche » (pour la description Workflow) et le modèles G-DEVS « event » (pour la description du processus opérationnels).

La figure 12 présente, à titre d'exemple, un modèle G-DEVS atomique d'*event* du *process* PRP/PRM/PRN générique. Dans ce modèle, nous décrivons le comportement de la tâche correspondante, dés réception du lot jusqu'à son envoi au *step* suivant qui est généralement un *process* de mesure de type PRM/PRN. Le module DEVS *Event_Process* peut recevoir un lot de l'opération (*script*) précédente ou venant d'un service de *starter*, s'il s'agit de la première opération de la route. Pour cela, il vérifie d'abord si l'*event* concerné est de type USE-Plate : cet *event* générique doit être remplacé par le bon *event*, que nous trouvons dans la table de masque séquence associée au produit. Pour cela, il consulte cette table dans l'état *Consultation_TMSQ*. S'il ne trouve pas le bon *event*, alors le lot est '*holdé'* (c'est-à-dire que l'avancement du lot est suspendu jusqu'à ce qu'une décision spécifique soit prise). Par contre, s'il le trouve, il se met en état *Demande_Res* pour demander une ressource *via* le gestionnaire de ressource. Si la ressource est disponible, il passe dans un état *Allocation_Ressource* pour attendre la fin de traitement. Par contre, dans le cas où la ressource n'est pas disponible, il se met en *Attente_Libération_Res*. Ce module peut aussi recevoir un évènement externe *via* le port d'entrée *Recette* indiquant la non qualification d'une recette sur la ressource choisie. Pour cela, il se met en attente *Recette_Introuvable* pour attendre la qualification de la ressource.

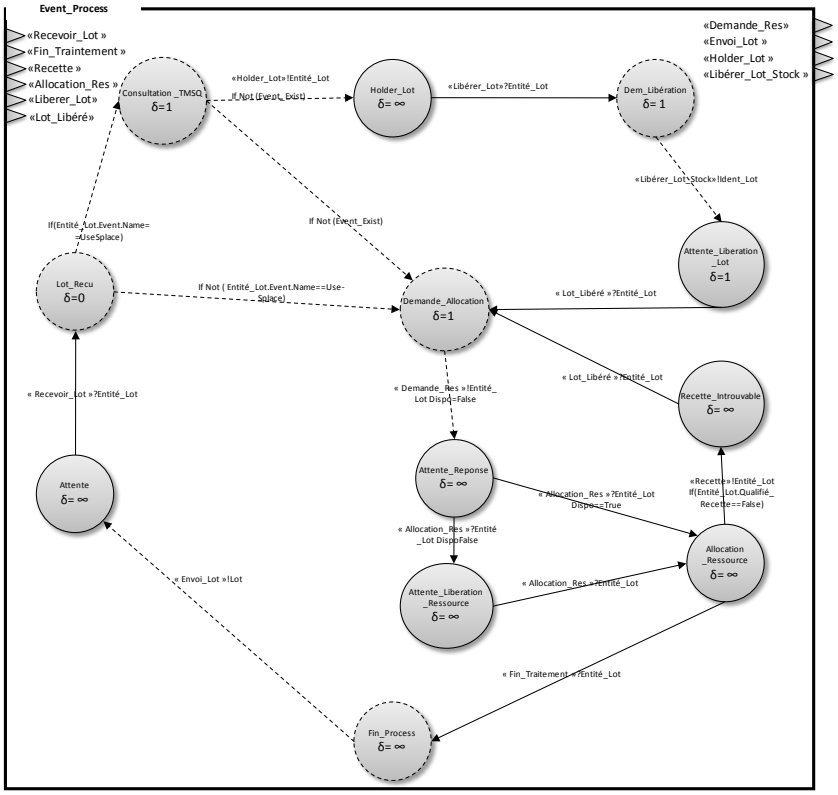


Figure 12. *Modèle DEVS d'un event du process PRP/PRM/PRN*

5.3. Structure de Simulation Distribuée HLA

La plateforme proposée (i.e. figure 13) permet l'interaction des composants logiciels depuis la création des modèles processus jusqu'à leurs simulations ou leurs exécutions en temps réel pour la conduite de Système. Ce système doit pouvoir faire intervenir des applications hétérogènes externes issues des métiers d'applications. L'approche d'interopérabilité « fédérée » retenue demande d'instaurer une communication entre les composants sans les intégrer. L'utilisation de la norme HLA permet d'assurer la communication, de part sa capacité à définir des fédérés sans avoir à recoder les composants, par un ajout d'une couche logicielle (Local RTI Component) permettant la Publication/Souscription à des messages HLA. La communication des applications distantes est relayée par un RTI et permet d'assurer, l'interopérabilité entre les applications utilisées dans le processus mais également à d'autres applications de se connecter à la plateforme.

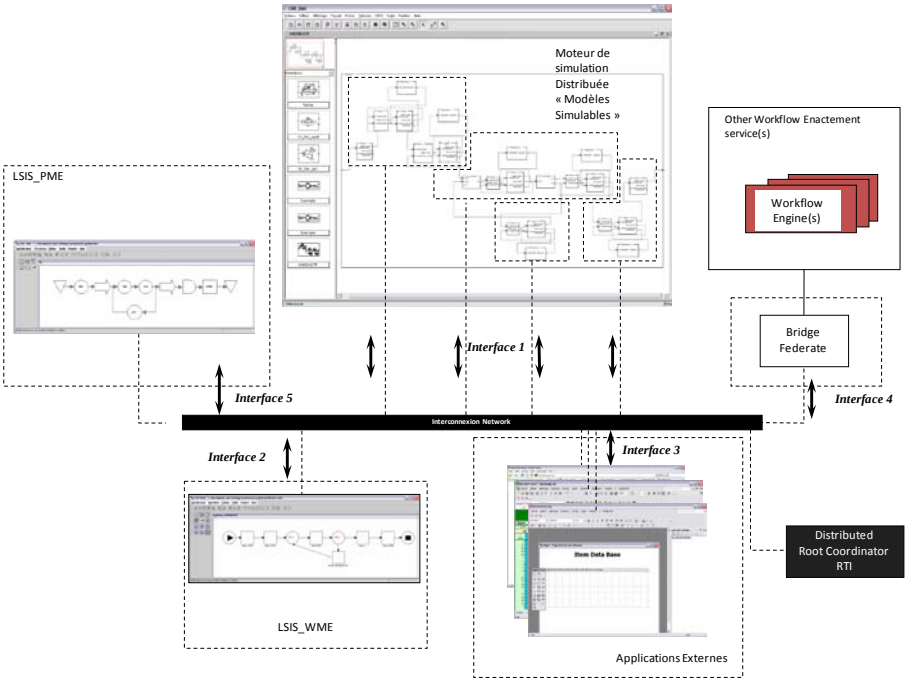


Figure 13. *Modèle DEVS d'un event du process PRP/PRM/PRN*

Nous présentons ici plus en détail la plateforme proposée dans la Figure 13. L'interface 1 connecte le moteur de simulation LSIS_DME aux autres composants de la plateforme. Ce composant reçoit les modèles processus transformés, depuis les applications spécifiques grâce aux interfaces 2 et 5, selon les processus décrit en figure 1&2. Le choix est porté sur l'implémentation de modèles « sources libres » paramétrables au niveau des modèles processus mais également au niveau des modèles exécutables (au niveau du moteur de simulation). Ce moteur de simulation

est le coordinateur du Workflow global distribué, il synchronise l'échange d'informations entre applications hétérogènes (interface 3) impliquées dans l'exécution globale, et déclenche leurs utilisations. HLA permet de faciliter la connexion de ces applications, non intégrables dans la simulation, qui était initialement développées dans un objectif d'utilisation locale et spécifique à un métier (parfois de façon *ad hoc* pour certains outils).

5.4. *Résultats*

Les résultats obtenus lors de la simulation par LSIS_DME permettent d'évaluer différents aspects de la performance des processus étudiés. Par exemple, le nombre d'items défectueux, le niveau de stock moyen, la durée moyenne de transfert des produits sont autant d'éléments qui peuvent être calculés lors d'une expérimentation avec un jeu d'essai. LSIS_DME permet également de consulter l'échéancier de tous les évènements sélectionnés dans le modèle (figure 14).

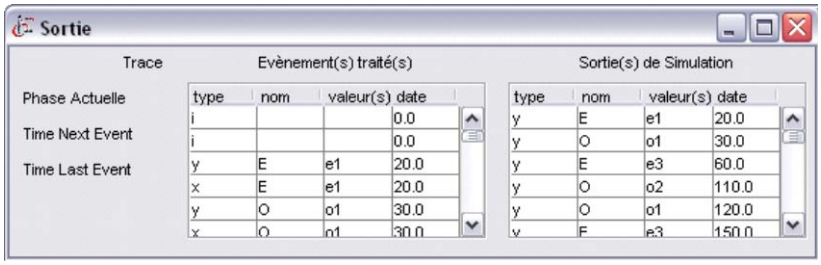


Figure 14. *Exemple d'échéancier de sortie de LSIS_DME*

Ceci permet de faire par la suite d'autres analyses statistiques, en utilisant un outil informatique de traitement statistique, un tableur par exemple. Le tableau 1 présente le résultat de simulation du Workflow simple OPER 1020 (i.e. §5.2).

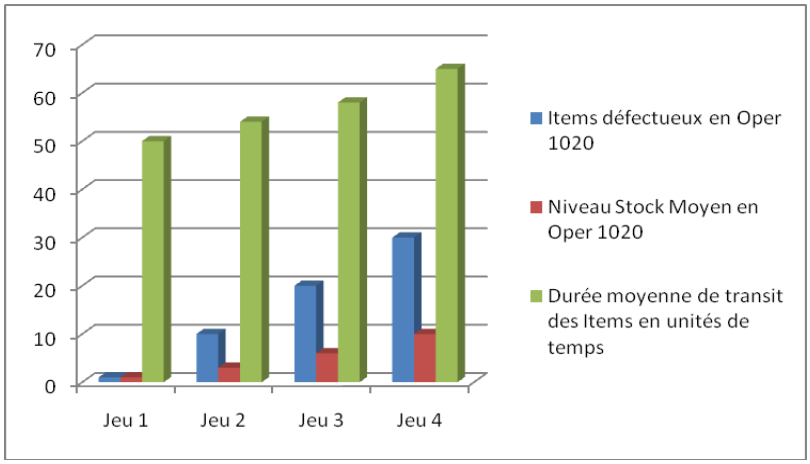


Tableau 1. *Exemple d'analyse comparative avec 4 jeux de données sur OPER1020*

Nous y observons, en particulier, la durée moyenne de transit des items. Ces données sont déterminées à partir de la trace de simulation contenant les événements reçus, traités et générés, du modèle couplé G-DEVS de la route HXX.

Ces informations permettent de déterminer une planification judicieuse des items et peuvent indiquer une remise en question des enchaînements de tâches. En particulier, l'analyse du Stock de l'Oper 1020 nous indique un niveau-moyen élevé, dans le cas d'un nombre d'items défectueux égal à trente pourcents. A partir de ce constat dans ce cas simple et dans l'hypothèse de trente pourcents de produits défectueux, nous proposons de paralléliser l'Oper 1020. Le modèle modifié indique, par simulation, que l'on peut réduire ainsi la durée de transit à environ 56 unités de temps.

6. Perspectives

Cette approche de simulation distribuée est très intéressante pour l'étude d'applications réelles multi lignes. Dès que ces lignes ne sont pas totalement indépendantes se posent différents problèmes de synchronisation et de réactivité. Cette dimension représentait d'ailleurs une limitation à l'extension aux lignes en flux poussés de nos travaux antérieurs sur les lignes de production en Juste à Temps (Ricciardi *et al,* 2001 ; Pujo *et al,* 2004), où les synchronisations n'ont pas besoin d'être anticipées.

Dans les unités de production manufacturières flexibles, nous observons de plus en plus comme architecture générale des processus 'terminaux' (c'est-à-dire des processus d'assemblage final des produits) des structures convergentes en 'arête de poisson' (figure 15) : tout au long du processus opérationnel principal viennent s'insérer le résultat de processus opérationnels secondaires d'approvisionnement et/ou de soutien, dont une partie peut être externalisée (El Haouzi *et al*, 2007). Cette structure est extrêmement classique en production synchrone, où les équipements sont somme toute dimensionnés en fonction de la cadence de la ligne principale. Cela pose le problème de la synchronisation interlignes. En production flexible, la synchronisation totale entre la ligne principale et ses lignes d'approvisionnement ne peut être envisagée, et il faut passer par des évènements de déclenchement. Ceci répond par exemple aux besoins de préparation à la demande de composants spécifiques montés par ailleurs sur une chaîne d'assemblage standard. Un bon exemple de cette 'personnalisation de masse' se rencontre sur les lignes d'assemblage des véhicules industriels : chaque véhicule fait l'objet d'une configuration spécifique à chaque client (Giard and Mendy, 2007). Pour cela, des préparateurs collectent l'ensemble des composants propres à chaque véhicule et les livrent au bon moment sur le poste de montage *ad hoc*. Cela nécessite une bonne synchronisation des opérations de préparation, dont le déroulement n'est pas toujours le même en fonction du nombre et de la nature des options en question propre à chaque véhicule.

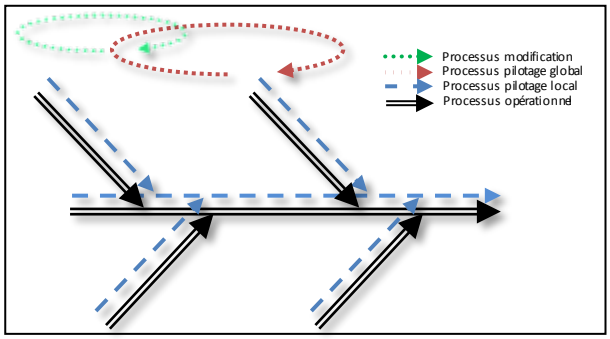


Figure 15. *Structures générales de processus d'assemblage final*

Face à cette problématique, la simulation multi processus distribuée est l'outil idéal pour ajuster le fonctionnement de l'ensemble du système de production. Entre autres problèmes pouvant être résolus, le choix de l'emplacement sur la ligne principale de capteurs d'information permettant le déclenchement des processus de préparation, chez un sous traitant ou dans la même installation, est un problème crucial qui contraint fortement la fiabilité de la synchronisation (El Haouzi *et al*, 2007).

De nombreux autres points intéressants à étudier peuvent être abordés *via* une simulation distribuée multi processus. Notamment, les problèmes de synchronisation en production de masse personnalisée (Chandra and Kamrani, 2004) demeurent une perspective intéressante. Précisément dans ce contexte, la simulation permettrait d'apporter des réponses à des problèmes identifiés, nous citons par exemple :

- la détermination du positionnement du point de découplage (Mason-Jones and Towill, 1999),
- l'étude du lissage de charge dans un réseau partenarial logistique (Ounnar *et al*, 2009), avec le réglage des différentes pondérations pour l'aide à la décision,
- la recherche du juste équilibre entre les approches 'lean' et 'agile' (Christopher, 2000), avec le besoin d'observer et de simuler l'intégralité de la chaine logistique,
- l'impact de nouvelles approches en ordonnancement multicritère (Pujo *et al*, 2009), avec l'ajout aux critères favorisant le 'lean' (Ounnar and Pujo, 2009) de nouveaux critères favorisant la synchronisation inter systèmes de production...

Enfin, une fois de bons mécanismes de décision élaborés et validés *via* une simulation globale multi processus du système de production, l'opérationnalisation de ces règles reste une étape difficile, car il existe toujours des différences potentielles entre le comportement des modèles et le monde réel. Là encore, un

environnement de simulation G-DEVS / HLA offre des perspectives rassurantes en terme de mise en œuvre et d'opérationnalisation du processus de décision (figure 16).

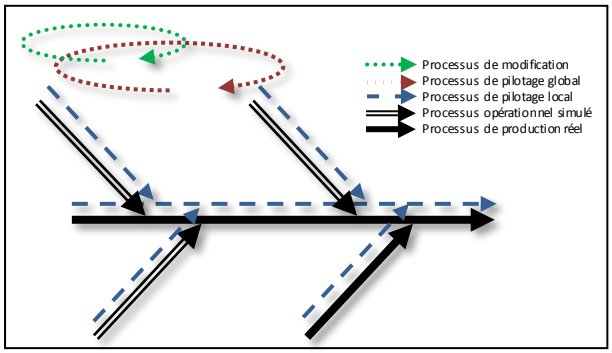


Figure 16. *Environnement G-DEVS / HLA pour le couplage simulations / réalité*

En effet, HLA a été conçu dès l'origine par le DMSO pour pouvoir assurer l'interopérabilité entre des modèles de simulation hétérogènes et des systèmes réels interagissant avec ces modèles. Il est donc tout à fait possible de connecter le modèle global multi processus sur une partie du système réel, en remplacement des modèles relatifs à cette partie. Ceci permet de tester par partie l'ensemble du système décisionnel et de s'assurer de la validité des différents modèles constitués.

7. Conclusion

Nous avons présenté ici un environnement G-DEVS / HLA dans lequel nous pouvons spécifier et simuler des systèmes de production multi processus, et plus particulièrement permettant de combiner des processus opérationnels, décisionnels et informationnels décrits selon des modes de représentation différents. L'ensemble de ces travaux s'inscrit dans l'esprit des grandes lignes directrices de la réflexion prospective exposée dans (Fowler and Rose, 2004) : simulation temps réel, distribuée, interopérable, multi formalismes…

Tout d'abord, nous avons constitué une bibliothèque de modèles G-DEVS génériques qui nous permet d'exploiter directement les formalismes orientés 'processus' couramment pratiqués dans le monde industriel. Ensuite, grâce aux propriétés de couplages et à notre algorithme de transformation de formalismes, nous proposons un environnement homogène de simulation distribuée basée sur une sémantique opératoire claire et non ambigüe. Pour cela, nous utilisons d'une part G-DEVS comme formalisme unificateur de modélisation, G-DEVS étant la seule solution permettant une analyse sémantique sérieuse et un contrôle des modèles de processus générés automatiquement, et d'autre part HLA comme solution de simulation distribuée permettant l'interopérabilité entre ces modèles.

Après avoir rappelé les principes de transformation des descriptions en formalismes 'processus' habituels en modèles formels G-DEVS, nous avons donné quelques exemples dans le domaine de la micro électronique. Ces exemples ont le mérite de révéler toute la complexité de ces systèmes, qui peuvent être ainsi abordés grâce à la puissance de calcul d'une approche distribuée et à la finesse de représentation autorisée par G-DEVS. La simulation distribuée, ainsi positionnée comme outil d'aide à la décision, se révèle être un atout majeur pour l'appréhension des systèmes complexes multi processus, décrits selon de multiples formalismes.

Nous allons poursuivre ce travail en nous concentrant sur la notion d'interfaçage et d'interopérabilité entre monde réel et monde simulé, avec de nombreuses perspectives d'applications pour l'aide à la décision en production industrielle.

Les travaux présentés ici sont financés dans le cadre d'un projet impliquant le Laboratoire des Sciences de l'Information et des Systèmes, le Conseil Général des Bouches du Rhône et ST Microelectronics – Zone industrielle de Rousset 13106 Rousset cedex, France, que nous remercions pour leur soutien.

8. Bibliographie

Abdul-Kader W., "Simulation of an Unreliable Production Line", *Journal of Decision Systems*, vol. 12 no. 1, 2003, p. 31-46.

Chandra C., Kamrani A., *Mass Customization: A Supply Chain Approach*, New York, Kluwer Academic, 2004.

Christopher M., "The Agile Supply Chain: Competing in Volatile Markets", *Industrial Marketing Management*, vol. 29 no. 1, 2000, p. 37-44.

El Haouzi H., Pannequin R., Thomas A., "Génération automatique de plateformes de simulation pour des systèmes organisés en flux tirés", *7e Congrès International de Génie Industriel*, Trois-Rivières, Québec, 5-8 juin 2007.

Fujimoto R. M., "Time Management in the High Level Architecture", *Transaction of the SCS Simulation*, vol. 71 no. 6, 1998, p. 388-400.

Fowler J. W., Rose O., "Grand Challenges in Modeling and Simulation of Complex Manufacturing Systems", *Transaction of the SCS Simulation*, vol. 80 no. 9, 2004, p. 469-476.

Giambiasi N., Escude B., Ghosh S., "G-DEVS A Generalized Discrete Event Specification for Accurate Modeling of Dynamic Systems", *Transactions of the SCS International*, vol. 17, no. 3, 2000, p. 120-134.

Giard V., Mendy G., "De l'approvisionnement synchrone à la production synchrone dans la chaîne logistique", *Revue française de gestion*, vol. 2 no. 171, Lavoisier, 2007, p. 65-88.

Gilbreth F., Gilbreth L., *Time and Motion Study as Fundamental Factors in Planning and Control*, New Jersey, The Mountainside Press, 1921.

Institute of Electrical and Electronic Engineers, *High Level Architecture (HLA) - Federate Interface Specification,* IEEE Standard for Modeling and Simulation (M&S). std 1516.2-2000, March 2001.

JIS Z 8206: Graphical symbols for process chart, *Japanese Standards Association,* 1982.

Mason-Jones R., Towill D.R., "Using the Information Decoupling Point to Improve Supply Chain Performance", *International Journal of Logistics Management*, vol. 10 no. 2, 1999, p. 13-26.

Miyashita K., Okazaki T., Matsuo H., "Simulation-based advanced WIP management and control in semiconductor manufacturing", *Proceedings of the 2004 Winter Simulation Conference,* Washington, DC, 5-8 Dec. 2004, R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds, p. 1943- 1950 vol.2.

Ounnar F., Pujo P., Mekaouche L., Giambiasi N., "Integration of a Flat Holonic Form in an HLA Environment", *Journal of Intelligent Manufacturing*, vol. 70 no. 1, 2009, p. 91-111

Ounnar F., Pujo P., "Pull control for Job Shop: Holonic Manufacturing System approach using multicriteria decision-making", *Journal of Intelligent Manufacturing*, 2009.

Pujo P., Pedetti M., Ounnar F., "Pilotage proactif des lignes de production kanban par modélisation DEVS et simulation temps réel", *MOSIM'04, 5° Conférence Francophone de MOdélisation et SIMulation*, Nantes, France, 1-3 septembre 2004.

Pujo P., M. Pedetti, and N. Giambiasi, "Formal DEVS modelling and simulation of a Flow-Shop relocation method without interrupting the production", *Simulation Modelling Practice and Theory,* vol. 14, 2006, p. 817–842.

Pujo P., Ounnar F., Blanc P., "Pilotage des systèmes manufacturiers *via* une analyse multicritère intégrant Produit, Ressource et Ordre. Application au pilotage en flux tiré d'un Job Shop", *Journal Européen des Systèmes Automatisés*, vol. 43 no. 3/4, 2009.

Ramamurthi V., Kuhl M. E., Hirschman K. D., "Analysis of production control methods for semiconductor research and development fabs using simulation", *37° Winter Simulation Conference,* Orlando, California, 2005, p. 2177-2185.

Ricciardi V., Pujo P., Frydman C., "DEVS modelling for the proactive control by simulation of kanban production lines", *The International Workshop on Modeling & Applied Simulation* (*MAS 2003*), Italy, 2-4 Octobre 2003.

Rose O., "General simulation applications in semiconductor manufacturing: why do simple wafer fab models fail in certain scenarios?", *32° Winter Simulation Conference,* Orlando, Florida, 2000, p. 1481-1490.

Song H.S., Kim T.G., "The DEVS framework for discrete event systems control', *5th Conference on AI, Simulation and Planning in High Autonomous Systems*, Gainesville, USA, 1994, p. 228-234.

Workflow Management Coalition, *Terminology & Glossary.* WfMC-TC-1011, 3.0, Feb 1999.

Workflow Management Coalition, *Workflow Process Definition Interface — XML Process Definition Language (XPDL).* WFMC-TC-1025, Oct 2005.

Zacharewicz G., Giambiasi N., Frydman C., "Lookahead Computation in G-DEVS/HLA Environment", *Simulation News Europe Journal (SNE)*, vol. 16 no. 2, 2006, p. 15-24

Zacharewicz G., C. Frydman, and N. Giambiasi, "G-DEVS/HLA Environment for Distributed Simulations of Workflows" *Transaction of the SCS Simulation*, vol. 84 no. 5, 2008, p. 197–213.

Zacharewicz G., Chen D., Vallespir B., "Short-Lived Ontology Approach for Agent/HLA Federated Enterprise Interoperability", *IEEE Proceedings of International Conference I-ESA China*, Beijing, China, 2009 April 22-23, IEEE, p. 329-335.

Zeigler B.P., *Theory of Modelling and Simulation*, New York ,USA, Ed. John Wiley, 1976.

Zeigler B.P., Praehofer H., Kim T. G., *Theory of Modeling and Simulation*, New York, USA, 2nd Edition, Academic Press, 2000.