

MODELISATION D'UN SYSTEME DE PILOTAGE ISOARCHIQUE ET SIMULATION VIA UN ENVIRONNEMENT DISTRIBUE HLA

F. OUNNAR, P. PUJO, L. MEKAOUICHE, N. GIAMBIASI

Laboratoire des Sciences de l'Information et des Systèmes (LSIS) UMR CNRS 6168
Université Paul Cézanne - Avenue Escadrille Normandie Niemen
13397 Marseille Cedex 20 - France
Tél. : (33) 4 91 05 60 10 – Fax. : (33) 4 91 05 60 33
{ fouzia.ounnar, patrick.pujo, lynda.mekaouiche, norbert.giambiasi }@lsis.org

RESUME : *Le pilotage auto organisé de systèmes manufacturiers fait explicitement référence à une approche hétérarchique où les prises de décision décentralisées permettant l'organisation et l'exécution des tâches sont effectuées localement auprès de chaque équipement. A cette fin, nous présentons l'intégration de centres de décision locaux, les EPA (Entités de Pilotage Autonome), via une modélisation DEVS (Discret Event system Specification) incluant la description du mécanisme d'interaction avec les autres EPA, dans un système de pilotage holonique. La mise en œuvre de la simulation d'un tel système a été réalisée via un environnement HLA (High Level Architecture). Nous décrirons l'implantation de ce mécanisme d'interaction qui permet de sélectionner la meilleure réponse à un appel d'offre lancé au préalable sur le réseau de communication entre EPA. Un exemple d'application de cette approche à un réseau de partenariat logistique sera décrit.*

MOTS-CLÉS : *Pilotage auto organisé, Holonic Manufacturing System (HMS), Discret Event system Specification (DEVS), High Level Architecture (HLA), Entité de Pilotage Autonome (EPA)*

1. INTRODUCTION

Un axe fondamental de recherche en productique vise à améliorer le pilotage des organisations industrielles. Ceci peut s'envisager soit au travers de l'affinage des modes de gestion industrielle traditionnellement utilisés dans les entreprises, soit par le développement de nouvelles approches des modes de prise de décision (Pujo and Kieffer, 2002). Dans cette seconde voie, de très nombreux travaux portent sur l'étude des architectures hétérarchiques. Le travail présenté ici relève de cette voie. Le projet de recherche vise globalement à proposer une architecture de pilotage fondée sur une prise de décision isoarchique, c'est-à-dire totalement auto organisée et décentralisée (s'effectuant sans aucune considération d'ordre hiérarchique). Dans ce contexte, nous considérons un système de production comme décomposable en entités à piloter. L'absence de centre de décision hiérarchique nous amène à doter chacune de ces entités d'une capacité propre de prise de décision qui leur permet, au travers d'un système de communication, d'élaborer, par négociation et en temps réel, une solution de pilotage qui soit la plus efficace et la plus efficiente possible.

Pour ce faire, nous n'avons pas retenu l'approche multi agents, qui favorise plus une distribution du pilotage (intelligence artificielle distribuée) qu'une décentralisation : l'isoarchie est difficile à obtenir avec des SMA, car il existe toujours un agent animateur, facilitateur, coordinateur ou superviseur. Ces différentes terminologies se

traduisent toujours par un agent de niveau décisionnel supérieur, garantissant le fonctionnement du système, c'est-à-dire capable de prendre des décisions en cas de conflits entre agents ou en cas de non résolution distribuée du problème (Pujo and Ounnar, 2007). Nous avons donc opté pour une approche holonique inspirée de l'architecture de référence PROSA (Product Resource Order Staff Architecture) (Van Brussel *et al.*, 1998). PROSA autorise des structures de prise de décision isoarchiques, ce qui n'est pas le cas avec d'autres architectures holoniques, telles que ADACOR et MetaMorph. Pour décrire notre approche holonique du pilotage isoarchique, nous nous inspirons des 3 types de holons de base (*Produit, Ressource* et *Ordre*) de l'architecture PROSA, en créant des interactions entre de tels holons. Pour cela, nous associons à chaque équipement de production un centre de décision, que nous appelons EPA pour Entité de Pilotage Autonome, l'ensemble constituant un *Holon Ressource*. Chaque EPA communique avec ses homologues via un réseau de communication. Ensemble, elles assurent une répartition des tâches, sur la base d'une évaluation locale de la performance potentielle de leur *Holon Ressource* associée et d'une négociation collective pour l'affectation optimale de ces tâches. Notre EPA a fait l'objet d'une modélisation selon le formalisme DEVS (*Discret Event system Specification*) (Zeigler, 1976), (Zeigler, 2000a). Ce formalisme DEVS permet la spécification de modèles modulaires, hiérarchiques et

formels. Il est possible de procéder à des vérifications formelles des modèles DEVS, ce qui est une aide précieuse lors de la conception des modèles et de leur validation. Les différents aspects de cette modélisation d'une EPA ont déjà été largement diffusés dans diverses communications. L'objectif n'étant pas ici de présenter à nouveau de manière détaillée ces modèles, nous rappellerons brièvement les fondements de notre approche holonique.

Dans le but de valider cette proposition, il reste toutefois nécessaire de faire fonctionner le modèle global d'un réseau d'EPA. L'émergence en temps réel de la solution de pilotage doit être simulée sur différents postes de travail et via un réseau de communication réel. La performance obtenue doit pouvoir être analysée et comparée pour être évaluée. Nous nous limiterons ici à présenter le modèle global de simulation du réseau permettant de mettre en œuvre des expérimentations : une expérimentation par simulation se traduira par le pilotage d'un ensemble de tâches enchaînées au travers d'un système de simulation à architecture distribuée. Pour cela, nous avons intégré les modèles DEVS des EPA dans l'environnement de simulation distribué HLA. Après un rappel des principes de base de HLA, nous présentons l'intégration de ces différents modèles DEVS de comportement de l'EPA dans l'environnement HLA. Nous donnerons *in fine* un aperçu de la mise en œuvre de cette approche au travers de la réalisation d'une maquette de simulation d'un réseau de partenariat logistique dont les concepts avaient été définis dans (Ounnar and Pujo, 2005).

2. MODELISATION DEVS D'UNE APPROCHE HOLONIQUE DU PILOTAGE ISOARCHIQUE

Dans un système de pilotage isoarchique, chaque entité participe à l'achèvement du but commun en assumant l'organisation et le management de son propre pilotage (Ounnar and Pujo, 2001). Pour cela, elle prend part à l'élaboration et à l'évaluation de solutions, en l'absence de hiérarchie.

2.1. Architecture holonique d'un système de pilotage isoarchique

Le principe général de l'affectation de tâches est basé sur un mécanisme de sélection de la meilleure réponse à un appel d'offre diffusé auprès de toutes les entités pour chaque tâche à effectuer (Ounnar et al., 2004). De tels fonctionnements sont proposés par le courant de pensée scientifique du holonic manufacturing (Deen, 2003). Dans notre proposition, le pilotage isoarchique se caractérise plus précisément par une architecture organisationnelle du type 'structure holonique horizontale' (*flat holonic form*) (Bongaerts et al., 2000). Ce concept d'isoarchie est fondé sur les notions de prise de décision décentralisée (c'est-à-dire sans décision de niveau hiérarchique supérieur) et d'auto organisation, laquelle prend en compte le comportement propre de chaque entité. L'auto organisation du pilotage

est alors un mode de prise de décision en temps réel sans aucune organisation prévisionnelle préétablie.

Notre structure holonique horizontale est organisée autour du concept d'EPA. Du modèle PROSA, nous retenons les concepts typologiques, mais nous prôtons une autre utilisation des 'holons de base'. Dans notre approche, une EPA est un centre de décision associé à chaque entité de production. Cette EPA confère au *Holon Ressource* sa capacité d'auto organisation au sein d'un ensemble d'entités ayant les mêmes capacités : nous nous situons dans un contexte d'architecture décentralisée où les entités ne sont pas pilotées par une entité de pilotage de niveau décisionnel hiérarchiquement supérieur. Face à une tâche dont l'exécution est proposée à un ensemble de telles entités de production, chaque EPA associée possède la capacité d'auto-évaluer la performance qu'elle peut lui associer dans le but de participer aux négociations d'affectation de cette tâche.

Cette architecture donne un rôle fondamental aux EPA : elles gèrent la totalité des échanges d'information dans le réseau reliant les différentes entités et organisent l'enchaînement des traitements de ces informations qui aboutissent aux différentes prises de décision. La figure 1 illustre les échanges entre les entités partenaires passant par les EPA et les relations entre les partenaires via un réseau de communication.

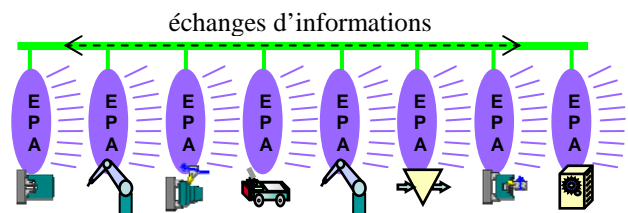


Figure 1. les Entités de Pilotage Autonome dans une structure holonique horizontale.

En fait, les EPA sont au cœur des relations entre les holons de base du modèle PROSA :

- les *Holons Ressource*, qui sont associés aux entités du système de production ; chacun possède une EPA qui lui est propre et qui auto assure le pilotage de l'équipement de production associé,
- les *Holons Ordre*, qui circulent dans le système de production, représentent l'aspect organisationnel des tâches à effectuer sur les produits manufacturés par les ressources ; cela inclut les notions de lots, de date de livraison,
- les *Holons Produit*, qui sont associés à chaque produit manufacturé, en supportent un descriptif technique (modèles, gamme...) et un descriptif de l'état d'encours ; le premier vient compléter la spécification des tâches de production, et le second permet d'individualiser toutes les informations propres à la tracabilité.

Au cours de leur parcours dans le système de production, les *Holons Produit* suivent donc les produits auxquels ils sont associés. La progression sur ce parcours est gérée

par l'évolution des *Holons Ordre*, qui déclenchent les tâches de production.

Toutes les informations nécessaires à l'évaluation de performances par l'EPA se trouvent dans ces holons. Chaque EPA a des échanges privilégiés avec son *Holon Ressource*, lequel fournit par exemple des informations sur son planning prévisionnel, sa capacité... Par ailleurs, les données liées aux *Holons Produit* et aux *Holons Ordre* circulent en deux temps. D'abord, de manière temporaire, elles sont diffusées sur le réseau via les Appels d'Offre (AO), pour que puisse s'établir l'évaluation de performance comparative dans les différentes EPA. Ensuite, lorsque le choix est effectué, elles sont transférées de façon définitive en même temps que les entités physiques pour ce qui concerne les *Holons Produit* ou les entités informationnelles pour les *Holons Ordre*. Les données sont alors hébergées dans les EPA correspondant à la ressource traitant le produit et/ou l'ordre. En résumé, les EPA pilotent les *Holons Ressource* en assurant l'affectation des *Holons Ordre* qui concernent la production des *Holons Produit*.

Une EPA qui a obtenu la responsabilité de l'exécution d'une étape *i* de la transformation d'un produit a au préalable évalué la faisabilité de la tâche correspondante sur le *Holon Ressource* qui lui est associé. Pour cela, elle a reçu comme les autres EPA un AO_i émis par l'EPA en charge de l'étape *i-1*. Cet AO_i apporte, grâce au *Holon Produit*, des informations relatives à l'état initial du produit à l'entrée de l'étape *i* : il s'agit de la somme cumulée de toutes les transformations subies par le produit jusqu'à l'issue de l'étape précédente *i-1*. Il contient également la description de l'état final demandé à l'issue de l'étape *i* (la différence représentant la valeur ajoutée de l'étape *i*, ie. le contenu de la tâche *i* correspondante). Il contient également des éléments techniques sur la manière d'amener cette valeur ajoutée (éléments de la gamme, par exemple). L' AO_i est également porteur d'informations liées au *Holon Ordre* : taille du lot de produits, priorité, date de mise à disposition et date limite de livraison des produits... L'analyse de toutes ces données, utilisées pour générer une tâche compatible avec les capacités de la ressource, associée à la gestion du planning d'occupation de cette dernière, permet l'évaluation de la faisabilité de la tâche sur la ressource en question.

Si cette performance se révèle être meilleure que celles proposées par d'autres EPA, nous avons trouvé potentiellement une meilleure ressource pour l'exécution de la tâche *i*. Au bout d'un certain temps, défini en fonction de la dynamique de l'application, l' AO_i est clôturé.

Il y a alors affectation de l' AO_i : la meilleure EPA à cet instant prend en charge, par un pilotage direct de sa ressource, l'exécution de la tâche *i* et veille à l'obtention du résultat final de l'étape *i*.

Parallèlement, l'EPA doit veiller à la préparation de la suite des événements, c'est-à-dire d'initier le choix de l'EPA en charge des opérations à effectuer sur le produit pour l'étape *i+1*. Pour cela, elle émet un AO_{i+1} qui contient les éléments éventuellement actualisés du *Holon Ordre* et du *Holon Produit* (voir figure 2). Ce principe

permet de poursuivre pas à pas le processus de transformation du produit. Eventuellement, s'il survient un incident durant l'exécution de la tâche *i*, elle peut relancer un AO_i pour trouver une ressource de substitution capable de terminer l'exécution de la tâche *i*.

2.2. Composition interne d'une EPA

Un certain nombre de primitives fonctionnelles sont communes à toutes les EPA, quelque soit l'application. D'une manière générique, nous pouvons identifier sept modules principaux composant une EPA : les modules Communication, Interaction, Optimisation, Planning, Décision, Action et Perception (figure 2).

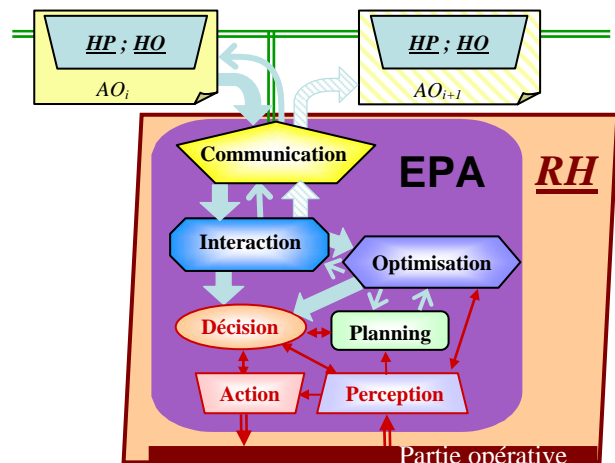


Figure 2. Relations entre les holons de base et les modules d'une Entité de Pilotage Autonome.

Après un rapide survol du rôle respectif de ces modules, nous nous concentrerons sur les trois modules qui permettent le fonctionnement global du pilotage auto organisé et décentralisé.

2.2.1 Description des modules d'une EPA

Chaque EPA est donc composée de sept modules.

Le module Communication assure le rôle de lien avec les autres EPA. Il traite tous les aspects d'ordre technologique que sous tend l'usage d'un réseau industriel de communication : mise en forme des messages, transport des données et surveillance de la bonne transmission des messages... Dans le cas de notre maquette de faisabilité, c'est l'environnement de simulation distribué qui prend en charge directement ces aspects. Le module Interaction gère le dialogue avec les autres EPA. Il va analyser les messages reçus et déclencher les traitements pour pouvoir répondre de la meilleure manière. Pour cela, le module Optimisation élabore la meilleure solution possible, en prenant en considération les caractéristiques techniques de la ressource (qui limitent sa performance) et ses disponibilités. Ces dernières sont fournies par le module Planning, qui gère l'emploi du temps de la ressource.

Les trois derniers modules assurent le pilotage local de la ressource. Dans le cas d'un réseau de partenariat logistique, les ressources sont des entreprises.

Globalement, ces trois modules en assurent la gestion interne. Le module Décision couvre les fonctions de gestion prévisionnelle et génère les programmes de production que le module Action doit exécuter. Ce dernier est donc équivalent à la fonction 'lancement' d'un MES (Manufacturing Execution System). Le module Perception correspond quant à lui à la fonction 'suivi', et intègre l'acquisition de données techniques de production (ce qui permet le bouclage tâche via le module Action) et l'agrégation de ces données à destination des modules Décision, pour la surveillance de la tâche, et Optimisation, pour la capitalisation et la validation des traitements de génération de tâche. Dans notre cas, nous avons fait l'hypothèse d'un fonctionnement idéalisé de la ressource vis-à-vis de l'analyse faite par le module Optimisation. Cette hypothèse est réaliste dans le cadre d'une simulation visant à monter le bon fonctionnement du mécanisme d'interaction entre EPA.

Nous allons donc maintenant approfondir les modules Interaction, Optimisation et Planning.

2.2.2 Module Interaction

Le module Interaction permet d'assurer l'affectation des tâches par la mise en concurrence de ressources pouvant potentiellement répondre à un appel d'offre. Pour que le résultat soit cohérent, il faut mettre en œuvre un protocole de décision basé sur des règles et des critères impartiaux et communs. Ce mécanisme de décision appartient à la famille des protocoles d'interaction de la famille ContractNet (Smith, 1980) (Smith and Randall, 1983). Il s'agit d'observer sur un réseau quelle est l'entité qui fait la meilleure réponse à un appel d'offre : c'est l'entité à qui sera confiée l'exécution de la tâche correspondante. Les principales fonctionnalités du module Interaction se résument en deux points :

- la gestion de l'interaction avec les autres EPA, avec la transmission des différentes informations concernant les appels d'offre ou les réponses aux appels d'offre reçus (RAO_k) venant de l'extérieur vers le module d'Optimisation ou vice versa,
- le positionnement de chaque EPA par rapport aux offres reçues : il est procédé à une mise à jour de la meilleure performance connue pour chaque appel d'offre AO_k ; lorsque le module d'Optimisation fournit la réponse au calcul de performance, cette réponse est comparée avec la meilleure réponse connue.

En fonction de l'évolution des attributions des tâches et des RAO et RAOL (réponses locales qu'elle émet, suite aux AO reçues) qu'elle voit circuler sur le réseau, l'EPA gère également l'état des différents AO en cours de traitement :

- AO négociable : appel d'offre en cours de négociation, pour lequel éventuellement une première réponse a été reçue, ce qui constitue une performance de référence,
- AO engageable : appel d'offre qui est pour l'instant attribué à l'EPA, sa performance étant à cet instant la meilleure parmi celles diffusées sur le réseau ; ce statut peut être perdu si une meilleure offre apparaît, puis

regagné si par exemple des disponibilités se libèrent sur le planning de l'EPA,

- AO pré engagé : appel d'offre engageable et qui est le prochain à être traité sur le planning des tâches de l'EPA (la date de fin de négociation est atteinte),
- AO engagé : appel d'offre auparavant pré engagé, dont la mise en œuvre est en cours ou sur le point de l'être, et qui à ce titre, est placé de façon ferme et définitive sur le planning de l'entité à la date d'engagement de celle-ci. L'AO devient alors géré par le module Décision.

Nous avons modélisé ce module à l'aide d'un modèle DEVS couplé composé de trois modèles atomiques :

- le premier modèle atomique gère la réception et la diffusion des AO et des AOL (AO lancés par l'EPA),
- le second traite les réponses reçues RAO et les réponses fournies par le module Optimisation RAOE,
- le troisième modèle atomique mémorise toutes les performances RAOL reçues en réponse aux AO émis localement par l'EPA.

Ces modèles sont totalement détaillés dans (Mekaouche *et al.* 2005a).

2.2.3 Module Optimisation

Le module Optimisation permet à l'EPA d'auto évaluer la capacité de la ressource à effectuer la tâche décrite au travers de l'appel d'offre reçu : il s'agit d'estimer sa propre performance pour pouvoir répondre. Cette évaluation de la performance est fondée sur quelques principes intangibles qui rendent de manière fiable le résultat robuste :

- toutes les EPA en compétition pour répondre à un appel d'offre utilisent un mode de calcul identique pour obtenir leur évaluation de performance ; ce qui fait la différence entre les résultats sont les différences de caractéristiques des ressources et leur état courant à l'instant du calcul ; il y a donc une totale transparence sur le mode d'obtention des résultats,
- les algorithmes d'évaluation de cette performance doivent être basés sur des modèles aussi proches que possible du comportement réel de la ressource ; pour cela, une analyse 'métier' de la ressource doit être possible : pour un type de ressource donné, une modélisation paramétrique (c'est-à-dire dépendant des caractéristiques techniques réelles de la ressource) est utilisée avec comme données le modèle de description de la tâche à évaluer et celui de l'état actuel de la ressource. Cette performance est obtenue par une simulation de l'exécution de la tâche *i* sur la ressource. Ceci implique une relation très forte entre les modules Optimisation et Planning de l'EPA, les données caractérisant la ressource associée au *Holon Ressource* et celles contenues dans le *Holon Ordre* et le *Holon Produit*.

Grâce à ces deux conditions, l'émergence de la meilleure solution se fera dans des conditions d'équité et de transparence qui favoriseront la recherche de la meilleure solution.

Les méthodes pour élaborer le calcul de la performance sont intimement liées à la nature de la ressource et aux éléments permettant de juger de la pertinence de la

solution. Il s'agit de méthodes d'expertise consistant à utiliser des approches issues de l'Intelligence Artificielle (système expert), de la Recherche Opérationnelle (analyse multicritère) et/ou de la Fabrication Assistée par Ordinateur (algorithmes de génération de trajectoires machine).

Dans l'application présentée ici, les *Holons Ressource* sont des usines. On cherche alors à améliorer l'efficacité de la chaîne logistique entre ces usines au travers du pilotage des relations clients fournisseurs. Devant la difficulté pour prendre en compte tous les aspects des relations clients fournisseurs et des dysfonctionnements provenant de l'une ou l'autre des deux parties, une méthode d'analyse multicritère fournit une solution tout à fait satisfaisante. Nous avons sélectionné la méthode Analytic Hierarchy Process (AHP) (Saaty, 1980) qui permet de prendre en compte des critères qualitatifs et d'autres quantitatifs, de décomposer un problème complexe et de l'examiner à de multiples niveaux, puis de mesurer la cohérence des résultats obtenus.

Face à un appel d'offre émis par un client potentiel, le fournisseur établit via AHP un classement de tous les appels d'offre reçus, par ordre d'intérêt, en fonction de sa capacité à les traiter. L'EPA lance alors le calcul de la performance des AO auxquels il est le plus avantageux de répondre. Ceci prend en compte différents aspects : coût, qualité & délai. Par exemple, l'historique de la qualité de service de l'EPA intervient dans le calcul de la performance : un faible coût cache peut être des désagréments sur le respect des délais ou sur la qualité des produits.

Les modèles DEVS couplés et DEVS atomiques du module Optimisation sont présentés en détail dans (Mekaouche *et al.*, 2005b).

Parmi les critères quantitatifs figure la notion de respect des délais de livraison et de maîtrise des temps d'attente intermédiaire. C'est une notion qui intervient quel que soit le mode de fonctionnement du module d'Optimisation. Elle est prise en charge par le module Planning.

2.2.4 Module Planning

Le module planning gère le planning de production du *Holon Ressource* dans le but de tester la possibilité d'insérer de nouveaux AO dans ce planning. Il gère donc l'insertion des nouvelles tâches évaluées en fonction de l'état des AO déjà placés sur le planning :

- AO négocié : tâche à placer sur le planning,
- AO engageable : espace temps momentanément réservé sur le planning par une tâche susceptible d'être affectée à une autre ressource,
- AO pré-engagé : espace temps dédié à la réalisation de la prochaine tâche de la ressource,
- AO engagé : tâche en cours d'exécution.

Le module planning permet de calculer ou de vérifier la date de réalisation d'un appel d'offre. Il peut se contenter de vérifier les conditions d'insertion de nouvelles tâches parmi les tâches déjà figées. Il peut également contenir une heuristique de placement de tâches et

d'ordonnement qui optimise localement la productivité de la ressource.

L'approche proposée n'exclue en rien la contribution de l'acheteur ou du logisticien, bien au contraire. En effet, l'intervention de l'homme se situe à plusieurs niveaux :

- en amont, il faut choisir les critères et indicateurs qui sont utilisés dans l'évaluation de la performance des fournisseurs ; ce choix est établi via le partenariat.
- chaque fournisseur peut définir ses propres préférences sur les critères et les indicateurs qui sont pris en compte dans le calcul de sa performance, pour sélectionner les AOs sur lesquels il souhaite entrer en négociation.
- toutes les décisions sont prises avec l'assentiment d'un 'pilote' humain, qui supervise le bon déroulement des relations inter entreprises ; chaque client garde la possibilité d'indiquer qu'il préfère travailler avec le fournisseur x : il sait alors la perte de productivité qu'une telle décision engendre.

Cette approche permet donc d'assurer la distribution des ordres par la mise en concurrence de fournisseurs pouvant potentiellement répondre à un appel d'offre émis par un client, avec des règles et des critères impartiaux et communs à tous, tout en laissant à chacun des degrés de liberté. Il s'agit essentiellement d'une assistance automatique pour le pilotage de chaque partenaire.

3. HIGH LEVEL ARCHITECTURE

HLA est un environnement qui a été développé par le Defense Modeling and Simulation Office (DMSO) du département de la défense américain (Department of Defense (DoD)). Il permet de répondre à des besoins en matière de simulation militaires de situations d'engagement opérationnel de troupes. Toutefois, son utilisation s'est aujourd'hui étendue à d'autres domaines d'application, comme la logistique ou la simulation de chaînes de production (Zeigler, 2000b), (Pujo *et al.*, 2005).

3.1. Définitions

Une **Fédération** HLA désigne un système de simulation distribué permettant à un ensemble de simulations élémentaires appelées **fédérés** de s'échanger des informations. Un fédéré HLA n'est pas obligatoirement une simulation : ce terme désigne tout participant matériel ou logiciel d'une Fédération. HLA présente en effet une capacité d'interopérabilité de la simulation, ce qui permet d'associer d'une part des composants du monde réel (des entreprises existantes) et d'autre part des modèles de simulation (des entreprises virtuelles).

HLA est formellement définie par trois composants : les règles HLA, les spécifications d'interface et l'Object Model Template.

Les **règles HLA** présentent les principes de HLA (McLéod, 2005). Au nombre de dix, elles régissent pour moitié le fonctionnement des fédérés et pour moitié celui des Fédérations (IEEE P1516).

Les **spécifications d'interface** de HLA décrivent les services d'exécution fournis au fédéré par l'infrastructure d'exécution *Run Time Infrastructure* (RTI). Ce RTI est constitué d'un ensemble de processus informatiques écrit dans un langage de programmation donné. Il peut être assimilé à un noyau réduit de système d'exploitation distribué, dont le rôle est le management des circulations d'informations entre fédérés. Ces spécifications indiquent comment les fédérés interagissent pendant l'exécution de la Fédération (IEEE P1516.1).

Le RTI constitue une implémentation informatique des spécifications d'interface de HLA. Il fournit un ensemble général de 6 catégories de services qui supportent la simulation, qui assurent l'interaction entre les fédérés ainsi que la gestion de la Fédération au travers d'un réseau local ou longue distance.

Par exemple, une famille de service gravite autour de la gestion du temps, avec l'avancement du temps logique (ou temps simulé) d'une manière coordonnées entre les fédérés, la synchronisation du temps d'exécution pour l'échange de données de simulation.

L'**Object Model Template** (OMT) est la description des éléments (objets et interactions) qui sont partagés à travers une Fédération (IEEE P1516.2). HLA exige que chaque fédéré et Fédération doit documenter son modèle d'objet en utilisant l'OMT. Ces modèles sont prévus pour faciliter la réutilisation des simulations. En effet, le modèle objet de HLA ne se préoccupe pas des données internes aux fédérés, qui sont utilisées pour l'implémentation du modèle comportemental des entités simulées.

Les classes ne sont pas décrites par leur comportement, c'est-à-dire qu'il n'existe pas de méthodes décrivant ces classes. Ce point est conforme avec l'esprit HLA qui considère que le comportement des entités simulées par un fédéré doit être masqué au sein du fédéré et ne doit en aucun cas apparaître au niveau de la communication.

Pour cela, HLA spécifie deux types de modèles d'objets. Le premier est le **Modèle d'Objet de Fédération (FOM)** qui décrit l'ensemble des objets, attributs et interactions qui sont partagés à travers la Fédération. Le deuxième est le **Modèle d'Objet de Simulation (SOM)**, qui fournit des informations sur la faculté de simulation d'échange de l'information lorsqu'elle participe à une Fédération. Il existe un autre composant documenté selon l'OMT, qui est le **MOM (Management Object Model)**. Ce modèle d'objet peut être utilisé par les fédérés pour obtenir des informations sur la Fédération à laquelle ils participent.

3.2. Composants d'une exécution d'une Fédération HLA

La figure 3 montre l'architecture d'une simulation distribuée dans une Fédération HLA. Tous les composants représentés dans la figure font partie de la Fédération à l'exception du RTI. Ne sont pas représentés :

- **FedExec** (exécutif de Fédération) : c'est le processus qui gère une Fédération. La création de ce processus est réalisée par le premier fédéré qui joint avec succès la Fédération. Il permet aux autres fédérés de rejoindre et de démissionner de la Fédération pendant

l'exécution et facilite l'échange de données entre les fédérés participants.

- **Fichier FDD (FOM Document Data File)** : ce fichier contient l'information du FOM utile au RTI pendant l'exécution.
- **RID File (RTI Initialisation Data)** : ce fichier d'initialisation du RTI contient les informations spécifiques données par le concepteur du RTI et nécessaires pour diriger le RTI.
- **RtiExec** (exécutif RTI) : c'est un processus global qui gère la création et la destruction des FedExec.
- **LibRTI** : il s'agit des bibliothèques contenant les méthodes correspondant aux services HLA utilisables par les fédérés. Les fédérés emploient libRTI pour invoquer les services HLA.

4. INTEGRATION DES MODELES DEVS-EPA DANS UN ENVIRONNEMENT HLA

4.1. Relations entre les différents composants de simulation

Dans le but d'intégrer les modèles DEVS des EPA dans l'environnement HLA, nous avons décomposé le fédéré EPA en :

- Simulateurs des modèles DEVS-EPA ;
- Coordinateur local ;
- FOM ;
- Interface entre les simulateurs des modèles et le RTI.

La figure 3 montre les différentes relations entre les simulateurs des modèles DEVS-EPA qui, dans notre cas sont les simulateurs associés aux différents modèles atomiques qui forment le modèle couplé DEVS-EPA, et le coordinateur local ainsi que l'interface entre les simulateurs et le RTI.

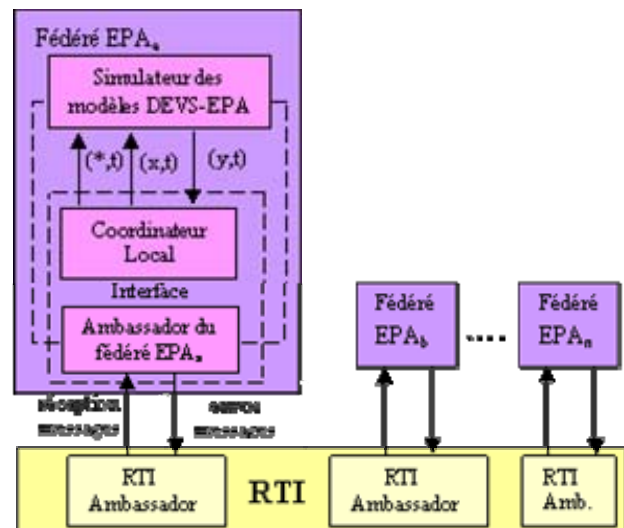


Figure 3. Relations entre les composants de simulation

4.1.1 Simulateurs des modèles DEVS-EPA

Ils assurent la simulation des modèles atomiques en analysant leur comportement selon le formalisme DEVS : calcul des changements d'états et éventuellement des

événements de sortie résultants de la réception des événements internes. Pour représenter nos modèles DEVS-EPA, nous avons utilisé deux classes abstraites, une pour les modèles DEVS atomique 'ModelAtomic', à partir de laquelle héritent les modèles atomiques DEVS-EPA, dans laquelle nous trouvons toutes les méthodes définies en DEVS. La deuxième classe 'ModelCouple' concernant les modèles DEVS couplés, à partir de laquelle héritent les modèles couplés DEVS-EPA. Les différentes méthodes de cette classe sont appelées à partir du coordinateur local afin de sélectionner le modèle atomique influencé par un tel ou tel événement.

4.1.2 Coordinateur local

Le coordinateur local manipule un échéancier contenant des Xmessages (événements d'entrées) et des *messages déduits des durées de vie de ses successeurs (des simulateurs atomiques des différents modèles atomiques). Le coordinateur local s'occupe également de la gestion de l'horloge locale (simulation locale), en sélectionnant le prochain message chronologique de son échéancier par rapport à sa date actuelle, et en le transmettant au simulateur successeur (la destination). Il comporte également une liste d'attente dans laquelle il stocke les messages en cours de traitement (au niveau des successeurs) et une liste d'acquiescement qui permet de vérifier que le traitement des messages envoyés est bien terminé. Il conserve aussi les relations de couplage avec ses successeurs dans d'autres listes.

4.1.3 FOM

Les données communes des SOM des différents fédérés permettent de produire le FOM de la fédération DEVS/HLA. Elles sont de deux types : les objets et les interactions. Les objets sont des informations partagées persistantes contrairement aux interactions qui sont des données éphémères (simplement émises et reçues).

4.1.4 Interface entre les simulateurs des modèles d'EPA et le RTI

La Figure 4 montre que le RTI est en relation avec les simulateurs des modèles par le biais du noyau de simulation. En effet, le rôle du noyau de simulation est de diriger la simulation. En d'autres termes, il permet la sélection du prochain événement chronologique de son échéancier, pour ensuite procéder à son traitement. Pour cela, il fait appel aux différentes méthodes du coordinateur local, afin de sélectionner la destination (prochain modèle à activer en fonction de l'événement). Mais dans le cas d'une simulation distribuée (dans notre cas des simulateurs de modèles DEVS-EPA qui interagissent entre eux via le RTI), le couple (Coordinateur Local – Noyau de Simulation) devra aussi assurer la gestion des messages provenant des autres fédérés. En effet, il reçoit des messages provenant du RTI via le code héritant de la classe FederateAmbassador, il invoque les méthodes du RTIAmbassador pour interroger le RTI. Enfin il est à l'écoute de tous les événements produits par la partie interface utilisateur. Le couple (Coordinateur Local – Noyau de Simulation) peut donc simuler localement un

modèle couplé de façon autonome. Pour cela, chaque modèle DEVS couplé et son Coordinateur Local – Noyau de Simulation associé, doivent avoir un SOM définissant les informations qu'ils sont capables de fournir dans une simulation distribuée respectant HLA. Nous allons détailler rapidement l'ensemble des éléments cités dans cette figure, éléments implémentés pour obtenir la Fédération d'EPA simulant le réseau de partenariat logistique.

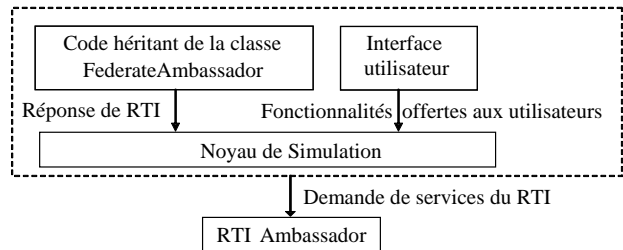


Figure 4. Interface entre les simulateurs et le RTI

4.1.4.1 Interface utilisateur

Cette interface permet à l'utilisateur de rejoindre le réseau d'entreprises, de proposer des AOLS, d'initialiser son planning, etc. Quelques fonctionnalités nécessitent l'invocation des services du RTI en employant les méthodes de la classe RTIAmbassador.

Le premier fédéré EPA créé, agissant comme un manager, crée une exécution de Fédération en invoquant une méthode particulière de son *RTIAmbassador*. Cela signifie que la Fédération qui représente le réseau d'entreprises est lancée. Une fois qu'une exécution de Fédération existe, d'autres fédérés EPA peuvent la rejoindre en invoquant la méthode *joinFederationExecution* du *RTIAmbassador* : il devient possible de construire le réseau de partenariat logistique.

Puisque les fédérés EPA proposent tous des AOs et des réponses à des AOs et qu'ils reçoivent des AOs et des réponses à des AOs, ils doivent publier et souscrire à ces classes d'interaction en utilisant le service de déclaration de la spécification d'interface. De même, les fédérés doivent souscrire à la classe d'objet qui identifie chaque entreprise (EPA) en utilisant les services d'enregistrement et de découverte de la gestion d'objet pour qu'ils soient clairement identifiés comme membres de la Fédération. Un fédéré peut démissionner de sa Fédération toujours par invocation des méthodes de son *RTIAmbassador*. Le dernier fédéré qui démissionne de la Fédération met fin à la Fédération.

4.1.4.2 Code héritant de la classe FederateAmbassador

Les fédérés sont des objets héritant de la classe *FederateAmbassador*. Cette classe définit toutes les interfaces pour les fonctions de rappel de service, mais ne peuvent pas être employées avant que l'on ne fournisse les mises en œuvre réelles de ces fonctions dans le code du fédéré. Ces fonctions permettent la réception des messages et des réponses produites par le RTI. Par exemple, pour que le fédéré puisse recevoir des interactions, il doit implémenter la méthode 'ReceiveInteraction', appelée par le RTI quand il détecte que ce fédéré a souscrit à une telle interaction et que d'autres fédérés ont envoyé une instan-

ce de cette classe d'interaction. C'est grâce à cette méthode qu'un fédéré peut recevoir des événements.

4.1.4.3 Noyau de simulation

Il permet la sélection du prochain évènement chronologique de son échéancier, pour ensuite procéder à son traitement. Le noyau doit également s'assurer de la gestion des messages provenant des autres fédérés. Nous présentons succinctement ci après les principes de gestion du temps sur lesquelles nous nous sommes basés pour l'intégration des modèles DEVS-EPA dans un environnement HLA.

4.1.4.4 Principe de gestion de temps

Les services de gestion du temps permettent à tout fédéré EPA de contrôler son avance dans le temps logique, vis-à-vis de celle des horloges logiques des autres fédérés EPA participant à la Fédération (Zacharewicz *et al.*, 2006). Les mécanismes de gestion du temps assurent notamment une avance dans le temps qui peut respecter les principes de causalité des événements. Le premier fondement de HLA est qu'il n'existe aucune référence à une horloge globale de la Fédération. Par contre, chaque fédéré EPA doit avoir son propre temps logique.

L'architecture HLA supporte trois mécanismes d'avance dans le temps des fédérés participant à une Fédération :

- en temps coordonné, lorsque l'avance dans le temps de chaque fédéré EPA est coordonnée avec celle des autres fédérés EPA,
- sur événement, lorsque l'avance dans le temps d'un fédéré est séquencée par la répercussion des événements (dans ce cas, l'avance dans le temps du fédéré est subordonnée aux dates associées aux événements reçus),

- enfin, de manière optimiste, lorsque chaque fédéré avance dans le temps à sa propre vitesse, indépendamment de celles des autres fédérés. Dans ce cas, l'occurrence d'événements dans le passé d'un fédéré provoque des mécanismes de retour en arrière "rollback" afin de revenir à un état de la simulation cohérent avec la date de traitement de l'événement.

4.2. Exemple de Fédération DEVS/HLA :

Le travail décrit ci-dessus a été implémenté sous forme d'une maquette en utilisant le langage java. Nous avons implémenté un fédéré EPA, dupliqué en autant de partenaires que n'en compte le réseau logistique. Une simulation permet d'illustrer, d'une part le comportement interne d'une EPA, et d'autre part son comportement avec les autres EPA (illustration de l'auto organisation).

Considérons un réseau composé d'un client et de 4 fournisseurs. A partir de l'interface (cf. figure 5), un partenaire peut devenir un membre du réseau d'entreprises. Ce partenaire se place en qualité de Client ou de Fournisseur. Le client émet un appel d'offre sur le réseau comportant des données relevant du *Holon Produit [PH]* et du *Holon Ordre [OH]* :

- référence du produit [PH],
- type de travail réalisé (savoir faire métier) [PH],
- durée théorique de la tâche (ici, 55 min) [PH]
- délai maximum souhaité (ici, 10 jours) [OH],
- quantité du lot (ici, 22 pièces) [OH],
- date de fin de négociation (ici, 105 min) [OH],
- date de mise à disponibilité (ici, le 07/08) [OH].

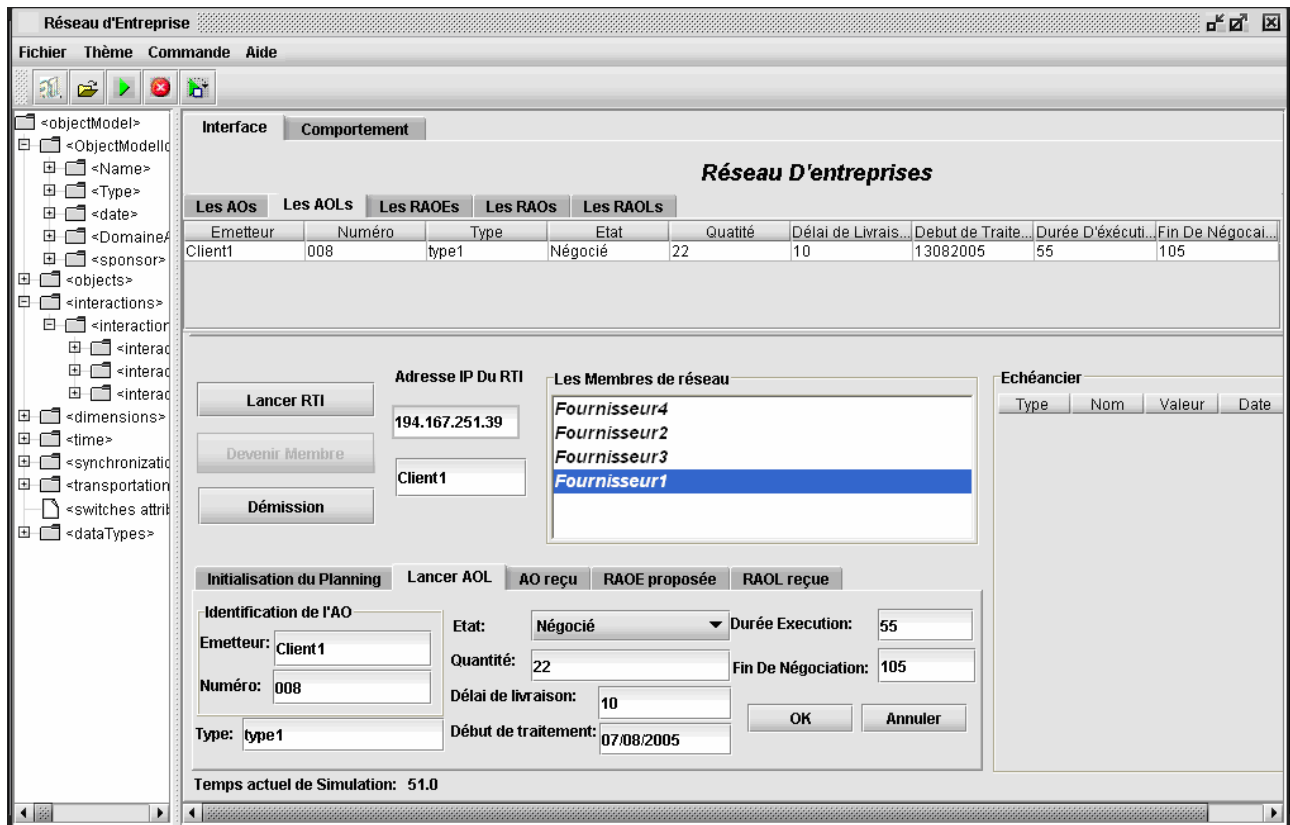


Figure 5. Interface H/M d'un fédéré EPA

deux types de perturbations : test de la limite de saturation - test de la conjonction d'accumulation de charge. Les résultats obtenus lors du pilotage auto-organisé montrent un équilibre et un lissage de charge au niveau de toutes les entreprises du réseau. L'exemple complet ainsi que les détails de l'analyse des performances sont disponibles dans (Mekaouche, 2007).

Dans le but de continuer cette validation, nous désirons appliquer maintenant ce principe de pilotage auto organisé dans le cadre d'un réseau partenarial logistique réel. Dans un premier temps, nous utiliserons les capacités d'interopérabilité de HLA pour obtenir un réseau partenarial logistique composé pour partie d'entreprises réelles et pour partie d'entreprises simulées, avant de basculer sur un pilotage auto organisé appliqué à l'ensemble des entreprises partenaires.

D'autres applications sont également en cours de développement en vue de la validation de l'approche sur un large éventail de situations industrielles de pilotage.

REFERENCES

- Bongaerts, L., L. Monostori, D. McFarlane, and B. Kadar, 2000. Hierarchy in distributed shop floor control, *Computers In Industry*, Vol 43, p. 123-137
- Deen, S.M., 2003. *Agent-Based Manufacturing – Advances in the Holonic Approach*. Springer-Verlag Ed, ISBN 3-540-44069-0.
- IEEE P1516, Draft Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules.
- IEEE P1516.1, Draft Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Federate Interface Specification.
- IEEE P1516.2, Draft Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Object Model Template (OMT) Specification.
- McLeod, 2005. HLA Courses. McLeod Institute of Simulation Science (MISS), <http://www.ecst.csuchico.edu/~hla/courses.html>
- Mekaouche, L., 2007. *Pilotage holonique auto-organisé de réseaux logistiques : Validation par modélisation et simulation distribuée*. Thèse de doctorat, Université Paul Cézanne Aix-Marseille III.
- Mekaouche, L., F. Ounnar, P. Pujo and N. Giambiasi, 2005a. Customers–Suppliers Relationship Self Organized Control Modeling using DEVS formalism. *Proceedings of the 17th IMACS World Congress Scientific Computation, Applied Mathematics and Simulation*, ISBN: 2-915913-02-1 – Paris, France.
- Mekaouche, L., F. Ounnar, P. Pujo and N. Giambiasi, 2005b. Management of calls for proposals within self organized enterprises network. *Proceedings of the 21st Industrial Marketing and Purchasing (IMP) Conference - Rotterdam*.
- Ounnar, F., L. Mekaouche, P. Pujo and N. Giambiasi, 2004. Decentralized Self Organized Control of a Partnership Network in an Intelligent Supply Chain. *Proceedings of the International IMS Forum 2004*, p. 913-1326.
- Ounnar, F., et P. Pujo, 2001. Décentralisation des mécanismes de pilotage de la relation donneurs d'ordres / fournisseurs. *Proceedings of the 4e Congrès International de Génie Industriel, Aix-en-Provence*, p. 1175-1185.
- Ounnar, F., and P. Pujo, 2005. Supplier evaluation process within a self-organized logistical network. *International Journal of Logistics Management*, 16:1, p. 159-172.
- Pujo, P., et J.P. Kieffer, 2002. Concepts fondamentaux du pilotage des systèmes de production, in: P. Pujo and J.P. Kieffer, ed., *Fondements du pilotage des systèmes de production*, Edition Hermès, *Traité IC2 Productique*, Paris. P. 25-50.
- Pujo, P., F. Ounnar and C. Zanni, 2005. A self organized holonic control for mechatronics complex systems: application to a robotized car park. *International Journal of Automation in Austria*, p 117-128.
- Pujo, P., et F. Ounnar, 2007. Vers une approche holonique des systèmes mécatroniques complexes - Proposition d'un système de pilotage auto-organisé et isoarchique. *JESA*, 41(6), p. 673-706.
- Saaty, T., 1980. *The analytic hierarchy process*. Mc Hill.
- Smith, R. G., 1980. The contract net protocol. High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers* 29(12), p. 1104-1113.
- Smith, R. G., and D. Randall, 1983. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence* 20, p. 63–109.
- Van Brussel, H. J. Wyns, P. Valckeneers, L. Bongaerts and P. Peeters, 1998. Reference architecture for holonic manufacturing systems: PROSA. *Computers in Industry*, 37 (3), p. 255-276
- Zacharewicz G., N. Giambiasi and C. Frydman, 2006. Lookahead Computation in G-DEVS/HLA Environment, in: *Simulation News Europe Journal (SNE) special issue 1 "Parallel and Distributed Simulation Methods and Environments"*, vol. 16, n° 2, p. 15-24, September 2006. ISSN, 0929-2268.
- Zeigler, B., 1976. *Theory of Modelling and Simulation* Ed. John Wiley, New York.
- Zeigler, B., H. Praehofer and T.G. Kim, 2000a. *Theory of modeling and Simulation*. 2nd Edition, Academic Press, New York.
- Zeigler, B., G. Ball, H. Cho, J.S. Lee and H. Sargoughian, 2000b. Implementation of the DEVS Formalism over the HLA/RTI: Problems and Solutions. *Proceedings of the 32° Winter Simulation Conference*, Orlando, Florida.