



**HAL**  
open science

# Vers une approche holonique des Systèmes Mécatroniques Complexes : proposition d'un système de pilotage auto organisé et isoarchique

Patrick Pujo, Fouzia Ounnar

## ► To cite this version:

Patrick Pujo, Fouzia Ounnar. Vers une approche holonique des Systèmes Mécatroniques Complexes : proposition d'un système de pilotage auto organisé et isoarchique. *Journal Européen des Systèmes Automatisés (JESA)*, 2007, 41 (6), pp.673-706. hal-01212614

**HAL Id: hal-01212614**

**<https://hal.science/hal-01212614>**

Submitted on 3 Nov 2015

**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.

---

# Vers une approche holonique des systèmes mécatroniques complexes

## Proposition d'un système de pilotage auto-organisé et isoarchique

Patrick Pujo — Fouzia Ounnar

---

*RÉSUMÉ.* Nous présentons dans ce papier les conditions pour lesquelles un système artificiel constitué d'entités mécatroniques peut être lui-même qualifié de mécatronique. Nous montrons que c'est le cas lorsque l'intelligence conférée aux dites entités leur permet de faire fonctionner ce système sans l'intervention d'un centre de pilotage de niveau hiérarchique supérieur. Le principe général de l'affectation de tâches est basé sur un mécanisme de sélection de la meilleure réponse à un appel d'offres diffusé auprès de toutes les entités pour chaque tâche à effectuer. L'approche holonique nous fournit un support conceptuel pour proposer un système générique de pilotage auto-organisé et isoarchique, adapté aux systèmes mécatroniques complexes et fondé sur l'existence d'un pilotage autonome associé au sein de chaque entité mécatronique. Pour conclure, nous présentons un éventail de systèmes que nous pouvons considérer de par leur système de pilotage comme étant des systèmes mécatroniques complexes...

*ABSTRACT.* In this paper, we present the conditions for which an artificial system constituted by mechatronic entities can be it self considered as mechatronic. We show that it is the case when the intelligence conferred to the entities allows them to operate this system without the intervention of a control centre on a higher hierarchical level. The general principle of the tasks assignment is based on the selection of the best response to call for proposal launched all the entities for each task to be made. The holonic approach provides a conceptual support to propose a generic system of self organized and isoarchic control. This control system is adapted to the complex mechatronic systems and based on the existence of an autonomous control associated to each mechatronic entity. To conclude, we present some complex mechatronic systems that have such control system.

*MOTS-CLÉS :* pilotage auto-organisé, système mécatronique complexe, entité de pilotage autonome, holon, architecture PROSA.

*KEYWORDS:* self organized control, complex mechatronic systems, autonomous control entity, holon, architecture PROSA.

---

### 1. Introduction

Nous pouvons définir un objet mécatronique comme étant un produit industriel dont la conception résulte de l'intégration de composants élémentaires mécaniques, électroniques, informatiques et automatiques. Cette définition classique sous-entend l'association et l'intrication de ces technologies en une entité mécatronique, et cela se traduit par différents aspects :

- un produit mécatronique possède des fonctions élaborées, ce qui lui donnera de l'intérêt vis-à-vis d'un produit classique, en termes d'autonomie au sein de son environnement et de surveillance de son mode de marche,
- un produit mécatronique présente tout à la fois une augmentation de sa complexité interne, avec l'introduction de technologies numériques, et une simplification de ses conditions d'utilisation (vision externe du produit).

Cette présentation initiale n'est cependant pas suffisante, car elle ne reflète pas pleinement les sauts technologiques que cette intégration permet d'effectuer. Là où les problèmes n'étaient résolus que par des solutions relevant d'une seule technologie (par exemple, mécanismes mus par des cames, ou régulations purement analogiques), il devient possible d'imaginer de nouvelles solutions combinant l'apport conjugué de plusieurs technologies. En particulier, l'électronique numérique amène à l'ingénierie mécanique la possibilité d'intégrer à l'entité mécatronique une commande et un pilotage 'intelligents' Ceci ouvre des horizons à explorer, et de nombreuses recherches actuelles peuvent s'inscrire dans le cadre d'une mise en œuvre dans des systèmes mécatroniques : cette intelligence peut supporter de nombreuses et nouvelles fonctions, telles que la surveillance, le diagnostic, la maintenance, la calibration, la coordination, la synchronisation, la simulation, la communication... fonctions qui seront embarquées sur l'objet même. Finalement, c'est ce saut technologique vers l'autonomie et l'intelligence d'un objet qui qualifie le mieux son caractère mécatronique : architecture système pluritechnologique et forte capacité de traitement de l'information.

Dans ces conditions, un système complexe, composé de plusieurs entités mécatroniques, peut-il être lui-même qualifié de mécatronique ? A notre sens, c'est une condition nécessaire, mais non suffisante. Si un tel système est organisé de manière hiérarchique sans prendre en compte de nouvelles capacités décisionnelles propres aux entités le constituant,

nous sommes face à une approche relativement classique de la conception d'un système, sans caractère innovant particulier. Par contre, si les entités mécatroniques qui le composent sont capables, de manière autonome, coordonnée et égalitaire, d'assurer le fonctionnement global du système, sans l'intervention d'un système central de supervision, de commande ou de management, nous sommes face à un saut technologique du même ordre. Nous qualifions cette architecture décisionnelle d'isoarchie, au sens où chaque entité est dotée de la même capacité de décision. Cette propriété peut être aisément obtenue lorsque les mécanismes de décision sont dupliqués sur chaque entité et paramétrés à façon sur chacune. Cette propriété d'isoarchie, appliquée à un système complexe, justifie l'introduction du qualificatif mécatronique comme postulat pour la définition d'un système mécatronique complexe : toutes les décisions sont prises grâce à l'autonomie des constituants mécatroniques du système. L'absence d'un système de décision central interdit alors toute organisation prédéfinie ou prévisionnelle du fonctionnement du système : ce fonctionnement va devoir être organisé au fur et à mesure par les entités. Cette auto-organisation sous-entend un caractère temps réel qui prend en considération toutes les informations caractérisant chacune des entités contribuant à l'établissement de ce fonctionnement. Nous parlons alors de fonctions de pilotage auto-organisé. Ces fonctions sont intégrées dans l'intelligence associée à chacune des entités mécatroniques composant un système mécatronique complexe.

Dans ces conditions, une entité mécatronique et/ou un système mécatronique peuvent être décrits comme un *tout*, dans la mesure où ils représentent quelque chose de complet en soi, n'ayant besoin de rien d'autre pour assurer leur fonctionnement, et comme une *partie*, dans la mesure où ils ne sont qu'un élément d'un système plus général dont ils assurent une partie du fonctionnement, aussi bien au niveau fonctionnel qu'au niveau décisionnel. Ce constat nous rapproche d'une vision holarchique des systèmes mécatroniques.

Nous allons décrire dans ce papier une telle approche holarchique appliquée dans le contexte des systèmes de production automatisés flexibles. Nous illustrons la notion de machine mécatronique, entité robotisée à la fois constituée de composants mécatroniques et constituant de systèmes manufacturiers. Nous verrons que cette définition peut être récursive et que notre approche peut être déployée à plusieurs niveaux de granularité pour piloter de manière mécatronique le fonctionnement de systèmes manufacturiers. Le système de pilotage que nous proposons apporte un caractère mécatronique au système piloté *via* ces propriétés d'auto-organisation et d'isoarchie. Il s'applique à des systèmes de production, mais aussi à des machines telles que des robots industriels ou des MOCN. Après avoir présenté les différents concepts liés à notre approche, nous développons notre proposition d'un système de pilotage auto-organisé et isoarchique, basé sur une capacité de pilotage autonome associée à chaque entité mécatronique. De plus, cette approche permet de traiter le prévu et l'imprévu de la même manière, ce qui garantit robustesse de fonctionnement, réactivité et autonomie. Dans une telle solution, ce pilotage autonome contrôle le fonctionnement de chaque entité mécatronique. Pour cela, il est doté de deux fonctions particulières. La première permet de gérer les interactions avec les autres entités mécatroniques. La seconde permet de situer les capacités de l'entité vis-à-vis de ce que les autres entités peuvent faire pour que le système global puisse continuer à évoluer. Nous décrivons un modèle de comportement de ce pilotage autonome en utilisant le formalisme DEVS (*Discret Event system Specification*) (Ziegler, 1973). Ce formalisme permet la spécification de modèles modulaires et hiérarchiques, ce qui est très important pour une définition précise de ce comportement. 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 sont détaillés. Enfin, nous présentons un éventail de recherches concernant le développement de systèmes de pilotage auto-organisés et isoarchiques pour systèmes mécatroniques relevant du domaine de la production manufacturière automatisée. Ce papier n'a pas l'ambition de présenter un état de l'art mais de dresser un bilan des recherches menées dans notre équipe dans le domaine de l'approche décentralisée et auto-organisée du pilotage des systèmes multiphysiques, appliquée aux systèmes mécatroniques de production. Deux types de systèmes sont abordés : des systèmes du domaine de la robotique et des systèmes du domaine de la productique. Ces travaux synthétisent essentiellement une expérience de recherche d'une quinzaine d'années. Ces différents travaux trouvent aujourd'hui leur cohésion et leur unité *via* l'approche holonique dont nous présentons ici le modèle formel DEVS. Dans chacun des cas présentés, nous précisons l'objectif attendu du système en question, le recensement et le rôle des différents holons en interactions, ainsi que les résultats obtenus. Nous concluons en insistant sur le caractère générique, récursif et formel de notre approche holonique du pilotage des systèmes mécatroniques complexes.

## **2. Caractère holonique des systèmes mécatroniques complexes**

Dans cette section, nous allons présenter ce qui rend particulier un système mécatronique complexe par rapport à un système plus classique et nous positionnons ces différences vis-à-vis des approches conventionnelles du pilotage des systèmes. Nous terminons par l'introduction de notre approche du pilotage des systèmes mécatroniques complexes.

## 2.1. Théorie des systèmes, théorie des hiérarchies

La notion de système est une abstraction. C'est la représentation qu'en fait un observateur qui va être considérée. Cette représentation sera forcément partielle vis-à-vis de la réalité, car elle aura été construite avec un point de vue et selon une finalité : elle doit être utile à l'observateur. Un système n'existe donc qu'au travers du modèle qu'on s'en fait dans un but de simplification et de compréhension. De très nombreux travaux ont étudié la modélisation des systèmes et ont proposé des modèles 'standard'. Par exemple, lorsque la notion de système complexe est énoncée, la référence à la théorie du système général (LeMoigne, 1977) s'impose tout naturellement. Tout système complexe y est décrit au travers du modèle canonique OID (Opérationnel, Informationnel, Décisionnel). Ce modèle propose la structuration d'un système en trois sous-systèmes fonctionnels (figure 1) :

- le sous-système Opérationnel, qui assure les aspects matériels de l'activité du système, ce qui se traduit par l'expression d'une valeur ajoutée par le système sur son environnement (un système qui n'aurait aucun impact sur l'extérieur est un système qui n'existe pas),
- le sous-système Informationnel, qui gère les aspects immatériels permettant la représentation du système dans son environnement, c'est-à-dire l'observation et la mémorisation des transactions du système avec son environnement et des variables d'état interne représentatives de son comportement,
- le sous-système Décisionnel, qui assure le pilotage du système en vue d'un objectif à atteindre, et qui pour cela traite de l'information, analyse le différentiel entre l'état du système et l'objectif à atteindre, prend des décisions influençant le comportement du système et régule ce comportement en fonction de ces décisions.

Par ailleurs, tout système peut être décomposé en sous-systèmes d'ordres inférieurs, eux-mêmes décrits par le modèle OID. Cette récursivité permet donc de modéliser tout système artificiel, y compris des systèmes mécatroniques simples.

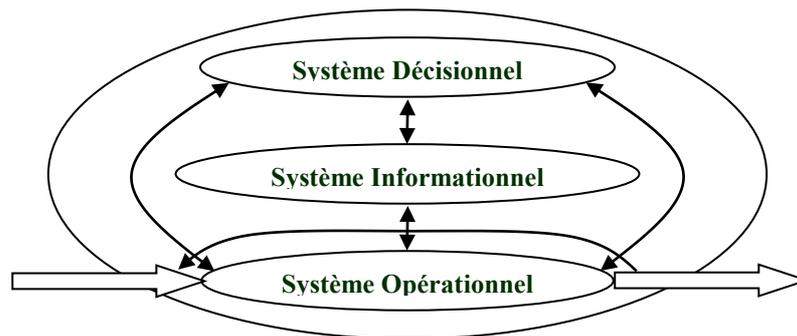
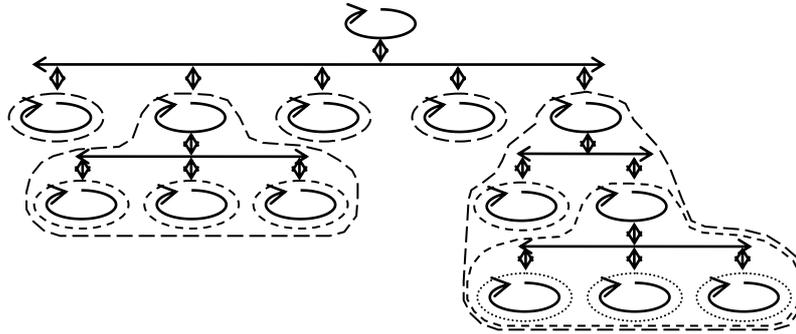


Figure 1. Modèle canonique OID d'un système complexe (LeMoigne, 1977)

Le modèle OID est alors un bon support pour illustrer l'intégration des technologies caractérisant un produit mécatronique : une partie de ces technologies a trait à la réalisation de la valeur ajoutée apportée par le produit (les aspects mécanique, hydraulique, électrotechnique...) tandis que l'autre partie assure le pilotage (les aspects traitement numérique liés à la microélectronique, à l'acquisition des capteurs et au traitement des informations et à la prise de décision automatique...). Toutefois, cette approche de la modélisation d'un système est centrée sur ce qui se passe à l'intérieur du système : elle ne permet pas de décrire la part décisionnelle que peut avoir un système sur un système d'ordre supérieur. En cela, la théorie du système général n'est pas vraiment adaptée à la modélisation des systèmes mécatroniques complexes, car elle ne répond pas bien aux contraintes induites par notre postulat de définition.

Arthur Koestler, via la théorie des hiérarchies, rejette les courants classiques de la sociologie scientifique, tels que le behaviorisme – doctrine de la psychologie basée sur l'étude des comportements visant à l'établissement de lois constantes à partir d'observations – et l'holisme – doctrine épistémologique qui ramène la connaissance du sous-système à celle du système auquel il appartient. Il propose dans (Koestler, 1967) de revisiter la notion de décomposition hiérarchique des systèmes (*Open Hierarchical Systems Theory*) : tout système doit être observé comme une hiérarchie mult niveau de sous-ensembles semi-autonomes formant des arborescences verticales et des réseaux horizontaux (figure 2). L'élément de base de cette approche y est appelé 'holon', néologisme créé à partir du grec 'holos', le tout, et du suffixe 'on', désignant une particule élémentaire. Ce holon peut représenter un système, avec un intérêt supplémentaire vis-à-vis de l'application aux systèmes mécatroniques. Un holon est à la fois l'expression des propriétés du système qu'il représente – c'est-à-dire qu'il est un *tout* cohérent et autonome –, mais aussi de celles relevant du système auquel il appartient – c'est-à-dire qu'il

est une *partie d'un tout* au fonctionnement duquel il contribue. L'approche holonique peut s'appliquer à toutes sortes de systèmes : des systèmes naturels – biologiques, par exemple – et des systèmes artificiels – c'est-à-dire créés par l'homme.



**Figure 2.** *Modèle hiérarchique d'un système holonique*

Ces deux aspects du holon sont très importants pour notre approche des systèmes mécatroniques. L'entité mécatronique, qui peut être elle-même un système mécatronique, tire de l'intégration technologique qui résulte de sa conception un certain nombre d'avantages qui lui confère une autonomie fonctionnelle propre. Par exemple, elle sera capable d'effectuer une autocalibration qui lui permettra de suppléer à des problèmes récurrents entraînant normalement des actions de maintenance. Ce premier aspect confère à l'entité mécatronique l'identité propre d'un tout autonome, c'est-à-dire existant par lui-même (*Self-Assertative tendencies*). Par ailleurs, l'entité mécatronique, en tant que holon, se trouve être une partie intégrée dans un tout (*Integrative tendencies*). Sur ce second aspect, l'intelligence associée à l'entité mécatronique doit lui permettre de contribuer pour une partie au fonctionnement global du système. En fait, le concept de holon permet de représenter de manière relativement souple diverses configurations architecturales, selon l'équilibre relatif entre tendances Self-Assertative et Integrative. Cela va des architectures purement hiérarchiques à des architectures hétérarchiques, avec un intérêt prépondérant pour ces dernières. Trentesaux (2002) dresse un panorama très complet des architectures possibles, dans le cadre du pilotage des systèmes manufacturiers. Par ailleurs, le concept de holon est tout à fait compatible avec le modèle canonique OID : il intègre les aspects immatériels (information et décision) et les aspects matériels (objet physique réel). Dans la suite de ce papier, nous ne précisons pas toujours auquel(s) de ces aspects le terme de holon (*H*) fait référence, car cela est sous-entendu par le contexte. Nous présentons donc dans ce papier une solution de pilotage du système mécatronique qui répond à ces différentes assertions et qui relie le concept de pilotage aux autres concepts en relation avec ce système. Nous allons projeter cette approche holonique sur des systèmes mécatroniques qui pourraient être rencontrés dans les systèmes de production manufacturière, soit en termes de machines mécatroniques, soit en termes d'association de machines mécatroniques.

## **2.2. Architecture holonique d'un système de pilotage auto-organisé**

De très nombreux travaux portent sur l'étude de ces architectures hétérarchiques. Le travail présenté ici relève de cette voie. Le projet de recherche vise à proposer une architecture de pilotage fondée sur une prise de décision totalement auto-organisée et décentralisée, c'est-à-dire s'effectuant sans aucune considération d'ordre hiérarchique. Dans ce contexte, nous considérons un système mécatronique comme décomposable en entités mécatroniques 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 au niveau du système global. Dans un système de pilotage auto-organisé, chaque entité participe à l'achèvement du but commun en assumant l'organisation et le management de son propre pilotage (Pujo et Ounnar, 2001) : elle prend part à l'élaboration et à l'évaluation de solutions, en l'absence de hiérarchie. Ceci s'appuie sur une structure holonique dont nous allons décrire la constitution et l'architecture.

### **2.2.1. Choix du concept de Holon**

La distribution de la prise de décision constitue l'un des domaines de l'intelligence artificielle distribuée (IAD). Elle est généralement abordée *via* des systèmes multi-agents (SMA). Ces SMA sont effectivement relativement en adéquation avec les structures de pilotage non conventionnelles. Toutefois, le concept de 'holon' présente la particularité forte, par rapport au concept d'agent, de pouvoir représenter un système cohérent et autonome tout en étant simultanément une partie d'un

système de niveau supérieur. Cette dualité fonctionnelle permet d'implanter dans un holon des comportements à haute valeur décisionnelle lui conférant une grande autonomie. On peut dire que le concept d'holon est récursif et que celui d'agent ne l'est pas. Une définition minimale d'un agent (il en existe beaucoup) est celle d'une entité qui perçoit son environnement, agit sur celui-ci et se comporte de manière rationnelle (raisonnement). Par conséquent, il est aisé d'adapter cette approche au pilotage non purement hiérarchique. Un agent est essentiellement un objet logiciel autonome, finalisé et capable de communiquer. Au contraire des autres objets logiciels, un agent fonctionne suivant sa propre initiative. Un système d'agents est conçu pour atteindre les objectifs programmés d'avance par son concepteur. Dans un SMA, tout est distribué : la connaissance, le contrôle, les compétences et les activités. Les holons ont tendance à préserver leurs individualités (chacun a son rôle dans le système) et à les utiliser pour maintenir l'harmonie dans le groupe. Une autre caractéristique de l'intelligence artificielle distribuée (IAD) réside dans la dimension cognitive propre à l'approche multi-agent (notion de plan, d'intention, de croyance, de but, etc.), tout comme la dimension purement réactive à base de population pseudovivante (colonies de fourmis, par exemple) permet de mettre en évidence l'émergence d'un comportement social ou d'un comportement global intelligent à partir de comportements sociaux élémentaires. L'approche multi-agent favorise plus une distribution du pilotage (intelligence artificielle distribuée) qu'une décentralisation : l'isoarchie est difficile à obtenir, car il existe toujours un agent animateur, facilitateur, coordinateur ou superviseur (il existe différentes terminologies, mais cela se traduit 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). Par ailleurs, pour travailler avec les propriétés de récursivité, la communauté multi-agent se tourne de plus en plus vers le paradigme holonique : on parle de plus en plus de système multi-agent holonique dès que le système étudié est plus complexe et qu'il est nécessaire de l'aborder de manière plus structurée *via* une modélisation récursive. Nous pouvons citer à ce propos les travaux de (Pichler, 2000) et de (Rodriguez *et al.*, 2005). Enfin, l'absence de plateformes génériques de développement de systèmes holoniques amène souvent les chercheurs en productique de la communauté HMS à développer certains concepts innovants grâce au paradigme holonique, puis à les opérationnaliser *via* des plateformes multi-agents du type de JADE par exemple. C'est une solution qui est tout à fait possible, en particulier lorsque l'architecture holonique proposée n'est pas isoarchique. En ce qui nous concerne, nous avons préféré développer l'intégralité des maquettes et démonstrateurs dont nous évoquons le fonctionnement dans notre papier.

### 2.2.2. Description de l'architecture de pilotage proposée

Plusieurs aspects doivent être pris en compte dans un système artificiel tel que nous l'avons présenté. Pour cela, nous identifions trois familles d'holons, selon que nous considérons les holons extérieurs au système en question, mais qui sont en relation avec lui, ceux qui composent le système physique proprement dit, et ceux qui définissent ses objectifs – c'est-à-dire ce qui va générer les interactions. Cette typologie est largement inspirée par l'architecture de référence PROSA (Van Brussel *et al.*, 1998). PROSA, développé dans le cadre des recherches effectuées dans le programme IMS (*Intelligent Manufacturing System*) (Mathews, 1995 ; Deen, 2003) sur l'hétéarchie dans les systèmes manufacturiers, permet de décrire des approches holoniques selon une décomposition en holons de type *Produit, Ressource, Ordre, et Staff*. Les holons *Staff* n'ont aucun pouvoir décisionnel, mais seulement une capacité exploratoire et propositionnelle. Ils n'ont aucun caractère obligatoire. Ceci autorise des structures de prise de décision isoarchiques, que les architectures holoniques autres que PROSA ne permettent pas : ADACOR (Leitão *et al.*, 2006) se positionne sans ambiguïté comme rajoutant aux holons de base de PROSA un holon superviseur responsable de la coordination, de l'optimisation dans un groupe de holons : il s'agit d'un centre de décision local et centralisé ; MetaMorph (Maturana *et al.*, 1999) est une architecture de pilotage holonique basée sur un pilotage *via* un ensemble de médiateurs qui sont autant de centres de décision centralisés. Précisons également que l'IEC 61499 qui reprend la notion de blocs fonctionnels de l'IEC 61131 en y ajoutant des mécanismes de déclenchement et de gestion des flux d'événements est globalement en accord avec l'architecture MetaMorph (Balasubramanian *et al.*, 2001). Cette norme ne permet pas des mécanismes de prise de décision isoarchique.

Nous proposons donc d'utiliser PROSA en élargissant son champ d'application aux systèmes mécatroniques complexes et en utilisant le cœur de PROSA, c'est-à-dire les 3 types de holons de base : les *Holons Produit*, les *Holons Ressource* et les *Holons Ordre*. Nous allons présenter succinctement ces holons, en montrer les limitations vis-à-vis des systèmes mécatroniques et en proposer une lecture étendue.

Dans PROSA, les *Holons Produit (HP)* donnent un descriptif technique des produits manufacturés (modèle physique du produit (géométrie), modèle de transformation (gamme), modèle de précision (exigences qualité)...). Il ne s'agit pas à proprement parler de holons attachés aux éléments du système de production, mais de holons attachés à des éléments extérieurs qui le traversent pour y être transformés. Dans le cadre d'un système mécatronique, cette situation peut également exister. Toutefois, il faut la compléter par la situation opposée, où c'est le système mécatronique qui traverse un environnement extérieur ou est simplement à l'intérieur. Ce faisant, le système mécatronique peut éventuellement le

transformer ou être transformé par lui. Les différents aspects de ce monde extérieur doivent être décrits par un modèle physique (la scène ou le produit sont définis par une description géométrique et leur propriété), un modèle de comportement (comment se déroule l'évolution du monde extérieur au système mécatronique en fonction des tâches effectuées) et un modèle de précision (visant à donner des intervalles de tolérance, des marges de sécurité anti-collisions...).

Les *Holons Ressource (HR)* correspondent dans PROSA aux entités du système de production. Van Brussel *et al.* (1998) montrent que la structuration hiérarchique des ressources est un facteur qui simplifie l'analyse du comportement. Ainsi, un holon 'usine' est un ensemble d'holons 'atelier', lesquels sont constitués d'holons 'poste de travail' composés d'holons 'équipement'. En continuant cette décomposition, nous pouvons aisément observer qu'un holon 'équipement' est un ensemble de holons 'opérateurs d'action', lesquels sont constitués d'holons 'actionneur' et/ou d'holons 'percepteur' (capteurs) (figure 3). Nous voyons là apparaître la continuité entre les mondes de la Productique et de la Mécatronique.

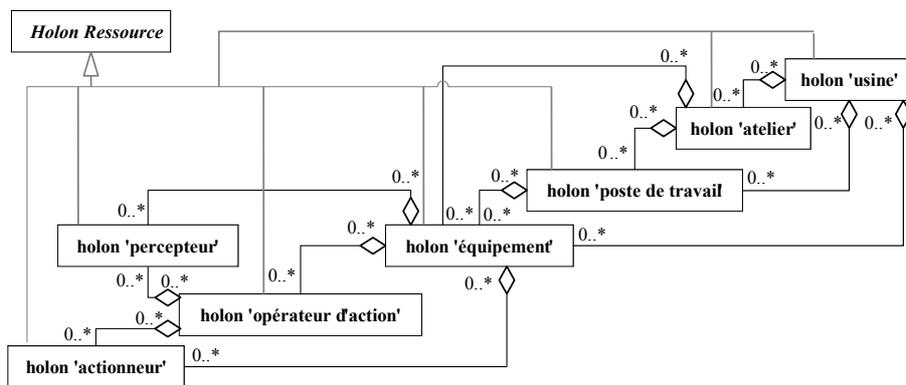


Figure 3. Les Holons Ressource d'un système mécatronique de production, inspiré de (Van Brussel *et al.*, 1998)

Pour assurer le bon fonctionnement du système, les relations entre *HP* et *HR* doivent être parfaitement maîtrisées. Généralement, ce sont les *HR* qui agissent sur les *HP*. Toutefois, les *HR* doivent rester à l'écoute des conditions de fonctionnement imposées par les *HP*, afin de s'adapter en permanence et d'optimiser le fonctionnement. Ces interactions mutuelles nécessitent une organisation qui se décline en deux aspects : la description des objectifs du système et leur projection dans le temps.

Dans le cas des systèmes manufacturiers, les *Holons Ordre (HO)* décrivent des tâches à effectuer sur les *HP* par les *HR*. Ils assurent également la gestion de l'état courant du *HP* en cours d'exécution et l'archivage des tâches achevées. Il s'agit là de la gestion et du suivi des tâches : traiter les *HP* avec les *HR*. Nous proposons de généraliser la notion de *Holons Ordre* à la notion de tâche, compatible avec tout type de systèmes automatisés (figure 4).

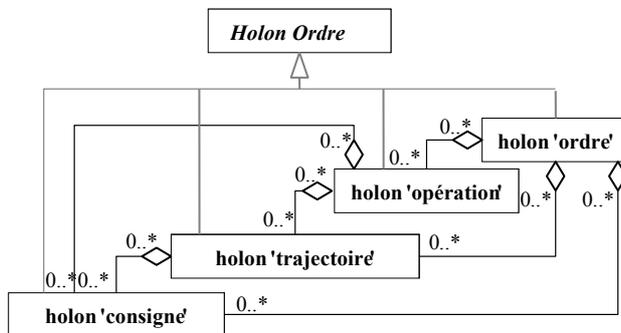


Figure 4. Les Holons Ordre d'un système mécatronique de production

Cette description des tâches doit être complétée par une organisation temporelle. Grâce aux interactions *HP-HO* et *HR-HO*, un *HO* possède un pouvoir de décision local qui permet un pilotage ponctuel. Il s'agit là de comportements

réflexes, dont le principal intérêt réside dans la simplification et la sécurisation du comportement global du système : des comportements purement réactifs permettent facilement d'introduire des règles d'évolution élémentaires permettant par exemples des enchaînements entre machines successives... et des règles de sûreté de fonctionnement visant à empêcher l'exécution de décisions génératrices de dysfonctionnements. Ceci ne concerne donc que des cadres de décisions simples. Pour faire face à des décisions d'organisation temporelle d'une plus grande complexité, telles que l'ordonnancement ou le réordonnancement des tâches, nous proposons de décentraliser cette fonction 'organisation' : un système de pilotage auto-organisé et décentralisé sur chaque *HR* permet d'assurer cette fonction selon une approche mécatronique et de traiter de manière similaire les systèmes robotiques et manufacturiers. Ainsi, nous allons proposer des mécanismes d'affectation décentralisée des tâches aux ressources fondés sur la génération automatique des actions liées à l'exécution de chaque tâche, en fonction des caractéristiques des ressources, et sur l'évaluation concurrentielle des performances obtenues, afin d'opter pour la solution la plus avantageuse.

### 2.3. Décentralisation du pilotage des Holons Ressources

Nous proposons donc de positionner localement un centre de décision pour le pilotage de chaque ressource directement associé à chaque *Holon Ressource*. Caractérisé par son contexte d'emploi dans un cadre de pilotage décentralisé, nous l'appelons Entité de Pilotage Autonome (EPA). Chaque EPA communique avec ses homologues *via* un réseau de communication. Ensemble, elles assurent une répartition des tâches (*HO*), sur la base d'une évaluation locale de la performance potentielle du *HR* associé et d'une négociation collective pour l'affectation optimale de ces tâches.

Dans notre proposition, le pilotage auto-organisé et décentralisé se caractérise par une architecture du type 'structure holonique horizontale' (*flat holonic form*) (Bongaerts *et al.*, 2000) et sur la duplication des EPA (une par *HR*), l'ensemble formant alors une structure isoarchique. Notre approche du concept d'auto-organisation est fondée sur une architecture décentralisée de la prise de décision et sur la prise en compte du 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 préalablement établie. Il n'y a donc plus aucune considération hiérarchique et/ou planificatrice.

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 5 montre comment les échanges entre entités partenaires passent par les EPA et comment les partenaires sont en relation *via* le réseau de communication de la structure holonique horizontale. Face à une tâche dont l'exécution lui est proposée par un message contenant un Appel d'Offre (AO), une EPA a la capacité d'auto-évaluer la performance qu'elle peut lui associer dans le but de participer aux négociations d'affectation de cette tâche.

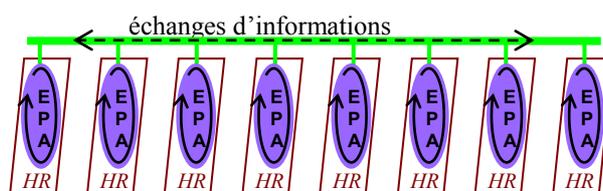


Figure 5. Les EPA dans une structure holonique horizontale de HR

Toutes les informations nécessaires à l'évaluation de performances par l'EPA se trouvent dans les holons de base de PROSA. Chaque EPA a des échanges privilégiés avec le *HR* auquel elle est associée. Le *HR* fournit par exemple des informations sur son planning prévisionnel, ses caractéristiques techniques, ses performances... Les données liées aux *HO* et aux *HP* circulent sur le réseau *via* les Appels d'Offre (AO). En résumé, les EPA pilotent les *HR* qui mettent en œuvre les *HO* sur les *HP* : c'est ainsi que la valeur ajoutée est donnée à la matière d'œuvre.

Une EPA qui a obtenu la responsabilité de l'exécution d'une étape *i* de l'ajout de valeur sur un *HP* a pour cela évalué au préalable la faisabilité de la tâche correspondante sur le *HR* 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 aux informations contenues dans le *HP*, une description relative à l'état initial du *HP* à l'entrée de l'étape *i* : il s'agit de la somme cumulée de tous les ajouts de valeur subis par le *HP* 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*, *i.e.* 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<sub>i</sub> est également porteur d'informations liées au HO : priorité, date de début au plus tôt de la tâche et date limite de fin d'achèvement... L'analyse de toutes ces données, utilisées pour générer une tâche compatible avec les capacités du HR, associée à la gestion du planning d'occupation de ce dernier, permet l'évaluation de la faisabilité de la tâche sur le HR en question.

Si cette performance se révèle être meilleure que celles proposées par d'autres EPA, nous avons trouvé une meilleure solution pour l'exécution de la tâche *i*. Au bout d'un certain temps, défini en fonction de la dynamique de l'application, l'AO<sub>i</sub> est clôturé. Il y a alors affectation de l'AO<sub>i</sub> : la meilleure EPA à cet instant prend en charge, par un pilotage direct de son HR, 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<sub>i+1</sub> qui contient les éléments éventuellement actualisés du HO et du HP (figure 6). Ce principe permet de poursuivre pas à pas le processus amenant la valeur ajoutée au HP.

### 3. Modélisation d'une entité de pilotage autonome

Nous venons de décrire le fonctionnement global de ce système de pilotage. Nous allons maintenant affiner la description interne d'une EPA. Puis nous développerons une modélisation selon le formalisme DEVS (*Discret Event system Specification*) (Ziegler, 1973 ; Ziegler, 1984) des différents modules permettant la réalisation du mécanisme d'interaction qui engendre l'auto-organisation du pilotage.

Nous avons choisi DEVS pour décrire nos modèles, car ce formalisme s'abstrait totalement de la mise en œuvre des simulateurs mettant en œuvre la modélisation du système. Nous avons donc des modèles n'ayant qu'une seule interprétation et indépendants de toute technologie informatique ou de tout simulateur du commerce. Les fondements d'une description formelle en DEVS et les règles de composition graphique des modèles DEVS sont décrits dans (Chêne *et al.*, 2004).

#### 3.1. Modélisation d'une entité de pilotage autonome

Un certain nombre de primitives fonctionnelles sont communes à toutes les EPA, quelle qu'en 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 6).

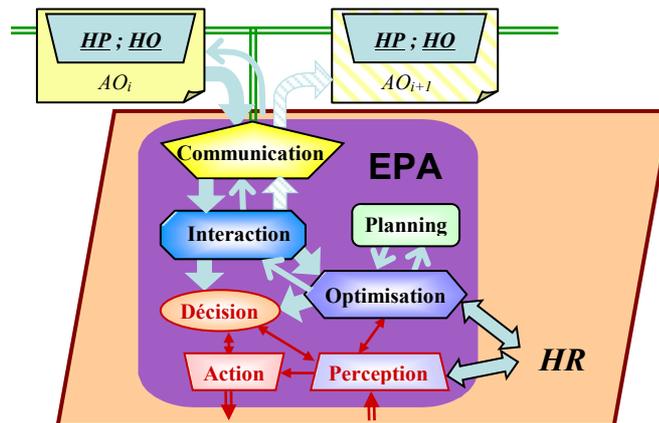


Figure 6. Relations entre les holons de base et les modules d'une EPA

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

### 3.1.1. Description des modules d'une EPA

Chaque EPA est 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... Selon les cas, différentes technologies ont été utilisées et validées : réseaux de terrain (CAN, VAN...), réseaux locaux (TCP IP *via* Winsocket) ou applications de communication. Dans le cadre de maquettes de faisabilité, nous utilisons l'environnement de simulation distribué HLA (*High Level Architecture*, IEEE P1516). 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, ce qui limite 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. On peut dire que globalement, ils assurent les mêmes fonctions de contrôle commande qu'un Directeur de Commande Numérique. Le module Décision gère les modes de marche de la ressource, c'est-à-dire qu'en fonction de l'état de supervision dans lequel il se trouve, il lance les programmes de tâche en délivrant au module Action les listes des commandes à exécuter. Le module Perception traite l'acquisition de données techniques pour assurer le bouclage tâche (*via* le module Action). Il assure également 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 le cadre de ce papier, où nous présentons essentiellement le fonctionnement du mécanisme d'interaction entre EPA, nous faisons l'hypothèse d'un fonctionnement idéalisé du *HR* vis-à-vis de l'analyse faite par le module Optimisation. Cette hypothèse nous conduit à ne pas traiter ici les relations entre les modules Interaction et Optimisation d'un côté, et Décision et Perception de l'autre côté.

### 3.1.2. Modèle DEVS couplé d'une EPA

Le comportement de différentes EPA simultanément en interaction est compréhensible à partir du modèle formel d'une EPA. Une EPA est modélisée à l'aide du formalisme DEVS couplé (figure 7). Nous ne donnons ici qu'une partie de ce modèle : celle correspondant aux modèles DEVS couplés des modules Interaction, Optimisation et Planning. Ils constituent donc ici l'ensemble des composants de l'EPA que nous abordons dans la suite de ce papier.

Les variables circulant entre les EPA sont des appels d'offre et des réponses :

**AO** : Appel d'Offre émis par l'EPA d'un autre *HR* du système mécatronique et reçu par l'EPA (il s'agit là d'un objet reprenant la totalité des informations nécessaires au traitement de l'AO et provenant des *HP* et *HO* en interaction).

**RAO** : Réponse reçue par l'EPA suite à l'Appel d'Offre précédent et émise par une tierce EPA (il s'agit là de la valeur de la performance d'un autre *HR* pour cet AO).

**AOL** : Appel d'Offre Local émis par l'EPA elle-même, pour la poursuite du processus amenant la valeur ajoutée au *HP* ou la recherche d'assistance auprès d'un autre *HR*.

**RAOL** : Réponse à un AOL, émise par une autre EPA et reçue par l'EPA.

**RAOE** : Réponse émise par l'EPA, suite à un Appel d'Offre reçu précédemment.

Nous appelons Trame la mise en forme des messages véhiculant ces différents objets : TrameAO et TrameRAO sont des classes d'objet définissant respectivement des AO et des RAO.

Les ports d'entrée et de sortie, ainsi que leurs valeurs sont ainsi définis :

**Ports d'entrée X**={AO1, RAO1, RAOL1, AOL0, Init} :

**AO1**={TrameAO} : indique la réception d'un AO, décrit par une Trame AO.

**RAO1**={TrameRAO} : indique la réception d'une réponse à un AO, décrite par une Trame RAO.

**RAOL1**={TrameRAO} : indique la réception d'une réponse à un AOL proposée par une entité sur le réseau. Il est de même type qu'un AO.

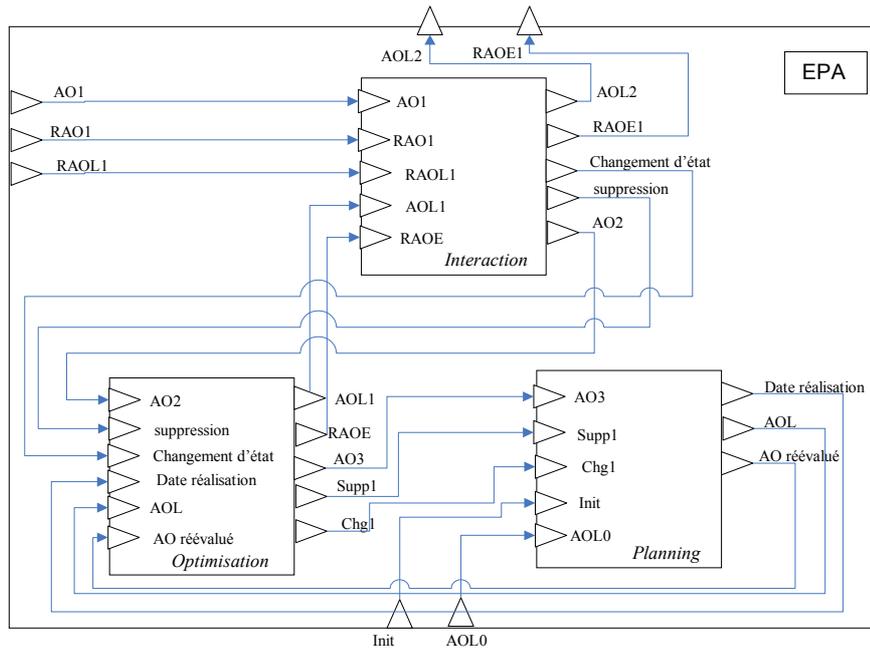
**AOL0**={TrameAO} : de même type qu'un AO.

**Init**={TrameAO} : une liste des TrameAO, nécessaire pour initialiser le planning.

**Ports de sortie Y**={AOL2, RAOE1}.

**AOL2**={TrameAO} : un AO local défini par une TrameAO.

**RAOE1**={TrameRAO} : une réponse de l'entité à un AO, défini par une TrameRAO.



**Figure 7.** Modèle DEVS couplé d'une entité de pilotage autonome

Les fonctions de couplage internes entre les modèles DEVS couplés des modules Interaction, Optimisation et Planning sont définies graphiquement par les liens décrits sur la figure 7 (Châne *et al.*, 2004). Nous allons maintenant approfondir ces modules et présenter partiellement leur modélisation.

### 3.2. Mécanisme d'interaction entre EPA

#### 3.2.1. Module Interaction

Le module Interaction permet d'assurer l'affectation des tâches par la mise en concurrence des *HR* pouvant potentiellement répondre à un AO. 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). Il s'agit d'observer sur un réseau quelle est l'EPA qui fait la meilleure réponse à un appel d'offre : c'est à elle que 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 affectations des tâches et des RAO et RAOL qu'il voit circuler sur le réseau, il 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 ; 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 préengagé : 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 préengagé, 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.

Vis-à-vis de ContractNet, notre protocole est simplifié : de nombreux messages de contrôle sont supprimés, car il n’est pas nécessaire que toutes les EPA répondent à chaque sollicitation. En effet, l’idée est de minimiser le nombre de messages d’interaction et de supprimer tout risque de blocage en cas de perturbations dans le système de communication entre EPA. L’AO est lancé par une EPA initiateur [e] auprès de tous les autres EPA, qui sont alors des EPA participant à la négociation. Une date limite de fin de négociation est associée à l’AO. Chaque EPA participant écoute tous les messages concernant ce AO et enrichit la réponse qu’elle élabore en fonction du contenu de ces messages. Elle répond [e] dès qu’elle peut fournir une réponse concurrentielle, c’est-à-dire uniquement si sa proposition est meilleure que celles déjà envoyées sur le réseau. Si elle ne peut fournir de réponse contenant une meilleure proposition avant la date limite de fin de négociation, elle ne répond pas [e]. A la date de fin de négociation, toutes les EPA participant et l’EPA initiateur savent quelle EPA a obtenu l’exécution de la tâche.

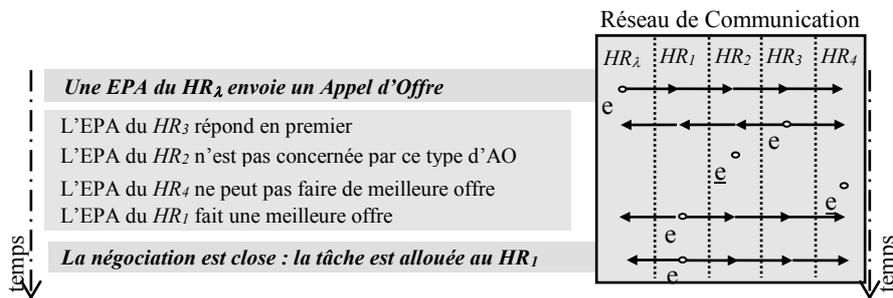


Figure 8. Déroulement chronologique du protocole d’interaction entre EPA

Ce protocole peut être vu selon la représentation FIPA (*Foundation for Intelligent Physical Agents*, (FIPA, 2002)) (figure 9). Nous voyons sur cette figure que ‘l’EPA initiateur’ envoie son AO aux m ‘EPA participants’, et chaque ‘EPA participant’, s’il a la capacité de répondre, envoie sa réponse aux m-1 autres participants et à l’initiateur. Faisons remarquer que notre protocole d’interaction est simplifié par rapport aux protocoles identifiés par la FIPA et n’utilise pas de systèmes de gestion centralisée de données, type ‘blackboard’.

Si aucune proposition n’est émise avant la date limite de fin de négociation, l’AO doit être relancé, après une éventuelle modification. Toutes les EPA peuvent être initiateur et participant. Plusieurs AO sont simultanément en négociation, puisqu’ils correspondent aux différentes tâches à exécuter dans un horizon proche sur chacun des HR.

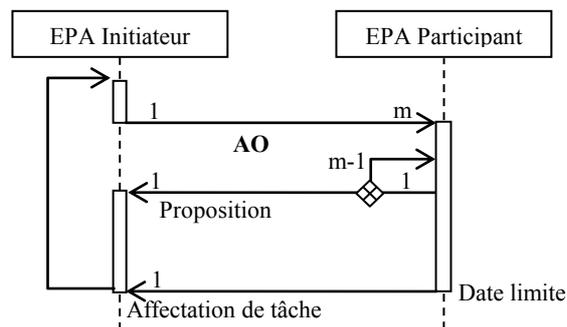


Figure 9. Représentation FIPA du protocole d’interaction entre EPA

Une EPA peut être donc simultanément initiateur d’un AO et participant de plusieurs autres. Ce protocole d’interaction donne une forte dynamique au système de pilotage. Il confère à l’EPA un comportement intelligent vis-à-vis du fonctionnement global du système auquel il appartient. C’est à ce titre que nous pouvons qualifier le HR doté d’une EPA associée de composant mécatronique d’un système mécatronique complexe.

### 3.2.2. Modèle DEVS couplé du mécanisme d'interaction

Nous avons modélisé le module d'Interaction à l'aide d'un modèle DEVS couplé composé de trois modèles atomiques (figure 10). 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 RAOE fournies par le module Optimisation. Le troisième modèle atomique mémorise toutes les performances RAOL reçues en réponse aux AO émis localement par l'EPA.

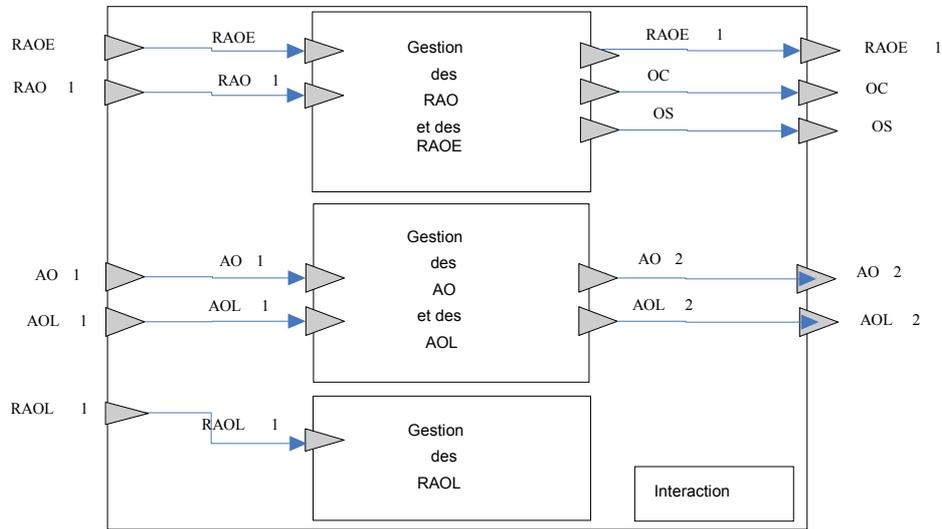


Figure 10. Modèle DEVS couplé du module d'Interaction

Nous ne pouvons détailler ici la totalité de ces trois modèles. A titre d'illustration, nous allons examiner le modèle DEVS du sous-module de gestion des AO et des AOL (figure 11). Ce modèle comporte 5 états, 2 ports d'entrée et 2 ports de sortie.

**Ports d'entrée X= {AO1, AOL1} :**

– AO1, AOL1 : du même type qu'un AO.

**Variables d'états S= {phase, T, L, L1} :**

– phase = {Attente initiale, Attente, AO reçu, AOL reçu, insertion}, avec  $\sigma$  : un réel positif égal à la durée maximale de l'état ;

– T : définit une instance de la classe d'objet de type TrameAO ; cette classe d'objet contient l- Num : numéro de l'AO, défini sur un ensemble des entiers ;

es attributs 'Num, Entité, Type, Etat, Date Réception, CE, DL' ;

- Entité : nom de l'EPA émettrice de l'AO, défini par une chaîne de caractères ;

- Type : décrit la faisabilité de l'AO, et prend ses valeurs dans : {vrai, faux} ;

- Etat : valeurs prises selon l'état de l'AO au niveau du planning de chaque EPA : {négociable, engageable, pré-engagé, engagé} ;

- Date Réception : date de réception de l'AO, de type date ;

- CE : condition d'exécution ;

- DL : délai de fin de la tâche ;

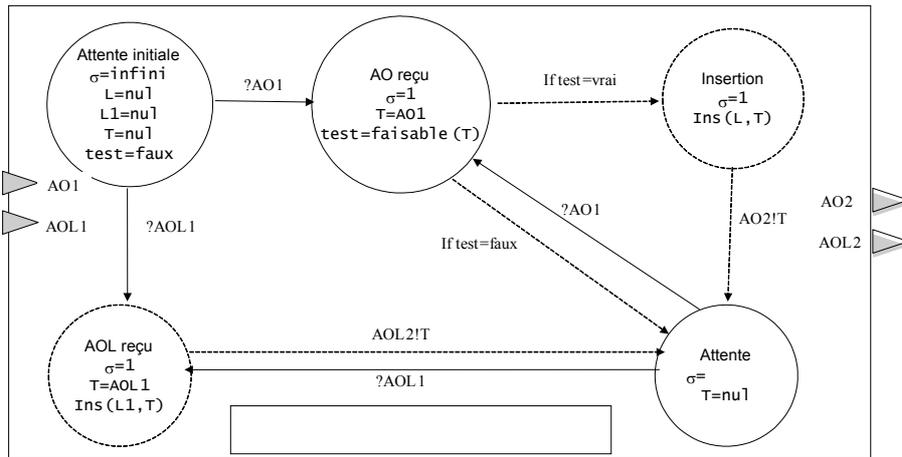
- plus tous les détails relatifs au HP et au HO ;

– L : une liste dans laquelle sont stockés les AO reçus ;

– L1 : une liste dans laquelle sont stockés les AOL émis par l'EPA.

**Ports de sortie Y= {AO2, AOL2}**

– AO2, AOL2 : de même type qu'un AO.



**Figure 11.** Modèle DEVS atomique du sous-module de gestion des AO et des AOL

Le modèle atomique reçoit un AO venant d'une des EPA. Il vérifie d'abord sa faisabilité : il faut que le HR ait la capacité de réaliser l'HO. Dans l'affirmative, il procède à l'insertion de l'AO dans une liste (L) qui contient tous les AO publiés sur le réseau. Ensuite, il le transmet au module d'Optimisation. Ce module peut aussi traiter un AOL proposé par l'EPA elle-même. Il l'insère dans une liste, puis le diffuse vers l'extérieur.

Tout ce mécanisme de prise de décision repose sur la capacité de chaque EPA à proposer une évaluation de la performance du HR associé pour exécuter une tâche à venir. C'est l'objet de notre prochaine section.

### 3.3. Optimisation et planning

#### 3.3.1. Module optimisation

Le module Optimisation permet à l'EPA d'évaluer la capacité du HR à effectuer la tâche décrite au travers de l'AO reçu : il s'agit d'estimer la performance correspondante. 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 : la différence entre les résultats qu'elles obtiennent provient des différences de caractéristiques techniques des HR et des différences de leur état courant à l'instant du calcul. Il y a donc une totale transparence pour l'obtention des résultats,

- les algorithmes d'évaluation de cette performance doivent être basés sur des modèles aussi proches que possible du comportement réel du HR ; pour cela, 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.

Ceci implique une relation très forte entre le module Optimisation de l'EPA et les données et fonctions du HR associé à cette EPA. 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 (optimisation, analyse multicritère...) et/ou de la fabrication assistée par ordinateur (algorithmes de génération de trajectoires pour machines outils ou robots industriels).

#### 3.3.2. Module planning

Le module planning permet d'étudier la possibilité d'insertion d'un AO dans le planning du HR en fonction de l'état des AO en cours de négociation. A partir de là, il calcule la date de réalisation de l'AO. A chaque changement d'état d'un AO, le planning est réévalué. Par exemple, lors du passage de l'état « engageable » à l'état « négociable » d'un AO, cela donne de nouvelles disponibilités pour d'autres AO : la tâche affectée à un HR ne devient ferme que lorsque cette tâche

est commencée. Ceci entraîne alors le déclenchement de la réévaluation par le module Optimisation de tous les AO en cours de négociation, afin d'en améliorer les performances.

#### **4. Illustration de l'approche *via* des machines industrielles mécatroniques**

L'approche auto-organisée et isoarchique dont nous venons de décrire les modèles DEVS génériques trouve son origine dans une thématique de recherche autour de la commande décentralisée d'équipements de production. Nous allons donner un rapide panorama des résultats que nous avons obtenus à Marseille dans ce domaine, avec les thèmes scientifiques successifs suivants : décentralisation de la commande, puis mécatronique, auto-organisation du pilotage et approche holonique isoarchique. Des démarches similaires de présentation sur des travaux connexes ont été présentées par (Ken Keys *et al.* 1995) ou encore (Lei *et al.* 1998). Toutefois, ces derniers travaux sur le pilotage hétérarchique et collaboratif se différencient des nôtres par une approche multi-agent basée sur l'architecture informatique distribuée CORBA. CORBA permet à une application de demander à un objet distribué d'effectuer une opération et d'en obtenir le résultat. CORBA ne permet pas d'assurer la synchronisation d'événements lors de l'exécution d'une simulation distribuée, qui est l'approche adaptée à la simulation d'un système isoarchique. Nous avons donc choisi HLA, qui est un protocole permettant de synchroniser un ensemble de simulateurs interagissant dans un but commun.

Il s'agit dans ce papier de présenter au travers du paradigme holonique et d'une modélisation formelle l'ossature commune à nos différents travaux.

Nous allons présenter de façon plus détaillée deux systèmes mécatroniques complexes et nous présenterons un éventail représentatif d'applications où notre approche du pilotage auto-organisé et isoarchique est applicable.

##### **4.1. Etude d'un robot industriel à architecture mécatronique**

###### **4.1.1. Du robot industriel à commande décentralisée au robot industriel à architecture mécatronique**

La commande décentralisée des machines industrielles de production a été initialement étudiée par (Brun Picard, 1988). Brun Picard a montré que les modèles comportementaux (géométrique et cinématique, par exemple) peuvent être calculés indépendamment axe par axe. Dans ces conditions, il est possible d'envisager également la commande axe par axe (Michel, 1988) des machines à commande numérique : robot industriel, machine outil... Ces travaux initiaux ont montré la nécessité de structurer les fonctions internes du système de commande décentralisé, en séparant par exemple les fonctions de communication des fonctions de décision ou d'action. La communication sert alors à assurer la synchronisation, la coordination et la surveillance des tâches robotisées.

A partir de là, une réflexion a été conduite sur la structure mécanique des robots industriels. S'il est possible d'obtenir une commande décentralisée, il est possible de ramener sur chaque axe de mouvement la fonction mécanique de conversion de couple (le réducteur), la fonction électromécanique de création de mouvement (l'actionneur), la fonction électronique de distribution de puissance (le variateur) et la fonction numérique de commande de mouvement (l'asservissement). Nous avons donc conçu des 'opérateurs mécatroniques de mouvement' localisés au niveau des articulations du robot industriel (Massé *et al.*, 1994). Ces opérateurs constituent les *HR* du robot mécatronique. Ils sont reliés entre eux par des corps mécaniques pour constituer la structure du robot. Les transmissions étant localisées dans les *HR*, elles ne traversent plus les corps. Il n'est plus nécessaire d'assurer une grande rigidité à ces corps. Il est alors possible d'alléger la structure du robot, augmentant ainsi la part de la puissance dissipée dans les actionneurs consacrée à la manipulation de la charge. Il faut alors pouvoir compenser les déformations de la structure mécanique sous l'influence des différentes charges. Pour cela, nous avons établi un modèle des déformations mécaniques dans un robot industriel (Pujo, 1994), puis nous avons intégré ces modèles dans le système de commande pour corriger les trajectoires de l'effecteur (Asgari, 1998).

###### **4.1.2. Description du système holonique**

Dans le cadre d'un système robotisé mécatronique, les *HR* sont sollicités par des AO contenant les *HP* et les *HO*. Les *HP* donnent les descriptions physiques des objets manipulés (géométrie, zones de préhension...) ainsi que celles des scènes de manipulation. Les *HO* définissent les conditions d'exécution des tâches à effectuer. Cela se traduit en termes de trajectoires (cycle, type de mouvement, conditions dynamiques...) et de précision requise pour le mouvement. Plusieurs types de tâches peuvent être envisagées.

Dans le cas de tâches simples, par exemple un mouvement point à point, la correction de trajectoire due aux déformations sera calculée uniquement aux points terminaux des trajectoires (début & fin). Le calcul de la performance de chaque *HR*, effectué par l'EPA associée, donne la durée du mouvement articulaire. Ceci permet aux EPA du robot de connaître le déplacement articulaire le plus long et d'assurer à partir de là la *synchronisation* de chaque mouvement : les *HR* débiteront leur action simultanément et la termineront de même, selon une planification de tâches gérée par le module Planning. Les problèmes de synchronisation inter axes ont été traités par Brun Picard (1988). Cela n'est pas décrit ici, car il s'agit d'interactions qui se déroulent directement entre les modules 'action'. Ces modules 'action' existent toujours dans les EPA d'un *HR* 'opérateur de mouvement', et les messages échangés ne sont pas des AO. Les EPA des *HR* 'opérateur de mouvement' organisent le mouvement global en fonction de la cinématique désirée (décrite dans le HO) et des limitations technologiques des *HR* 'opérateurs de mouvement'. La négociation ici consiste à segmenter la trajectoire pour maîtriser complètement les conditions d'exécution, avec pour chaque segment des conditions cinématiques et d'asservissement qui soient adaptées. Ce travail préparatoire se déroule juste avant le début de la trajectoire, à la réception du passage à l'état 'pré engagé' de l'AO correspondant à la tâche. Ces conditions d'exécution propres à chaque segment seront transmises aux modules 'décision' de chaque EPA qui se chargent de lancer les exécutions en série des mouvements asservis par les modules 'action'.

Dans le cas plus difficile d'une trajectoire avec interpolation à vitesse curviligne imposée, les EPA ont un rôle de *coordination* entre *HR*, c'est-à-dire que la synchronisation doit être effective à chaque instant. Cet objectif n'est pas simple à atteindre. En effet, si l'un des *HR* arrive en saturation, il n'est plus coordonné avec les autres *HR*, et la géométrie de la trajectoire suivie sera incorrecte. L'EPA doit donc négocier le découpage de la trajectoire en trois phases : avant la saturation, en suivant la vitesse curviligne imposée, pendant la saturation, en respectant les limitations du *HR* le plus pénalisant (c'est-à-dire en ralentissant de manière *ad hoc* la vitesse curviligne), et après la saturation, en reprenant la vitesse curviligne imposée. Là encore, ce sont les modules Planning de chaque EPA qui assurent la vérification et la validation de l'exécutabilité des tâches : l'exécution ainsi adaptée de la tâche va être un peu plus longue, et il faut vérifier si cela peut s'insérer dans la chronologie des autres tâches, ou au contraire si cela demande d'autres ajustements sur des tâches connexes. Bien entendu, les corrections dues aux déformations sous charge doivent également être considérées, mais elles n'ont pas d'influence sur le déroulement temporel de l'exécution des tâches. Toute tâche fait donc l'objet d'évaluation avant d'être lancée : une saturation peut être anticipée et une trajectoire ainsi découpée en segments chronologiques répondant à différentes logiques de commande. Les EPA ajustent ensuite les asservissements en fonction de l'exécution réelle du mouvement. C'est alors le module 'action' qui gère directement ces aspects, les synchronisations et les coordinations entre axes ne sont plus effectuées par des AO, mais par d'autres types de messages.

De nombreuses fonctions donnant plus d'autonomie aux opérateurs mécatroniques de mouvement peuvent également être facilement mises en œuvre. Par exemple, il est possible d'embarquer sur chaque articulation des fonctions de surveillance du bon fonctionnement des composants internes de l'opérateur. Ben Hmida (1998) propose ainsi d'observer les signaux issus des capteurs de l'opérateur de mouvement. En utilisant des techniques de filtres estimateur et de calcul de résidus, il est possible de détecter, de localiser et donc de diagnostiquer l'apparition de défauts sur ces composants ou sur l'actionneur.

Cette autonomie sur le fonctionnement interne des différentes entités mécatroniques qui composent le robot mécatronique peut être complétée par une autonomie fonctionnelle du système mécatronique qu'est ce robot. Prenons l'exemple d'une tâche de préhension d'un objet. Si cette tâche est simplement définie dans l'AO :  $f(HP, HO)$  par le choix d'un cycle d'approche paramétré, de surfaces de préhension sur l'objet et d'un cycle de prise de l'objet, il est possible de calculer localement, sur le robot, l'ensemble des mouvements à effectuer pour répondre à l'AO. C'est l'EPA générale du robot mécatronique qui réalise la *génération* des trajectoires correspondant à la tâche (Pujo *et al.*, 1995). L'EPA générale peut être une EPA spécialisée, correspondant à un *HR* 'système', mais peut être également localisée sur l'un des *HR* 'composant du système'. Dans ce cas, l'EPA 'locale' traite les décisions propres à son *HR*, mais traite également par délégation les décisions propres au *HR* 'système'. Ce que nous définissons comme l'EPA générale se contente de représenter la holarchie de *HR* 'composant du système'. Son objectif n'est pas de décider ce que doivent faire les composants de l'holarchie, mais de trouver des tâches que la holarchie peut effectuer.

#### **4.2. Vers une évolution des machines industrielles**

La même approche peut être envisagée pour des machines outils. Dès 1990, les principes de la commande décentralisée ont été testés sur des machines outils (Carmona, 1990). Sur les MOCN à architecture conventionnelle, les modèles comportementaux sont beaucoup plus simples qu'en robotique articulaire, et sont donc plus faciles à décentraliser. Notre expérience est limitée à des MOCN à architecture cartésienne, ayant peu d'axes de mouvement. Les

problèmes à résoudre sont relativement similaires à ceux rencontrés en robotique. La précision des interpolations requises est nettement plus importante, mais les modèles comportementaux sont plus simples et il existe une large littérature concernant la génération automatique de trajectoires sur MOCN. Pour notre part, nous nous sommes limités au cas du tournage (Pujo *et al.*, 1996).

Par ailleurs, notre approche holonique permet d'envisager des modèles temporaires pour assurer des fonctions de synchronisation et de coordination des *HR*. Par exemple, lors d'une tâche de manipulation d'un objet consistant à le passer d'un robot à une autre machine, il existe une phase délicate où l'objet est maintenu simultanément par les deux robots ou par un robot et une machine outil. L'objet est alors sollicité par deux commandes indépendantes l'une de l'autre, ce qui peut s'avérer gênant. Par une approche auto-organisée, il est tout à fait possible d'utiliser des modèles de comportement associant des *HR* appartenant à des machines différentes. Par exemple, les axes du poignet de l'un des robots vont être associés temporairement aux axes de l'autre. Cette nouvelle machine, virtuelle car créée artificiellement sur la base de parties opératives indépendantes, possède plus de degrés de liberté, et est commandable le temps du transfert de l'objet manipulé d'un robot à l'autre. Cette commande décentralisée multimachine va être gérée par les EPA : alignement des machines autour de l'objet manipulé, basculement vers la commande multimachine, saisie de l'objet par la seconde machine, relâchement de l'objet par la première, basculement vers les commandes décentralisées initiales des deux machines. Nous voyons là apparaître une autre caractéristique propre aux holons : un holon peut apparaître et disparaître au cours du fonctionnement du système mécatronique, et les relations qui lient les holons les uns aux autres sont elles aussi dynamiques.

Nous venons de voir comment à partir d'opérateurs mécatroniques de mouvement, composés chacun d'une partie opérative et d'un centre de décision de haut niveau (l'EPA), il était possible d'envisager le fonctionnement d'une machine mécatronique de production. Cette approche devient encore plus intéressante lorsque ces machines mécatroniques sont placées dans leur contexte d'utilisation, à savoir celui des systèmes de production automatisés flexibles.

Dans la section suivante, nous allons voir des exemples de systèmes de production pouvant eux-mêmes être qualifiés de mécatronique, grâce aux propriétés intrinsèques des machines mécatroniques qui les composent : ces propriétés leur permettent d'auto-organiser le fonctionnement global du système.

## 5. Illustration de l'approche *via* des systèmes manufacturiers mécatroniques

Le point précédent nous a amené à nous intéresser à l'aspect multimachine. Il s'agit en fait d'une extension de la thématique initiale 'commande décentralisée' au pilotage décentralisé de machines de production ainsi équipées. Ceci nous a amené à concentrer nos efforts sur la formalisation des mécanismes d'auto-organisation, qui sont plus complexes à établir au niveau multimachine : les contraintes devant être respectées par le système ne sont pas seulement des contraintes de synchronisation et de coordination, ce sont également des contraintes de coopération et de génération de tâches.

La *coopération* va focaliser sur l'organisation des tâches en termes de répartition, d'allocation et d'affectation des AO aux *HR*. L'*HO* est alors plus difficile à définir, car il est pluriel : il est composé de la concaténation de multiples objectifs distincts, mais devant être simultanément traités : par exemple, le traitement en cours des *HP* présents dans le système. L'*HO* se traduit plutôt en termes de recherche d'un cadre de solution permettant d'atteindre les sous-objectifs : nous sommes à un niveau où nous ne disposons plus de modèles de comportement de type modèles géométriques, cinématiques... qui permettent d'assurer l'unicité du comportement des opérateurs de mouvement. Malgré tout, l'*HO* peut être orienté par des objectifs globaux, par exemple de productivité maximale ou de production économique, qui vont contraindre l'ensemble du fonctionnement du système. Il en résulte une distribution 'en bonne intelligence' des actions à accomplir et cela va déboucher sur des décisions d'actions entraînant des synchronisations et des coordinations, selon différents mécanismes de coopération. Parmi ces mécanismes, nous pouvons citer comme cadre de prise de décision l'attribution, la négociation, le partage d'information, la mise en concurrence, la médiation ou l'arbitrage de conflits... selon les capacités décisionnelles dont seront dotées les EPA.

La *génération* est une fonction de création, en ligne, des actions à enchaîner pour réaliser une tâche, en fonction de son cahier des charges. Il ne s'agit plus d'envisager une action comme préalablement prédéfinie lors des étapes de préparation du travail, mais comme pouvant être construite en temps réel par les EPA, en fonction de l'objectif de valeur ajoutée qu'elle représente et des possibilités effectives des *HR* qui en auront la charge. Pour cela, les EPA disposent de règles de génération, basées sur des compétences et capables d'identifier chaque situation, d'élaborer des solutions localement optimales dans le respect d'un certain nombre de critères globaux de coopération. Ces règles vont jusqu'à la prise en compte de considérations purement technologiques, liées à l'activité du *HR* associé.

En réalité, la mise en œuvre de ce pilotage multimachine auto-organisé et décentralisé n'implique pas obligatoirement la décentralisation de la commande des machines concernées. Des machines à commande conventionnelles, c'est-à-dire centralisées, complétées par une EPA, peuvent faire parfaitement l'affaire. Des associations hybrides (c'est-à-dire comprenant des machines hétérogènes du point de vue de leur commande) peuvent également fonctionner. La récursivité de l'architecture décentralisée n'est pas une condition nécessaire... même si elle renforce à notre sens le caractère mécatronique d'un système complexe. Nous allons présenter succinctement quatre systèmes complexes pouvant être qualifiés de mécatroniques au sens de notre approche holonique.

## 5.1. Pilotage auto-organisé d'une cellule flexible de tournage

### 5.1.1. Présentation du cadre de l'étude

La cellule flexible de tournage est composée d'un certain nombre  $N$  de tours à commande numérique ayant des capacités différentes (puissance de broche, capacités dimensionnelles (diamètre maximum des objets produits...), outillage...), d'un certain nombre de postes périphériques spécialisés (mesurage, lavage) et d'un nombre  $K$  de robots de service (permettant la manipulation des objets produits entre les autres  $HR$ ). Cette cellule permet de réaliser des pièces de révolution et est dotée de zones palettes lui permettant de stocker quelques pièces en cours de fabrication, d'un magasin contenant un ensemble d'outils standard interchangeables, et enfin de zones palettes contenant les pièces brutes et celles recevant les pièces usinées.

### 5.1.2. Spécification du système holonique

Les  $HR$  composant cette cellule de tournage accomplissent des tâches de production s'articulant autour de cinq « métiers » différents : l'usinage de pièces mécaniques de révolution par tournage, la manipulation robotisée de pièces mécaniques de révolution, leur stockage, leur mesurage et leur lavage.

Les  $HR$  'tours à commande numérique' utilisent comme description de tâches un langage normalisé enrichi d'instructions assurant la synchronisation avec les autres  $HR$ . Les  $HR$  'robots' utilisent un langage effecteur du type AML, également enrichi des mêmes instructions de synchronisation. Les autres  $HR$ , qui sont des machines périphériques, utilisent plusieurs catégories de cycles prêts à être exécutés et qu'il est possible de lancer indifféremment. La génération de trajectoires sur chaque type de  $HR$  se fait dans les règles de l'art selon la connaissance « métier » associée (Pujo *et al.* 1995 ; Broissin, 1999). A partir du  $HP$  'pièce à réaliser n° xxx' (Pujo *et al.*, 1996), des caractéristiques du  $HR$  'tour<sub>x</sub>' et de celles de son outillage, la génération des trajectoires d'usinage par l'EPA associée à ce  $HR$  'tour à commande numérique' va se dérouler selon les étapes suivantes : choix des conditions de prise de pièce sur le tour, choix de l'outillage pour chaque opération d'usinage, choix des conditions de coupe pour chaque opération, génération des trajectoires d'outils et traduction en un programme de tâches pour commande numérique (langage ISO). En filiation directe des travaux portant sur la génération automatique de gammes de tournage (Anselmetti, 1985), cette génération de tâches sert d'une part, pour la génération automatique, en temps réel et sur l'EPA du programme de tâches correspondant à un AO et, d'autre part, pour l'élaboration de la performance d'un  $HR$  face à un AO. Cette performance est équivalente au coût de réalisation de la tâche. Nous obtenons ce coût par la somme du coût d'usinage proprement dit (fonction du coût horaire du  $HR$  et de la durée de l'usinage, laquelle dépend des trajectoires générées, de la puissance de broche du  $HR$ , des conditions de coupe choisies...), le coût de l'outillage (qui s'use), le coût de la préparation du  $HR$  (qui est en fait un coût d'immobilisation le temps de la reconfiguration), le coût de la matière, et le coût des temps d'attente lors du chargement et du déchargement du  $HP$  'pièce'.

Cette performance sert donc d'indicateur de comparaison entre les différentes EPA. En pratique, nous pouvons observer lors du fonctionnement de ce pilotage auto-organisé et décentralisé un effet de lissage des charges sur les différents  $HR$ . En effet, un  $HR$  très performant va être dans un premier temps plus chargé que les autres. Mais dès que son état de chargement devient trop important, cela se traduit par des attentes qui vont alors pénaliser sa performance. Les autres  $HR$ , moins performants en théorie, redeviennent alors concurrentiels et peuvent accéder à leur tour à l'affectation de tâches. En définitive, tant que tous les  $HR$  ne travaillent pas avec un taux d'occupation satisfaisant, cela signifie que de nouveaux AO peuvent être introduits dans le système de production mécatronique.

### 5.1.3. Résultats obtenus

Prenons l'exemple d'un  $HR$  'cellule mécatronique de tournage', où coopère un ensemble de neuf  $HR$  : {Tour<sub>1</sub>, Tour<sub>2</sub>, Tour<sub>3</sub>, Tour<sub>4</sub>, Robot<sub>1</sub>, Robot<sub>2</sub>, Mesure, Lavage, Stockage}. La maquette de simulation permet de visualiser la situation initiale correspondant à une production  $HP = \sum HP_{\text{pièce}}$  en cours (figure 12). A la date 800 est introduit un  $HO$  'lot urgent 4' de 6  $HP$  'pièce' comportant 2  $HO$  'phase' par  $HP$ . Les  $HR$  'tour' terminent les traitements en cours. La première ligne

de cette figure montre l'échelle des temps. Chaque tâche est repérée sur le Gantt par le triplet [n°lot, n°pièce, n°tâche], écrit verticalement au droit de la date d'engagement de la tâche par la ressource. Par définition, un HP 'lot urgent' est prioritaire. Son introduction remet en cause tous les préengagements des HR 'tour'. Les AO sont évalués par les EPA après tri par ordre de priorité, puis à priorité égale, par ordre chronologique (FIFO). Les produits en phase de finition restent dirigés normalement vers les palettes de sortie, mais les autres sont dirigés vers la zone de stockage temporaire et les EPA se répartissent en négociant les HP 'pièce<sub>x</sub>' de l'HO 'lot urgent 4' : 6 AO correspondant aux HO 'phase<sub>1</sub>' des HP 'pièce<sub>x</sub>' ont été émis en date 800.

Suivons le parcours du HP 'pièce<sub>1</sub>' [4,1,\*]. A la date 900, le HR 'robot<sub>1</sub>' prend le brut du HP 'pièce<sub>1</sub>' et le transporte [4,1,1] sur le HR 'tour<sub>4</sub>'. Cela signifie que c'est l'EPA du HR 'tour<sub>4</sub>' qui a remporté l'AO associé à l'affectation de la première phase d'usinage de la première pièce [4,1,2]. Cela s'explique, car il s'agit de la première machine à être libérée après l'introduction du lot urgent ! Avant la fin de cette tâche, pour préparer l'affectation de la seconde phase, l'EPA du HR 'tour<sub>4</sub>' diffuse un AO : f(HP 'pièce<sub>1</sub>', HO 'phase<sub>2</sub>'). Tous les HR 'tour<sub>x</sub>' étant déjà occupés à la production du lot urgent, c'est le HR 'stockage' qui remporte l'AO. Tout HP envoyé en stockage provoque la diffusion périodique, depuis l'EPA du stock, d'AO pour l'exécution de l'HO suivant le concernant. Ainsi, la reprise de la fabrication des pièces stockées s'exécute sans l'occurrence d'un processus particulier, dans l'ordre des priorités des HO associés, puis des dates de stockage des HP concernés. La durée de stockage augmente sensiblement mais progressivement la priorité, ce qui permet de sortir toutes les pièces du stock. Ainsi, le HP 'pièce<sub>1</sub>' va être traité par le HR 'tour<sub>3</sub>' [4,1,4], puis être lavé (HR 'machine à laver') [4,1,6] et mesuré (HR 'machine à mesurer') [4,1,8].

Parallèlement à ce mécanisme, nous trouvons le mécanisme d'affectation des tâches de transfert par les HR 'robot' : toute tâche de transfert fait l'objet d'un AO émis par l'EPA du HR d'où doit être évacué le produit, une fois la destination connue. Cet AO : f(HP 'pièce<sub>i</sub>', HO 'transfert<sub>k</sub>') contient la description du HP dans son état intermédiaire, de son mode de préhension et des cycles de prise et de dépose... ainsi que la destination et la date de début probable de la tâche. C'est l'EPA du HR 'robot' répondant le mieux à cet AO qui remporte l'AO et le HR associé qui effectue la tâche. Ainsi dans notre exemple de la figure 12, le HP 'pièce<sub>1</sub>' va être transféré par les HR 'robot<sub>1</sub>' et 'robot<sub>2</sub>', dans l'ordre HR 'r<sub>1</sub>' [4,1,1], HR 'r<sub>1</sub>' [4,1,3], HR 'r<sub>2</sub>' [4,1,3bis] (sortie du stock), HR 'r<sub>1</sub>' [4,1,5], HR 'r<sub>1</sub>' [4,1,7] et HR 'r<sub>2</sub>' [4,1,9].

L'ensemble des mécanismes de décision que nous venons de décrire a été obtenu sans prise de décision prévisionnelle, sans centre de décision centralisé et sans intervention humaine. Le HR 'cellule de tournage' est devenu un système de production totalement autonome, qui, à l'instar d'un holon, autogère totalement son propre fonctionnement. Cette caractéristique supplémentaire vis-à-vis d'un système de production classique lui confère l'attribut de mécatronique, au sens où il y a augmentation des fonctionnalités et intégration d'intelligence donnant l'autonomie.

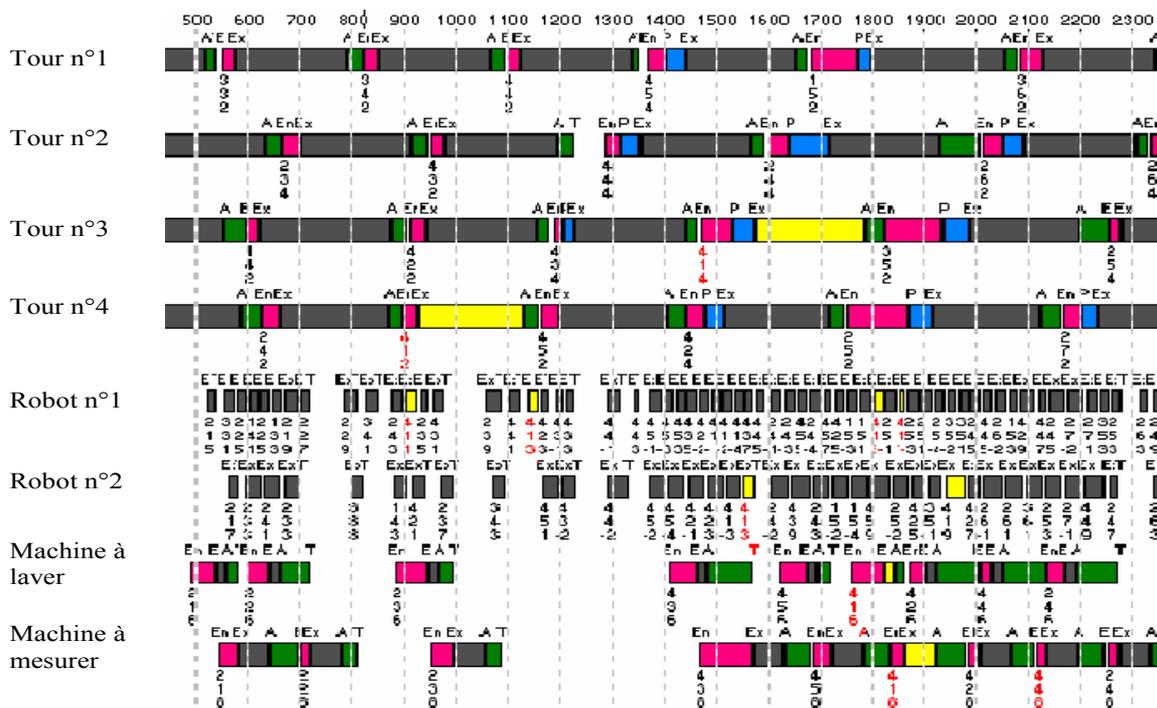


Figure 12. Gantt de suivi des opérations dans le HR 'cellule mécatronique de tournage', suite à l'introduction d'un HO 'lot urgent'

## 5.2. Autres applications

Nous abordons ici quelques exemples illustrant un éventail de problématiques traitées par des *HR* 'système mécatronique xx', en positionnement des éléments du modèle de référence PROSA et la définition de l'évaluation de performance.

### 5.2.1. Pilotage auto-organisé d'une flotte de robots mobiles

Un *HR* 'système de transport mécatronique' est constitué de *HR* 'AGV' qui vont auto-organiser les tâches de transport entre des *HR* 'postes de travail' qui lancent les différents AO correspondant à leurs besoins (Broissin *et al.*, 1996). Chaque *HR* 'AGV' va chercher à optimiser son déplacement pour répondre à temps à la demande et terminer la tâche le plus rapidement possible. Pour cela, il dispose d'un modèle de tous les cheminements possibles au sein de l'atelier, sous forme d'un graphe. Ce graphe est doté d'informations sur l'occupation temporelle de chaque segment de chemin par les *HR* 'AGV' effectuant des AO déjà affectés. Ceci lui permet de calculer le meilleur chemin possible, et de calculer la date de livraison effective qui constitue l'expression de la performance : le *HR* 'AGV' le plus rapide se voit l'AO affecté et les occupations prévisionnelles de chemin sont actualisées.

### 5.2.2. Pilotage auto-organisé d'une gare robotisée de triage

Un *HR* 'Hub de transbordement robotisé' accueille des trains porte-conteneurs venant de différentes origines et repartant vers ces mêmes destinations. Il est constitué de *m* *HR* 'robot portique' et de *p* *HR* 'robot navette' qui vont auto-organiser les tâches de transbordement des *HP* 'conteneurs' entre les trains. Chaque *HP* 'conteneur' doit être placé sur le train en partance pour la bonne destination. Ce qui est intéressant dans cette application, c'est que les *HR* 'robot portique' cherchent à résoudre localement le problème de transbordement, puis, pour tous les transferts non réalisables, font appel à d'autres *HR* 'robot portique' et aux *HR* 'robot navette' pour trouver d'autres solutions.

Ce mécanisme de résolution en deux phases décisionnelles est totalement décrit dans (Pujo *et al.*, 2001).

### 5.2.3. Pilotage auto-organisé d'un parking robotisé

Un *HR* 'parking robotisé' est composé de *HR* 'sas d'entrée-sortie', où des *HP* 'véhicule' sont laissés ou récupérés par les clients et de *n* *HR* 'robot mobile' qui les transfèrent entre des alvéoles de stockage. Ces alvéoles étant réparties sur plusieurs étages, les *HR* 'robot mobile' empruntent des *HR* 'robot ascenseur'. Une description de ce système est donnée dans (Pujo *et al.*, 2005). Là encore, nous avons deux phases décisionnelles : des *HR* 'robot mobile' se distribuent les tâches de transport des *HR* 'véhicule', en recherchant les chemins les plus rapides possibles pour offrir la meilleure qualité de service possible aux clients, et des *HR* 'robot ascenseur', qui sont au service des précédents pour assurer les changements d'étage, selon les techniques d'analyse et de recherche de chemins évoquées en section 5.2.1. Les *HO* correspondant aux restitutions de *HP* 'véhicule' aux clients sont prioritaires. Ils sont lancés dans l'ordre d'arrivée des clients. Cela n'implique pas que les véhicules soient restitués dans cet ordre : tout dépend de l'alvéole où il faut aller chercher le véhicule et des positions relatives des *HR* 'robot mobile' et 'robot ascenseur' dans le parking. Pour les stockages, si le taux d'utilisation du parking est élevé, les solutions de stockage les plus rapides seront privilégiées, alors que si la charge est faible, des solutions répartissant les véhicules dans le parking seront préférables.

## 6. Conclusion et perspectives

Nous avons présenté dans ce papier comment le paradigme holonique permet d'aborder le pilotage de systèmes mécatroniques complexes composés de ressources elles-mêmes mécatroniques. Les différents aspects d'un système 'mécatronisé' étant pris en compte par les trois types de holons, nous montrons comment auto-organiser le pilotage et obtenir ainsi des systèmes mécatroniques complexes plus autonomes, car plus performants en termes de réactivité et de robustesse. Après avoir rappelé les fondements du concept de holon et exposé les apports de l'approche holonique vis-à-vis d'une approche système habituelle, nous avons détaillé une Entité de Pilotage Autonome insérée dans chaque constituant de la holarchie de ressources d'un système mécatronique complexe. Ensuite, nous avons notamment détaillé le mécanisme d'interaction isoarchique entre EPA, à l'aide d'un modèle formel en DEVS. Les différentes fonctions collectives et interactives de décision qu'un système de pilotage auto-organisé doit assumer dans un système mécatronique complexe peuvent être classées en deux types, correspondant aux deux faces du holon ressource. D'une part, la face externe traite de la participation du holon ressource au fonctionnement autonome du système mécatronique complexe dans son environnement. D'autre part, la face interne traite de la participation des ressources internes au système pour assurer l'autonomie de son fonctionnement. Ce concept récursif permet à la fois l'expression des propriétés du système qu'il représente et celles du

système auquel il appartient. Nous avons énuméré une large variété de systèmes mécatroniques complexes concernés par la mise en pratique de cette approche, allant des systèmes flexibles de production composés de machines de production à architecture mécatronique, elles-mêmes composées d'opérateurs de mouvement. Les problèmes traités visent l'émergence de solutions lors de problèmes de coopération, de génération, de synchronisation et de coordination. Pour illustrer l'efficacité du pilotage auto-organisé et isoarchique face au critère fondamental de la réactivité dans un système complexe, nous avons montré en détail le fonctionnement réactif d'une cellule mécatronique de tournage, lors d'une perturbation extrêmement forte : l'introduction d'un lot urgent qui vient bouleverser totalement la production nominale. Les résultats sur cet exemple montrent que, sans aucune intervention extérieure (humaine ou non, hiérarchiquement de niveau supérieur ou pas...), l'ensemble des machines 'mécatroniques' composant la cellule peut s'auto-organiser pour résoudre le problème posé. Nous sommes bien face à une intelligence collective qui est fondée sur l'intelligence des entités individuelles qui composent le collectif. En cela, chacune des entités individuelles, s'il est une machine automatisée, porte un caractère mécatronique, et l'ensemble du collectif également. Et c'est cela qui amène les propriétés de réactivité et de robustesse que nous constatons dans nos expérimentations.

L'approche proposée est fondée sur une base commune de mécanismes standard de décision, modélisés de manière formelle, communs à tous les niveaux de décision et indépendants de l'application. Pour rendre spécifique le système de pilotage à l'application, nous intégrons des connaissances liées au fonctionnement du système concerné dans certains modules internes de l'EPA. Nous trouvons là un aspect fondamental de la mécatronique, à savoir la nécessaire intégration pluri disciplinaire. L'intelligence d'un système mécatronique complexe ne provient plus seulement de l'intégration de primitives fonctionnelles issue de l'automatique et de l'informatique industrielle sur une base électromécanique, mais également de l'intégration dans les EPA de connaissances issues de domaines tels que la gamme automatique en usinage, la planification de chemin en robotique mobile, la modélisation des systèmes poly-articulés, le diagnostic automatique... La mise en œuvre de cette approche du pilotage auto-organisé et isoarchique à toute nouvelle application ouvre donc des perspectives de recherche quant à la compréhension de l'application et à son intégration dans l'EPA pour conférer à de nouveaux systèmes complexes toute l'intelligence et l'autonomie requises à une qualification 'mécatronique'.

Ce modèle de pilotage auto-organisé peut s'appliquer à différents niveaux de décision, dès qu'il doit y avoir un choix entre différentes ressources en concurrence. Un autre volet de nos recherches sur cette approche du pilotage concerne par exemple les systèmes logistiques, au travers du pilotage auto-organisé des relations clients/fournisseurs (Ounnar et Pujo, 2005 ; Ounnar *et al.*, 2007).

## 7. Bibliographie

- Anselmetti B., Automatisation du tournage en commande numérique, Thèse de doctorat, Ecole Normale Supérieure de Cachan, 1985.
- Asgari B., Contribution à la prise en compte du comportement multitechnologique des robots industriels dans le contexte d'une architecture de commande décentralisée, Thèse de doctorat, Université d'Aix-Marseille III, 1998.
- Balasubramanian S., Brennan R. W., Norrie D. H., "An architecture for metamorphic control of holonic manufacturing systems", *Computers in Industry*, (46), 13-3, 2001.
- Ben Hmida F., Contribution à l'amélioration des axes asservis de Machines de Production, Thèse de doctorat, Université d'Aix-Marseille III, 1998.
- Bongaerts L., Monostori L., McFarlane D., Kadar B., "Hierarchy in distributed shop floor control", *Computers In Industry*, (43), 2000, p. 123-137.
- Broissin N., Pujo P., Bertrand J.C., "Decentralized Piloting of Transport Systems by Autonomous Vehicles", *Actes de WAC96*, Montpellier, France, 1996.
- Broissin N., Contribution à l'amélioration de la réactivité des systèmes de production automatisés et flexibles grâce à un pilotage basé sur une génération de tâches décentralisée, Thèse de doctorat, Université d'Aix-Marseille III, 1999.
- Brun Picard D., Etablissement d'un modèle pour la commande décentralisée, Thèse de doctorat es sciences, Université d'Aix-Marseille III, 1988.
- Carmona J.C., Contribution à l'étude et à la réalisation d'un système de commande décentralisée d'un tour industriel, Thèse de doctorat, Université d'Aix-Marseille III, 1990.
- Châne F., Giambiasi N., Paillet J.L., "Formal specification of reactive systems: DEVS or timed automata?", *Actes de CMS'2004, I3M*, Bergeggi, Italy, 2004.
- Deen, S.M., *Agent-Based Manufacturing – Advances in the Holonic Approach*, Springer-Verlag Ed, 2003.
- FIPA, Foundation for Intelligent Physical Agents, web site, <http://www.fipa.org/>, 2003.

- IEEE P1516, "Draft Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules".
- Ken Keys K., Hirschfeld R., Desai P. B., Sastry A., "Intelligent manufacturing systems (Smart Mechatronics & CIM) at Louisiana State University", *IFAC Mechatronics*, vol. 5, n° 7, 1995, p. 743-752.
- Koestler A., *The Ghost in the Machine*, London, Hutchinson, 1967.
- Lei M., Yang X., Tseng M.M., Yang S., "Design an intelligent machine center - strategy and practice", *IFAC Mechatronics*, vol. 8, n° 3, 1998, p. 271-285.
- Leitão P., Restivo F., "ADACOR: A holonic architecture for agile and adaptive manufacturing control", *Computers in Industry*, (57), 2006, p. 121-130.
- LeMoigne J.L., *La théorie du système général – théorie de la modélisation*, Paris, PUF, 1977.
- Massé G., Pujo P., Bertrand J.C., "Towards a mechatronics architecture of industrial robots", *Actes du 2<sup>e</sup> Congrès Franco-Japonais de Mécatronique*, Japon, 1994.
- Mathews J., "Organizational foundations of intelligent manufacturing systems – the holonic viewpoint", *Computer Integrated Manufacturing Systems*, vol. 8, n° 4, p 237-243, 1995.
- Maturana F., Shen W., Norrie D.H., "MetaMorph: An adaptive agent-based architecture for intelligent manufacturing", *International Journal of Production Research*, vol. 37, n° 10, 1999, p. 2159-2173.
- Michel P., Contribution à l'étude et à la réalisation de la commande décentralisée d'un axe de robot, Thèse de doctorat, Université d'Aix-Marseille III, 1988.
- Ounnar F., Pujo P., "Supplier evaluation process within a self-organized logistical network", *International Journal of Logistics Management*, vol. 16, n° 1, 2005, p. 159-172.
- Ounnar F., Pujo P., Mekaouche L., Giambiasi N., "Customers-Suppliers Relationship Management in an Intelligent Supply Chain", *Production, Planning and Control*, 2007.
- Pichler F., "On the Construction of A. Koestler's Holarchical networks", *Actes de EMCSR 2000*, April 25-28, Vienna, Austria, 2000.
- Pujo P., Contribution à la définition d'une modélisation technologique des robots industriels, Thèse de doctorat, Université d'Aix-Marseille III, 1994.
- Pujo P., Broissin N., Bertrand J.C., « Simplification de la programmation de Tâches dans les SAP par l'intégration de primitives « métier » dans les Machines de Production », *Actes du 5<sup>e</sup> Congrès International de Génie Industriel (GI'5)*, Grenoble, France, Avril 1995.
- Pujo P., Broissin N., Bertrand J.C., « Evaluation de règles de génération de tâches dans le contexte du pilotage décentralisé d'un Système de Production Automatisé et Flexible », *Actes du 2<sup>e</sup> Congrès Franco Québec de Génie Industriel*, Albi, Septembre 1996.
- Pujo P., Ounnar F., « Proposition d'un pilotage décentralisé, auto-organisé et rapproché pour système automatisé flexible : Application à un Hub de transbordement robotisé fer-fer », *Journal Européen des Systèmes Automatisés*, vol. 35, n° 7-8, 2001, p. 905-932.
- Pujo P., Ounnar F., Zanni C., "A self organized holonic control for mechatronics complex systems: application to a robotized car park", *International Journal of Automation Austria*, 2005, p. 117-128.
- Rodriguez S., Hilaire V., Koukam A., "Formal Specification of Holonic Multi-Agent Systems Framework", *Lecture Notes in Computer Science*, n° 3516 Springer, 2005, p.719-726.
- Smith R.G., "The contract net protocol: high level communication and control in a distributed problem solver", *IEEE Transactions on Computers*, C29, 1980, p. 1104-1113.
- Trentesaux D., Pilotage hétérarchique des systèmes de production, Mémoire d'Habilitation à Diriger des Recherches, Université de Valenciennes, 2002.
- Van Brussel H., Wyns J., Valkenaers P., Bongaerts L., Peeters P., "Reference architecture for holonic manufacturing systems: PROSA", *Computers in Industry*, (37), 1998, p. 255-274.
- Zeigler B., *Theory of Modelling and Simulation*, New York, Ed. John Wiley, 1973.
- Zeigler B., *Multifaceted Modelling and Discrete Event Simulation*, London, Academic Press, 1984.

Reçu le : 1er novembre 2005

Accepté après révisions le : 11 mai 2007