

# **Une base de connaissances pour la génération de modèles d'ateliers flexibles de production**

**Luiz Marcio SPINOSA, Eugène CHOURAQUI, Bernard ESPINASSE**

*DIAM- IUSPIM, Domaine Universitaire de Saint Jérôme,  
Avenue Escadrille Normandie Niemen, 13397 Marseille Cedex 20 - France  
(diam\_lms,diam\_ec,diam\_be)@vmesa11.u-3mrs.fr*

## **Résumé**

Cet article porte sur la présentation d'une *base de connaissances* et d'une *méthodologie* à partir desquelles des modèles spécifiques de représentation des Ateliers Flexibles de Production (AFP) peuvent être générés. Cette démarche est illustrée et validée à propos d'une application en gestion de production. Pour ce faire, nous avons fait appel à quatre approches complémentaires : (i) la Représentation de Connaissances concrétisée par le paradigme Orienté Objet (OO), (ii) les Réseaux de Petri (RdP), (iii) la Satisfaction de Contraintes et (iv) la Programmation Logique.

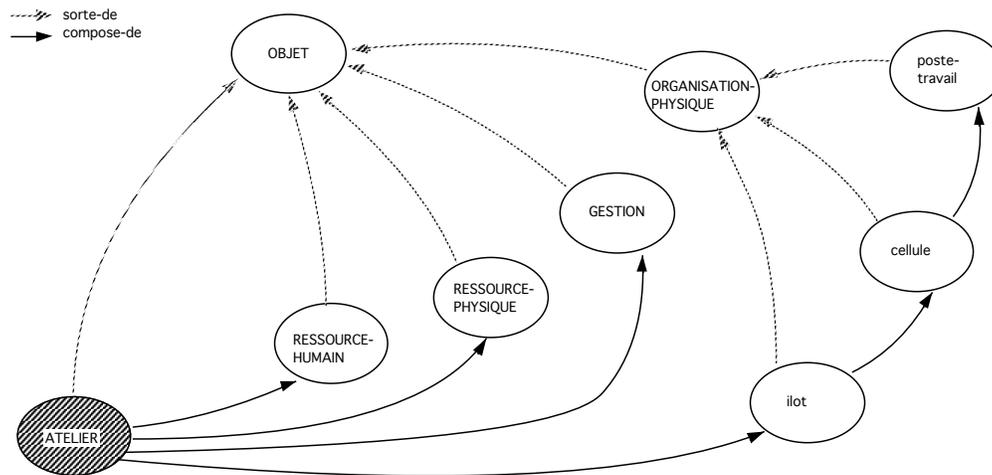
## **1. Introduction**

La modélisation des tâches des systèmes de production joue un rôle très important dans la démarche d'analyse de l'entreprise manufacturière. Elle permet de fonder le raisonnement sur une représentation cohérente du fonctionnement des entités opérationnelles. Si nous pouvons améliorer l'efficacité de ces entités, en réalisant des mesures sur un modèle représentatif, nous pouvons alors définir de façon intelligente de meilleures stratégies de gestion de production.

Le choix de l'approche de représentation OO repose sur le fait qu'au sein d'un système de production (ici les AFP), nous commandons toujours les mêmes types d'entités opératives (machines, transports, robots, etc). Il nous semble donc intéressant, vu leur nombre restreint, de les archiver au sein de bases de connaissances. Plusieurs auteurs ont adopté une telle démarche, parmi eux [Barbier 89][Fritschy 90][Amar 90][Arnoux 90]. Par contre cette approche présente actuellement de sérieuses limitations pour modéliser d'une part les caractéristiques de synchronisme et de parallélisme et d'autre part celles des opérations de mise en oeuvre des processus de fabrications soumises à des contraintes de temps. Il faut noter que certaines de ces caractéristiques peuvent être partiellement modélisées à l'aide, par exemple, de messages retardés (voir [Dugerdil 88]) ou des méthodes qui utilisent des techniques de communication et de synchronisation (voir [Dorseuil 91]). Mais dans les deux cas, il faut d'abord étendre les capacités logicielles supportant ces méthodes. Aussi, nous avons utilisé les RdP, plus spécifiquement les PNO (Petri nets with Objects formalism) afin de pallier ces insuffisances [Sibertin 91]. De plus, afin d'augmenter la capacité expressive d'un tel mariage et donc de diminuer encore les limitations observées ci-dessus, nous suggérons également l'utilisation des techniques liées aux Problèmes de Satisfaction de Contraintes.

L'articulation de ces approches appliquées aux connaissances liées à la productique conduit à la création d'une *base de connaissances* et au développement d'une *démarche méthodologique* de type

cognitive permettant de caractériser principalement le système physique de l'entreprise<sup>1</sup> (bien que nous considérons quelques concepts du système de décision).



```
(TRANSPORT*
* connaissances intrinsèques *
(sorte-de (valeur RESSOURCE-PHYSIQUE))
(vitesse (domaine REEL) (cardinalite (1,1))
(capacite (domaine integer) (cardinalite (2,2)
(unite produit/heure))
(quantite (domaine integer) (cardinalite (1,1)
(unite produit)) (si-ajout verifier-surcharge))
(taille (domaine integer) (cardinalite (1,1)
(unite moyen))
(etat (domaine {libre,occupe}) (default {libre})
(cardinalite (1,1)))
(type (domaine {lineaire-mono,lineaire-bi,boucle})
(default {lineaire-mono}) (cardinalite (1,1)))
* comportement (procédural) *
<surcharge>
si quantite est egale a capacite alors
signaler "Unite de transport surcharge" finsi
<calcul-taux-transfert>
calcul taux = total de produit transporte / total de temps utilise
<transport>
si quantite est plus petite que capacite alors en-transport = PRODUIT.en-transport
quantite = quantite + 1 finsi
<demande>
pendant que resultat soit vrai faire msg<destin,"demande-de-travail",resultat>
finpendant
eleve PRODUIT de la liste en-transport.PRODUIT
attache prise de PRODUIT au destin
<calcul-position>
cherche la position de PRODUIT sur la liste en-transport
(moyen (domaine {chariot,palette}) (default {palette})
(cardinalite (0,1)))
(position (domaine {ENTIER}) (cardinalite (1,1))
(si-besoin calcul-position(PRODUIT)))
(taux-transfert (domaine {REEL}) (cardinalite (1,1))
(si-besoin calcul-taux-transfert))
* connaissances extrinsèques *
(origine (domaine {POSTE-TRAVAIL})
(cardinalite (1,1)))
(destination (domaine {POSTE-TRAVAIL}) (cardinalite (1,1)))
(en-transport (domaine {list-ord<PRODUIT>}) (cardinalite (0,*))
(si-ajout quantite=quantite+1))
* comportement (réseaux de Petri) *
```

**Figure 1** - Schéma Partiel de la base de connaissances

<sup>1</sup> Nous suivons la définition donné par CIMOSA ([CIMOSA 91], pp. 1-3) pour le découpage d'un système de manufacture, à savoir : système de décision, système physique et système d'information.

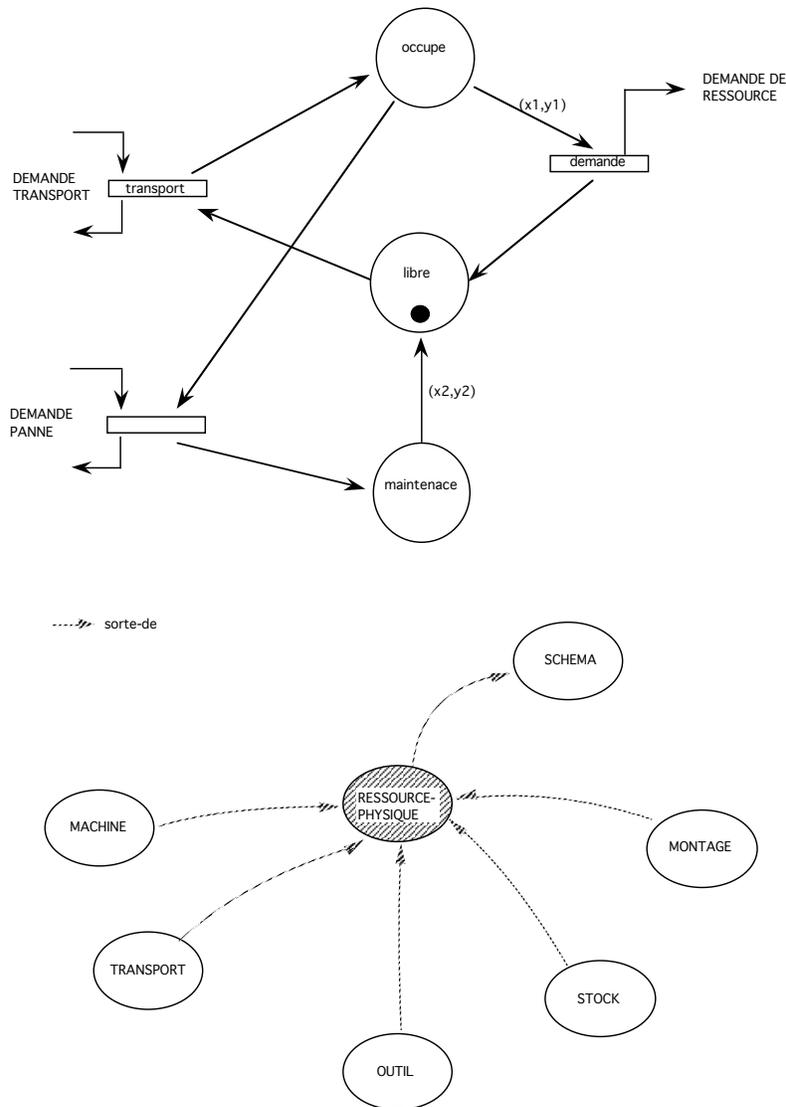


Figure 1 - Schéma Partiel de la base de connaissances (fin)

## 2. La base de connaissance

La base de connaissances est constituée de deux modèles principaux : le modèle de représentation des connaissances statiques et dynamiques des AFP et le modèle de représentation de contraintes intervenant dans la mise en oeuvre des AFP.

### 2.1. Le modèle de représentation des connaissances

La base de connaissances repose sur la description des éléments composants l'AFP. Elle est composée de 26 schémas type "objet" organisés dans une hiérarchie {méta-classe, classe, instance}. Un exemple d'objet de cette base de connaissance est le schéma TRANSPORT (Figure 1).

Chaque objet est composé de trois parties :

- **la représentation des connaissances intrinsèques** qui caractérisent un objet hors contexte, c'est-à-dire indépendamment de tout autre objet (par exemple l'objet TRANSPORT possède, entre autres, un attribut *type* qui désigne des caractéristiques dont les valeurs appartiennent à l'ensemble {linéaire-mono, linéaire-bi, boucle} ;

- **la représentation des connaissances extrinsèques** qui désignent les relations qu'entretient tout objet avec les autres objets de la base (par exemple l'objet TRANSPORT est décrit par plusieurs champs, dont les champs *origine* et *destination* qui établissent une relation avec les objets de la classe POSTE-DE-TRAVAIL, définissant ainsi que le transport sera effectué à partir d'un poste de travail défini par *origine* jusqu'au poste de travail repéré par *destination*). Les connaissances structurelles appartiennent à cette partie et permettent d'établir des relations hiérarchiques et la mise en oeuvre des mécanismes d'héritage (par exemple le champ *sorte-de* dans l'objet TRANSPORT spécifie que ce dernier est un type de RESSOURCE-PHYSIQUE, permettant au premier d'hériter tous les champs de ce dernier) ;
- **la représentation du comportement** désigne le savoir-faire de l'objet qui est représenté à l'aide de clauses PROLOG et/ou de procédures et/ou de segments de RdP. Ces éléments sont normalement déclenchés (i) soit par des attributs dits dynamiques décrits dans la connaissance, (ii) soit au travers de l'envoi de messages par d'autres objets (iii), soit enfin par les RdP.

## 2.2. Le modèle de représentation de contraintes

Nous avons construit un modèle complémentaire, grâce à une modélisation fondée également sur la connaissance, permettant la spécification de Problèmes de Satisfaction de Contraintes (PSC). Ce modèle a été intégré à la base de connaissances décrite ci-dessus afin d'atteindre deux buts essentiels : (i) augmenter la capacité expressive de la base à travers la formulation de PSC (cf. III.) et (ii) offrir un moyen supplémentaire plus puissant pour la construction des applications (cf. IV ).

Tout PSC  $p$  est défini par le 4-uple  $(x,d,c,r)$  de la manière suivante [Jegou 91] :

- un ensemble de variables  $x = \{X_1, \dots, X_n\}$  ;
- un ensemble de domaines finis et discrets  $d = \{D_1, \dots, D_n\}$  où  $D_i$  est le domaine associé à  $X_i$  ;
- un ensemble de contraintes  $c = \{C_1, \dots, C_m\}$  tel que  $C_i$  est défini par l'ensemble  $\{X_{i_1}, \dots, X_{i_j}\}$  ;
- une application  $r$  :

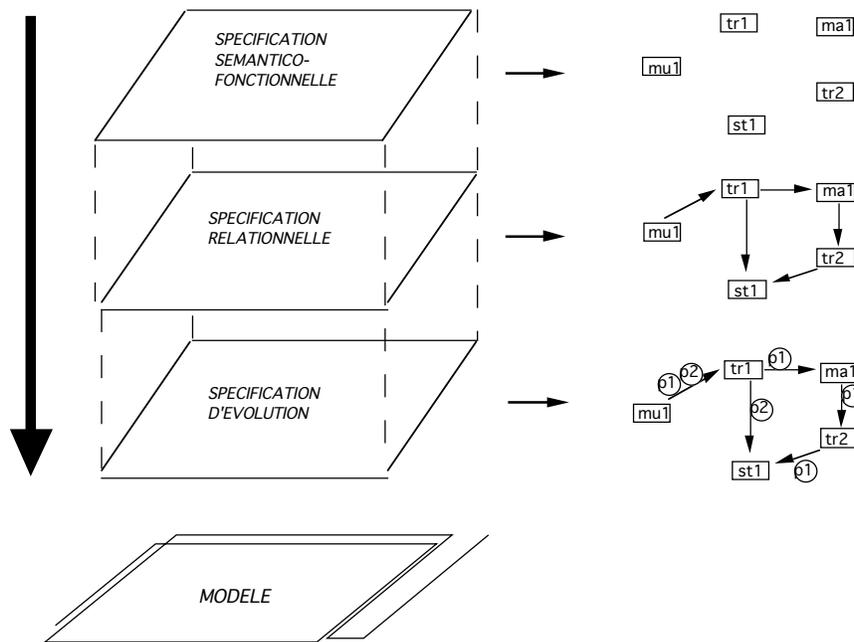
$$\left. \begin{array}{l} C - \bigcup_{C_i \in C} (\bigcup_{X_j \in C_i} D_j)^C \\ C_i \rightarrow (\bigcup_{X_j \in C_i} D_j)^C \\ \text{telle que } \forall r_i \in R(C_i), \forall X_j \in C_i, r_i(X_j) \in D_j \end{array} \right\}$$

Un *étiquetage consistant* est un n-uplet qui vérifie chacune des contraintes de  $c$ , c'est-à-dire que dans l'ensemble des solutions  $\{(X_{i1}, \dots, X_{in}), \dots, (X_{j1}, \dots, X_{jn})\}$  qui satisfont  $c$  on sélectionne un n-uplet  $(X_{i1}, \dots, X_{in})$ , valeurs respectives de  $\{X_1, \dots, X_n\}$ . Résoudre un PSC consiste à chercher l'*étiquetage consistant*., et différentes approches et méthodes ont été élaborées en vue de le calculer.

Les concepts liés aux PSC et leur résolution ont été représentés selon une approche orientée objet [Borderie 94].

## 3. La génération de modèles

Une démarche composée de trois phases a été définie en vue d'exploiter la *base de connaissances*. L'articulation de ces trois phases est reconnue comme étant un *processus de prototypage* et le modèle obtenu comme étant une *instance* des modèles prototypiques (Figure 2).



**Figure 2** - Processus de prototypage

Cette démarche se décompose comme suit :

- **la spécification sémantico-fonctionnelle** : elle permet d'obtenir un modèle de type structurel mettant en évidence les différents organes élémentaires constituant l'AFP. Cette étape s'articule principalement autour de la description des connaissances intrinsèques et du comportement des objets (cf. II.), et de l'instanciation des objets constitutifs de l'AFP.
- **la spécification relationnelle** : le modèle défini ci-dessus est étendu à la description des connaissances extrinsèques des objets (cf. II.), mettant en évidence une sémantique relationnelle basée en particulier sur la notion de partage de connaissances.
- **la spécification d'évolution** : elle représente la dynamique du modèle à travers la construction d'un graphe (RdP) représentant le comportement de l'AFP décrit par les différentes possibilités de déplacement d'un produit (cf. II.). La notion de déplacement est liée directement aux concepts des unités de transport, et plus particulièrement aux champs *origine et destination*. Ces champs permettent d'établir une liaison physique aussi bien qu'une liaison sémantique. Cette dernière rend possible la construction du graphe de déplacement.

Les processus de prototypage proposés permettent de construire le modèle informatique d'un atelier. Du fait de la forte influence que les éléments composant le modèle peuvent avoir entre eux, plusieurs contraintes doivent être imposées. Pour ce faire, des PSC ont été construits mettant en rapport ces différents éléments, ce qui augmente encore la robustesse du modèle.

Une fois le modèle instancié, l'AFP a une représentation abstraite, régulière et stable qui permettra la construction d'applications spécifiques.

## 4. Exemple d'application

L'application présentée ici est un exemple d'école, donc nécessairement incomplet, qui a pour objet essentiel d'illustrer la démarche décrite ci-dessus. Elle concerne d'une part la construction d'une cellule de fabrication représentant un des organes productifs d'un AFP (cf. IV.1) et d'autre part la construction d'un système fondé sur un raisonnement par contraintes (cf. IV.2).

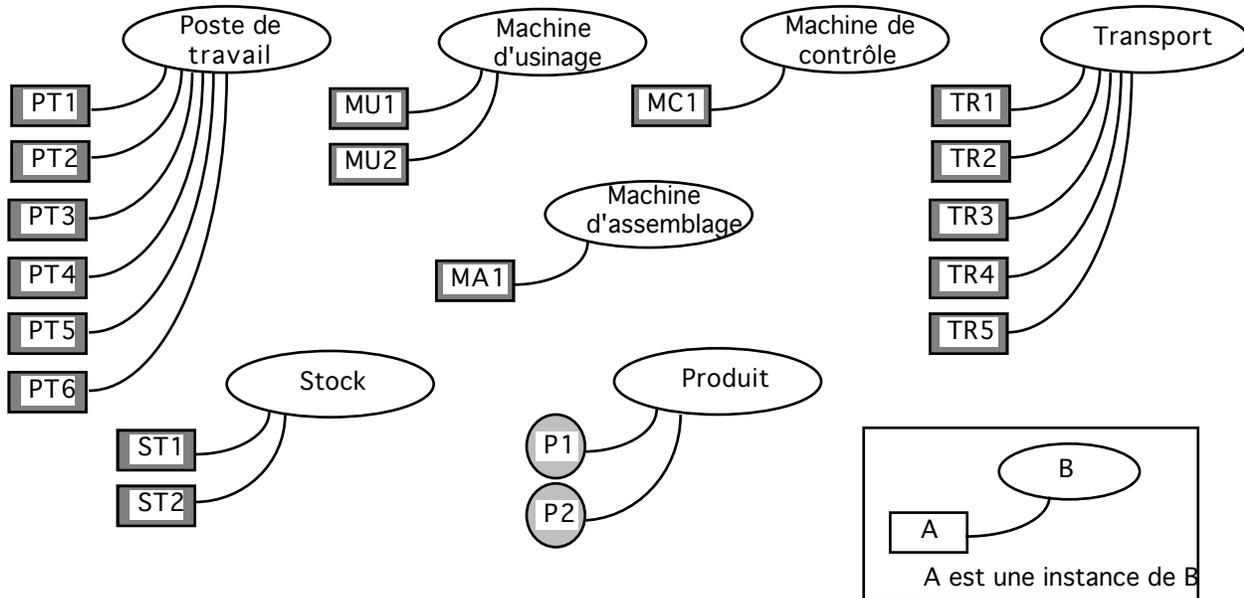
Nous supposons que l'AFP est composé : (i) de ressources physiques MU1 et MU2 (machines d'usinage), MA1 (machine d'assemblage), ST1 et ST2 (stocks), MC1 (machines de contrôle), TR1, ..., TR5 (organes de transport), (ii) des opérations mobilisées par ces ressources, (iii) des ressources

humaines, (iv) d'entités physiques organisées PT1, ..., PT6 (postes de travail), (v) de méthodes de gestion, et enfin (vi) des produits P1 et P2.

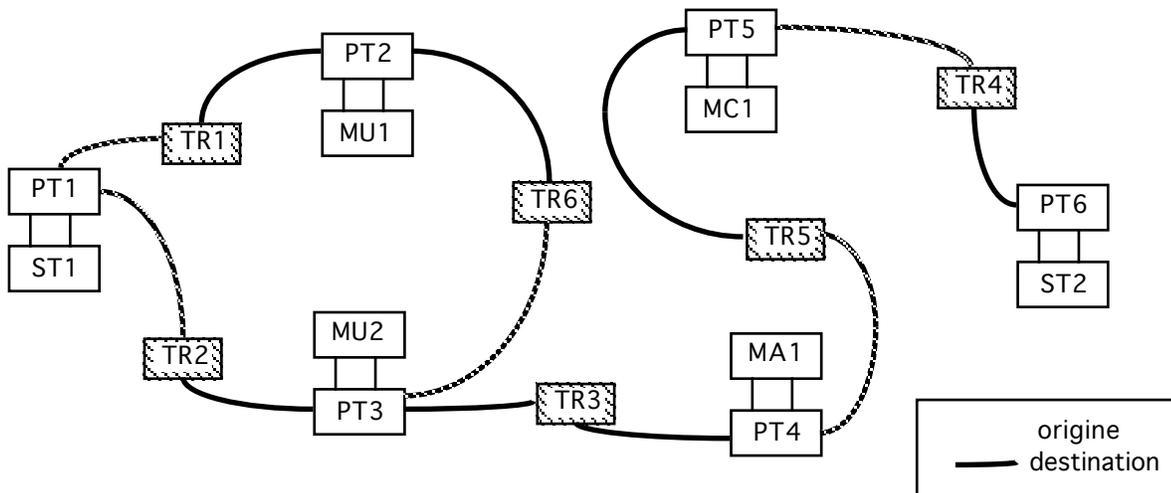
A titre de simplification seuls les schémas les plus significatifs correspondant aux points (i) et (vi) seront décrits, sachant que les autres points suivent la même méthodologie d'analyse et de représentation.

#### 4.1. Le processus de prototypage

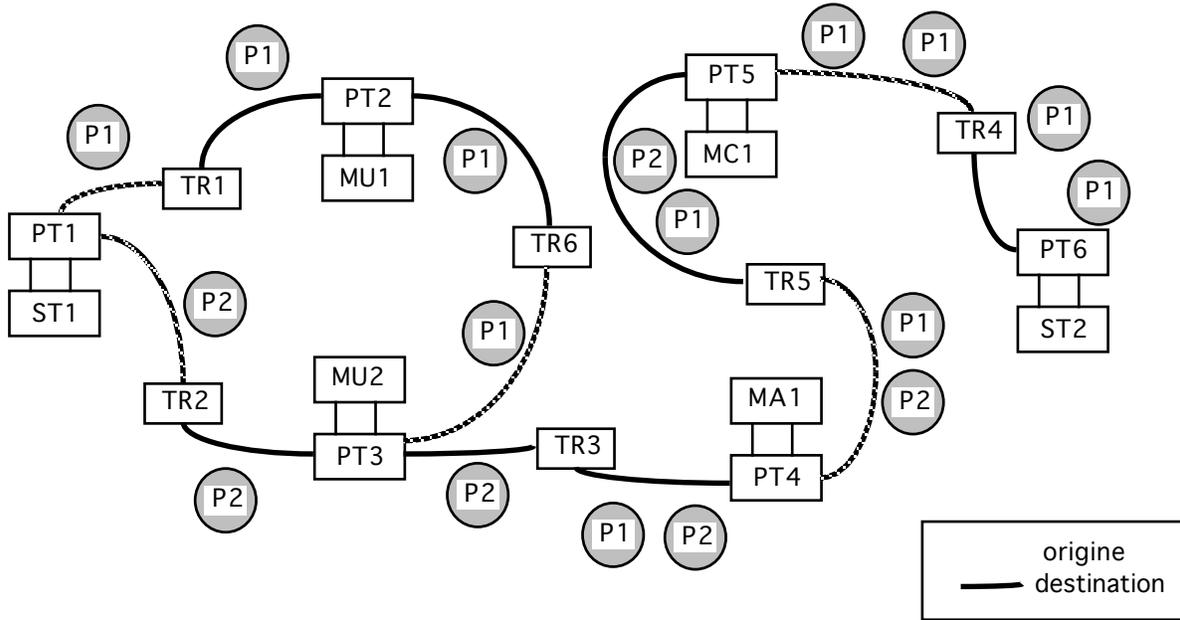
**Spécification sémantico-fonctionnelle** : l'objectif de cette phase est d'élaborer un catalogue raisonné des éléments mobilisés par l'AFP. Ces éléments sont caractérisés de manière spécifique à partir des objets contenus dans la base de connaissances grâce au processus d'instanciation.



**Spécification relationnelle** : les connaissances relationnelles entre les machines que composent les postes de travail, ainsi que les liaisons que ces postes entretiennent avec les unités de transports sont modélisées et agrégées aux éléments créés précédemment. Aux termes de cette phase nous obtenons le graphe représentant les possibilités de déplacement des produits et le modèle comportemental associé grâce à la prise en compte des segments des RdP décrivant les postes de travail ( $PT_n$ ) et les unités de transport ( $TR_m$ ).



**Spécification d'évolution** : nous définissons les chemins que doivent emprunter les produits au cours du processus de leur fabrication. Ces chemins sont établis grâce à la gamme de fabrication spécifique de chaque produit.



**Satisfaction de contraintes** : les PSC doivent être définis selon l'objectif de l'application. Dans le présent exemple nous donnons quelques éléments de définition d'un PSC qui exprime la cohérence entre la gestion commerciale et le système de production (cohérence entre les décisions de production et les capacités de production de stockage et de transport du système).

L'ensemble de variables  $x^5$  est :

- $\{X1 = \text{objectif-production.GESTION-PRODUCTION},$
- $X2 = \text{production-actuel.GESTION-PRODUCTION},$
- $X3i = \text{capacite.STOCK de l'instance } i,$
- $X4i = \text{capacite-moyenne.STOCK de l'instance } i,$
- $X5j = \text{capacite.TRANSPORT de l'instance } j,$
- $X6j = \text{capacite-actuel.TRANSPORT de l'instance } j,$
- $X7k = \text{capacite-production.MACHINE de l'instance } k\}$

Les domaines sont définis comme suit :

- $\{D1 = [0, f1]$  où  $f1$  est défini en fonction de la demande CARNET-COMMANDE,
- $D2 = [0, f2]$  où  $f2$  est défini en fonction de la capacité maximale du système,
- $D3i = [0, f3]$  où  $f3$  est défini en fonction de la capacité maximale de chaque instance  $i$ ,
- $D4i = [0, f4]$  où  $f4 = f3$ ,
- $D5j = [0, f5]$  où  $f5$  est défini en fonction de la capacité maximale de l'unité de transport  $j$ ,
- $D6j = [0, f6]$  où  $f6 = f5$ ,
- $D7k = [0, f7]$  où  $f7$  est défini en fonction de la capacité maximale de la machine  $k$  }

Les contraintes sont les suivantes :

- $\{C1 = ( \sum_{k=1}^n X7k ) / n \geq X1,$
- $C2 = ( \sum_{i=1}^m X3i ) / m \geq X1,$
- $C3 = ( \sum_{j=1}^k X5j ) / k \geq X1,$
- $C4 = \gamma \geq [X1 - X2],$
- $C5 = Xk \text{ amont} \geq X5j , C6 = X7k \text{ aval} \geq X5j ,$
- $C7 = X6k \text{ aval} \cong X4i \cong X6j + 1 \text{ amont} \}$

Les contraintes C1, C2 et C3 imposent que l'objectif de production (*objectif-production*) doit observer la capacité maximale du système de production, celle-ci étant respectivement exprimée par les capacités maximales de production des machines, les capacités maximales de stockage et les capacités

maximales de transport. Nous pouvons vérifier que ces contraintes sont basées sur les valeurs moyennes des sommes des capacités des ressources physiques sans considérer les chemins de déplacement des produits pendant la fabrication. Ces contraintes sont donc interprétées comme des contraintes de conseil (heuristiques) pouvant être mises en cause pour la résolution du PSC d'ordonnement.

La contrainte C4 impose que la production journalière (*production-actuelle*) doit être proche de l'objectif de production établi. Nous considérons que cette différence a un coefficient de variation  $\gamma$  d'acceptabilité. Le coefficient  $\gamma$  est défini par la gestion de l'entreprise par rapport au type du produit, aux caractéristiques du système de production, aux stratégies de production, etc.

Les contraintes C5 et C6 sont complémentaires et imposent que le système de transport lié à une machine observe sa capacité de production (*capacité-production*). L'unité de transport *amont* ne doit donc pas fournir à la machine une quantité d'éléments supérieure à sa capacité de production ; de façon complémentaire, l'unité de transport *aval* doit être prête à recevoir les éléments dès qu'ils sont produits par la machine.

La contrainte C7 impose qu'un système de transport (unités *origine* et *destination*) lié à un stock doit observer la capacité moyenne de stockage de façon à permettre un flux approximativement constant. Nous considérons que cette relation doit être approximative compte-tenu de la propriété d'amortissement de flux que possèdent les stocks.

A la fin de ces quatre étapes nous avons un modèle représentatif de l'AFP. Ce modèle permet de construire des applications effectives ; l'exemple présenté ici contribue à traiter les problèmes d'ordonnement.

## 4.2. Le PSC d'ordonnement

Le problème d'ordonnement concerne la génération d'une séquence d'exécution de tâches représentatives du processus de fabrication d'un produit, plus précisément l'affectation des tâches aux machines (ou vice-versa). Les différents critères à optimiser sont généralement la durée totale de l'ordonnement, le respect des délais et la minimisation du coût.

Les éléments qui normalement caractérisent l'ordonnement sont : (i) les délais d'exécution des tâches : un délai peut être représenté par la composition d'une date de début et d'une durée de la tâche dont les bornes sont la date au plus tôt et la date au plus tard ; (ii) l'ordre de précédences entre tâches, une conséquence naturelle du processus de fabrication ; et enfin (iii) la disponibilité des ressources tout en respectant (i) et (ii).

Le PSC décrivant l'ordonnement est le suivant :

L'ensemble de variables  $x$  : est :

$$\{ \begin{array}{l} X1i = \text{date-début.PRODUIT de l'instance } i, \\ X2i = \text{durée.PRODUIT de l'instance } i, \\ X3i = \text{date-delai.PRODUIT de l'instance } i, \\ X4j = \text{date-disponibilité.MACHINE de l'instance } j \end{array} \}$$

Les domaines sont définis comme suit :

$$\{ \begin{array}{l} D1i = [0, n] , \\ D2i = [n] , \\ D3i = [0, n] , \\ D4i = [0, n] ; \text{ où } n \text{ est un nombre entier et fini du point de vue informatique} \end{array} \}$$

Les contraintes sont les suivantes :

$$\{ \begin{array}{l} C1 = X3i \geq X1i + X2i , \\ C2 = X1i \geq X4j, \end{array} \}$$

$$C3 = X_{i+1} \geq X_i + X_{2i}$$

La contrainte C1 signifie que l'exécution d'une tâche *i* sur une ressource doit être réalisée avant la date délai. La contrainte C2 impose que la ressource *j* responsable de l'exécution d'une tâche *i* doit être disponible avant sa date délai. La contrainte C3 impose que l'exécution d'une tâche *i* doit être exécutée avant le début de la tâche *i+1* qui suit.

Normalement la résolution du problème d'ordonnancement et de cohérence est traitée de façon conjointe. A travers la résolution du PSC d'ordonnancement, nous pouvons obtenir la séquence d'exécution de tâches décrivant le processus de fabrication d'un produit et par conséquent la liste ordonnée des ressources nécessaires. De ce fait, nous pouvons vérifier si la structure de l'atelier est bien adaptée à l'exécution de cette séquence à travers la résolution du PSC de cohérence. Cette influence peut être vérifiée principalement sur les contraintes C1, C2 et C3 du PSC de cohérence (cf. III).

Inversement, connaissant la résolution du PSC de cohérence nous pouvons vérifier si la liste obtenue par le PSC d'ordonnancement est cohérente avec le modèle de l'atelier, c'est-à-dire si les ressources spécifiées par la séquence vérifient les lignes de fabrication existantes.

En résumé, le PSC de cohérence doit être exécuté pendant le processus de prototypage de l'atelier puis dans un deuxième temps, le PSC d'ordonnancement de chaque produit est exécuté. Nous retournons à l'exécution du PSC de cohérence en spécifiant C1, C2 et C3. En itérant successivement sur ces deux PSC nous pouvons avoir un affinement substantiel du modèle.

## 5. Conclusion

Les éléments présentés ici ont été implémentés à l'aide d'OBJLOG, un langage de programmation de connaissances orienté objet ([Faucher 91][Dugerdil 88]). L'environnement ainsi constitué peut être présenté, de façon simplifiée, sous la forme d'une architecture composée : (i) d'un module comportant la *base de connaissances* pour la génération de l'AFP, (ii) d'un module dédié à l'interaction avec l'utilisateur, qui s'organise autour de la *démarche méthodologique*, et enfin (iii) d'un "monde" obtenu à partir du processus de prototypage, selon les spécifications d'une application particulière, contenant d'une part le modèle AFP instancié et d'autre part les schémas et méthodes spécifiques développés par l'utilisateur.

Ce travail s'inscrit dans le cadre d'une recherche plus vaste relative à l'élaboration d'une méthodologie pour la génération de modèles complexes d'AFP ainsi que la construction de l'atelier logiciel supportant cette méthodologie [Spinosa &al. 95].

## Références

- [Amar 90] Amar, S., "Modélisation des moyens de production par langages orientés objets en vue de la conception de la commande d'un système de production flexible", 1990.
- [Arnoux 90] Arnoux, M., "Programmation Orientée Objet et Systèmes Multi-Agents : Application en Robotique et en Productique". Thèse présentée à l'Université de Nice et Sophia-Antipolis, 1990.
- [Barbier 89] Barbier, F., Haurat, A., "Glossaire des systèmes à objets : Application à l'informatisation de la fonction production d'une entreprise manufacturière". Génie Logiciel & Systèmes Experts. N° 17, 1989.
- [Baumgartner &al.91] Baumgartner, H., Knischewski, K., Wieding., H. "CIM. Proposition pour une mise en oeuvre de la productique", Siemens, 1991.
- [Borderie 94] Borderie, D. "Problème de Satisfaction de Contraintes. Contribution à la modélisation et l'implémentation des méthodes de résolution fondées sur les objets". Mémoire de DEA de l'Université d'Aix-Marseille III, 1994.
- [CIMOSA 94] "CIMOSA- Open System Architecture for CIM". Technical Baseline. Version 3.0, october 94. CIMOSA association (e.V.), pp. 1-3.
- [Dorseuil 91] Dorseuil, A., "Le Temps Réel en Milieu Industriel - concepts, environnements, multitâches". ed. Bordas Paris, 1991.
- [Dugerdil 88] Dugerdil, P., "Contribution à l'Etude de la Représentation des Connaissances Fondée sur les Objets - Le Langage Objlog". Thèse présentée à l'Université des Sciences de Luminy, 1988.

- [Faucher 91] Faucher, C., "Elaboration d'un Langage Extensible Fondé sur les Schémas - Le Langage Objlog+ ".Thèse présentée à l'Université de Droit, d'Economie et des Sciences d'Aix-Marseille, 1991.
- [Ferber 90] Ferber, J. "Conception et Programmation par Objets" Collection Technologies de pointe, Informatique, ed. Hermès, 1990.
- [Fritschy 90] Fritschy, D., "La simulation orientée objet : nouvelles exigences en matière de simulation informatique et temporelle des flux d'un système flexible et discontinu de production". Colloque International de Productique, ed. Teknea, Bordeaux, 1990.
- [Jegou 91] Jegou, P., "Contribution à l'Etude des Problèmes de Satisfaction de Contraintes : Algorithmes de Propagation et de Résolution - Propagation de Contraintes dans les Réseaux Dynamiques".\_Thèse présentée à l'Université des Sciences et Techniques du Languedoc, 1991.\_
- [Masini &al.89] Masini, G., Napoli, A., Colnet, D., Leonard, D., Tombre, K., "Les langages à objets". ed. InterEditions , 1989.
- [Sibertin 91] Sibertin-Blanc, C., Bastide, R., "Object-Oriented Structuring using High-Level Petri Nets" . Advances in Petri Nets'91, ed. Rozenberg. 1991.
- [Spinosa 92] Spinosa, L.M., "Contribution à la modélisation des systèmes de production fondée sur l'approche orientée objet et sur les réseaux de petri". Memoire de DEA de l'Université d'Aix-Marseille III, 1992.
- [Spinosa &al. 95] Spinosa, L.M., Chouraqui, E., Espinasse, B., "Distributed CIM and DAI : For a knowledge and multiagent oriented approach for models development". IEEE International Conference on Systems, Man and Cybernetics, october 1995, à paraître.
-