



**HAL**  
open science

## Modélisation intégrée et pérennisation des connaissances dans une approche PLM

Philippe Pernelle, Arnaud Lefebvre

### ► To cite this version:

Philippe Pernelle, Arnaud Lefebvre. Modélisation intégrée et pérennisation des connaissances dans une approche PLM. Ingénierie de la conception et cycle de vie des produits, Hermes sciences, pp.371-394, 2006. hal-00261828

**HAL Id: hal-00261828**

**<https://hal.science/hal-00261828>**

Submitted on 10 Mar 2010

**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.

---

# Modélisation intégrée et pérennisation des connaissances dans une approche PLM

**Philippe Pernelle, Arnaud Lefebvre**

Laboratoire PRISMA

*IUT B - Université Claude Bernard – Lyon I*  
*17 rue de France*  
*69627 Villeurbanne*

## **1.Introduction**

De nombreuses entreprises industrielles s'intéressent depuis déjà quelques années aux systèmes d'information capables de prendre en charge la totalité du processus de développement du produit. Ces systèmes, généralement regroupés sous le terme de PLM (Product Lifecycle Management) [CIM 01] ou SIP [GZA 00] (Système d'Information Produit), étendent le champ d'application initial des Systèmes de Gestion de Données Technique (ou PDM). Les évolutions technologiques et méthodologiques rendent plus attractif le choix stratégique de centrer son SI autour d'un PLM. En effet, les éditeurs de PDM ont fiabilisé leur plateforme logiciel et le délai de déploiement d'un projet PDM s'est considérablement réduit grâce notamment à des environnements pré-configurés. Concernant la modélisation au sein de ses systèmes, on trouve une littérature abondante [GZA 00] [PER 02] [BOC 04] [SUD 05] [IPP 05]. Malgré cela, les retours sur investissement en terme de réduction des délais ou de maîtrise des coûts restent difficilement mesurables. De plus, il subsiste de nombreux problèmes en amont et en aval du déploiement de tels systèmes (granularité des modèles Produit, évolution dans le temps des modèles, niveau de détails des workflow, organisation à mettre en place, etc.). L'objet de ce chapitre est de proposer un cadre de modélisation intégré afin de répondre aux différents problèmes de conception d'un système PLM qui puisse garantir une réelle pérennisation du savoir-faire industriel des entreprises.

### ***Les besoins d'intégration***

Avec la concentration du marché des éditeurs et la volonté de baisse des coûts informatiques, peu d'entreprises se lancent dans la conception ad hoc d'un système PLM. Or, les méthodes de conception des systèmes d'information [COA 92] [BOO 94] [ROL 01] [GIR 01] [RAL 01] proposent généralement une approche

## 2 Ingénierie de la conception et cycle de vie du produit

globale pour concevoir son propre SI. De fait, les méthodes génériques ne sont pas adaptées à la configuration des plates-formes PDM existantes. A ces difficultés, s'ajoutent les problèmes d'hétérogénéité du SI lorsque celui-ci est constitué par de multiples outils spécialisés (ERP, ASP, SCM, CRM,...). Dans ce cas, il est nécessaire de prendre en compte de l'intégration des flux de données ou des processus entre outils.

Parmi les autres difficultés rencontrées en phase de conception du système PLM, il faut noter la complexité et le manque de souplesse des modèles *produit*. Les principales motivations des démarches de standardisation [ISO 94] et de normalisation se font en fonction du métier de base de l'entreprise. Malheureusement, le niveau de détail est trop précis ce qui nuit à la « fluidité » des flux de données.

Enfin, de nombreux problèmes sont inhérents aux modes industriels actuels dans lesquels le processus de développement du produit peut être partagé par différentes entreprises engagées dans une collaboration [BAU 03]. L'intérêt des grands groupes pour ses structures industrielles distribuées ou en réseaux est leur potentiel de réactivité et de maîtrise des coûts [GAU 03]. Ce qui a été initié pour la production de bien se généralise finalement à l'ensemble du processus de développement des produits. La mise en oeuvre d'un système d'information intégré (ou simplement partagé) ne doit en aucun cas être un frein à ce potentiel d'adaptabilité

Ces différents points montrent la nécessité d'agir sur la conception du PLM et sur son évolution au cours du temps. Une des réponses possibles est de définir des modèles intégrés facilitant la conception du système. Ici l'intégration est à prendre dans le sens large au sens générale des Produit / Processus / Organisation.

Ce besoin d'intégration pour la conception peu se compléter d'un besoin de connaissances qui favorise une exploitation optimale du patrimoine technique de l'entreprise. N'oublions pas que le PLM gère l'ensemble du cycle de vie avec pour objectif initial la maîtrise des coûts et des délais par la capitalisation du savoir-faire. Force est de constater que la connaissance n'est que peu expliciter dans les modèles. Le besoin d'intégration de connaissance répond donc à une exigence de réutilisation du savoir-faire d'une entreprise. Il se situe clairement dans une démarche d'innovation ou l'on considère cette dernière comme un processus incrémental s'appuyant sur les expériences passées de l'entreprise. Dans ce cadre, le système PLM qui sont par nature des outils de capitalisation, ont vocation à intégrer (ou s'intégrer) la connaissance.

### *Savoir-faire et connaissances métier*

Il serait abusif de considérer un PLM comme un système de gestion de connaissances. S'il semble logique que les conditions nécessaires à une réduction des coûts et des délais passe une par une capitalisation des informations, on peut

aisément affirmer que ces conditions ne sont pas suffisantes : une partie de la difficulté réside dans l'exploitation et la réutilisation de ce capital. Or, dans un environnement où l'innovation s'appuie sur le savoir-faire, la capitalisation des connaissances au travers le savoir-faire métier constitue un moyen plus efficace pour permettre une bonne réutilisation du patrimoine technique de l'entreprise dans l'hypothèse où celles-ci peuvent être en partie formalisées.

Les connaissances manipulées au sein de l'entreprise sont variées et complexes. Leur classification dépend du point de vue des acteurs. Nous nous baserons sur une distinction couramment admise qui sépare les connaissances explicites et les connaissances tacites. Les connaissances explicites sont dédiées à un problème particulier et concernent des heuristiques qui reflètent l'expérience des experts. Elles sont exprimées sur un support quelconque (plans, mode d'emploi, notes techniques...). Elles sont parfois difficilement accessibles et leur mode de formalisation ne répond pas toujours au besoin exprimé. Les connaissances tacites correspondent au savoir-faire métier, formalisables plus ou moins facilement ou pas. Elles sont dites cruciales [GRU 00] et sont difficilement identifiables. Des études sont actuellement en cours de réalisation pour l'identification des connaissances cruciales au sein de l'entreprise.

Ce chapitre propose un cadre de modélisation intégré pour un système PLM ayant pour objectif de permettre :

- une conception plus rapide du SI avec un niveau de précision des modèles adaptés aux besoins
- une connectivité plus grande par rapport aux plates-formes existantes
- une capacité d'adaptabilité en fonction du contexte organisationnel et compatible des structures légères
- une capitalisation explicites des savoir-faire et des connaissances techniques

Dans la première partie de ce chapitre, nous proposons un axe de modélisation intégré autour de des principales composant d'un PLM. Puis, dans une seconde étape, nous présentons une implantation de notre approche de modélisation au travers un langage de représentation basé sur XML. Enfin, nous montrerons comment les connaissances techniques peuvent s'intégrer dans ce langage de représentation

## **2. Axe de modélisation autour du PLM**

Le contexte lié à la production de biens manufacturé permet de restreindre implicitement les concepts de Système d'Information Produit (ou PLM) comme un sous domaine des SI, orienté vers la gestion des cycles de vie des produits. A partir de différents travaux sur l'ingénierie des PLM [GZA 00] [Per 02] [Eyn 04] [Mer 04], nous pouvons synthétiser trois axes de modélisation qui déterminent les composantes primaires de SIP : Le Produit, le Processus, l'Organisation (PPO). La

#### 4 Ingénierie de la conception et cycle de vie du produit

démarche d'intégration proposée est articlé autour des différents éléments caractérisant ces trois composantes primaires d'un PLM.

##### *Structure des informations autour des produits*

Les modèles Produit constituent sont structurés autour d'approches génériques comme STEP, EDI... Notre démarche de structuration se veut avant tout souple et dynamique et s'articule autour d'entités techniques de lien et d'état

Ces entités doivent être perçues comme une généralisation de la notion d'article, capable de représenter un article physique, un composant, une fonction, une caractéristique géométrique, une annotation, etc. Ils permettent de stocker les données Produit sur la base d'éléments constitutifs qui font finalement apparaître une notion de méta-article. Nous les définissons donc comme un regroupement de propriétés autour d'un concept identifiable par un utilisateur et ceci indépendamment de la représentation dans laquelle se place l'acteur du PLM (conception, industrialisation, fabrication,...).

Des relations permettent de lier les entités techniques entre elles. Chaque lien porte un sens spécifique et possède des propriétés qui lui sont propres (exemple: lien de nomenclature organique, lien de nomenclature fonctionnelle, lien documentaire, lien d'annotation, etc.). Les liens doivent permettre une gestion dynamique des relations entre les objets techniques. Ainsi, la notion de nomenclature qui structure généralement un système PLM, n'apparaît pas explicitement par une relation (de type agrégation) entre les classes. L'idée principale sous-jacente est de favoriser une exploitation générique, notamment pour rechercher des éléments constitutifs. Un lien est constitué d'entrées qui référencent les différentes catégories d'objets à mettre en relation. A chaque entrée, on associe des références sur des objets ou sur des groupes d'objets. On distingue quatre catégories afin de spécifier les règles et les contraintes en fonction de l'utilisation du lien.

- les contraintes donnent la possibilité de définir des conditions soit sur les cardinalités des entités liées, soit sur des références d'entités, soit sur les deux;
- les compatibilités s'utilisent lorsqu'un objet peut se substituer à un autre, en l'occurrence dans une nomenclature. Elles permettent aussi d'identifier les cas inverses, c'est à dire lorsque qu'il y incompatibilité entre deux objets (incompatibilité fonctionnelle, incompatibilité de matière, etc.);
- les catalogues sont utilisés pour regrouper des objets techniques sans contrainte particulière. Ils permettent, par exemple, la définition de catalogues fournisseurs;
- les variantes constituent un type d'entrées plus complexes que les précédentes. Elles permettent de définir les liens de substitution sous des conditions explicitées

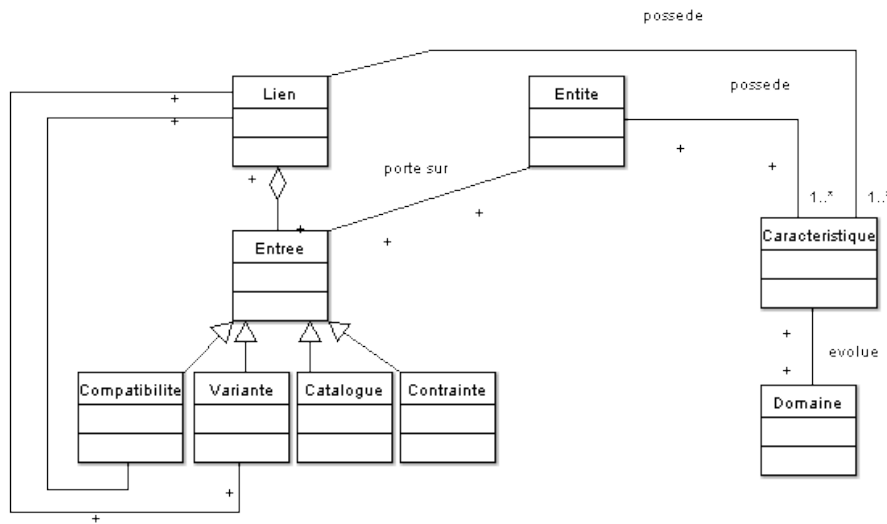


Figure 1 Diagramme des Entités

Une réelle gestion du cycle de vie du produit implique la prise en compte de tous les états intermédiaires des cycles de vie. Un *état* est un objet qui caractérise le statut d'une entité technique. Dans un système PLM, il existe des différents niveaux de statut. Par exemple, un document va pouvoir posséder plusieurs révisions valides, avec pour chaque révision, un degré de maturité différent. Cette diversité, nous amène à définir un état comme un ensemble de vecteurs d'état constitué de conditions d'entrée et de transitions vers d'autres états. Un espace d'état définit un ensemble d'états tel que pour tout état de l'espace il existe une séquence de transition entrante ou sortante pour passer vers un autre état de l'espace. Ainsi, la possibilité de définir ces espaces d'état permet à chaque entreprise de déterminer les statuts nécessaires à ces processus (versionnement, maturité, etc.).

### ***Processus et cycle de vie***

La question des processus au sein du système PLM est une question complexe. Indépendamment de tout formalisme, un système PLM a vocation à gérer les processus qui contribue directement au développement des produits tout au long de son cycle de vie. Nous considérons qu'un processus entre dans le cadre des processus gérés par le PLM, lorsque son objectif et/ou ses résultats permettent de

## 6 Ingénierie de la conception et cycle de vie du produit

générer des éléments du produit ,des processus manipulant les éléments du produit, des structures organisationnelles pour manipuler des éléments. Cela inclut à la fois les processus métier, les processus de gestion des flux, etc.

Dans une approche du cycle de vie centré processus, il y a une forte imbrication entre les processus de contrôle des états (de produit) et les processus métier. Plusieurs raisons à cela : D'une part, l'activité métier génère de l'information qui est par définition contrôlé par les processus de gestion. D'autre part, le formalisme pour représenter les processus est dépend du moteur d'exécution, et donc généralement unique par système. Cette imbrication des processus de gestion et métiers au sein d'un PLM reste problématique du fait de la disparité de leur niveau de structuration et de détails. Si, prendre un formalisme unique, est un garanti d'intégration et de continuité des flux, il est nécessaire d'introduire des éléments permettant de tenir compte de ces différents niveaux de structuration

### *Gestion des processus*

Nous proposons ici une représentation des processus basée sur un système de workflow. Comme le propose la WfMC [WFM 05], le méta-modèle du workflow est centré sur l'activité, en vue d'une automatisation. Le workflow est donc utilisé pour définir des processus et les exécuter. Ce dernier point n'est pas sans poser des problèmes car il engendre des blocages [SAI 01] (erreur, état instable) sur les exécutions et un manque d'évolutivité sur la définition

Notre approche de la modélisation des processus vise justement à introduire une certaine souplesse par l'introduction de caractéristiques et d'opérateurs spécifiques pour un système PLM . Au final, les caractéristiques propres à un système PLM permettent de tenir compte du faible niveau de détail de certains processus, l'ensemble des opérateurs proposés permet si besoin de construire des processus légers ou relativement peu structurés.

### *Caractéristiques des processus au sein d'un PLM*

Les caractéristiques propres au PLM sont définies au travers la notion . Les objectifs sont pertinents pour caractériser les activités dans un processus dans la mesure ils représentent les résultats attendus par l'exécution d'une l'activité. L'objectif ne peut se résumer aux seuls éléments de flux en sortie de l'activité et d'ailleurs dans bien des cas, ils seront difficilement évaluable. Ainsi, La caractérisation des objectifs se fait par la définition de différents types dont le rôle est de distinguer le degré de mesurabilité et de continuité. En effet, s'il est relativement aisé d'évaluer un objectif basé sur une valeur de seuil ou sur la production d'un document technique, il n'en est pas de même lorsque la finalité de l'activité participe au pilotage ou à la gestion de PLM. Par exemple, une activité de régulation d'un groupe de travail est difficilement mesurable en terme quantitatif.

Nous distinguons divers types d'objectifs afin de tenir compte des difficultés de mesure pour certains cas. Les objectifs *identifiables* sont des objectifs pouvant être

décrits et interprétés par un acteur du SIP, les objectifs *mesurables* sont des objectifs identifiables sur lesquels l'acteur peut disposer de moyen de mesure afin d'en évaluer la réalisation, les objectifs *évaluables* sont des objectifs mesurables dont l'évaluation peut être faite par un automate ou par un programme. Pour chacune de ces catégories, nous définissons des diagrammes d'état qui caractérise leur dynamique en donnant la possibilité de ne pas avoir un état final stable. D'une, nous avons des objectifs qui peuvent être atteints ou non atteints mais qui, une fois dans un de ces deux états, ne peuvent plus en changer. D'autre part, nous avons des objectifs dans lesquels il existe une transition permettant de passer d'un état atteint à non atteint.

Les objectifs ainsi identifiés, il est possible de définir des modes d'exécution de processus suivant quatre critères: le temps, la robustesse, la généricité, la mesurabilité. L'intérêt est d'avoir des modes d'exécution qui ne nécessitent pas un haut niveau de structuration ou de détail. Le Tableau 1 présente les modes d'exécution des processus en fonction des critères

<i>Critère</i>	<i>Mode</i>
Temps	<ul style="list-style-type: none"> <li>•Borné : processus dont l'état initial (respectivement l'état final) marque le début (respectivement la fin) de son exécution</li> <li>•Non borné : processus dont l'état initial ou l'état final ne peut être clairement identifié. C'est le cas de certains processus qui restent dans un état non final ou dans une alternance d'états non finaux</li> </ul>
Robustesse	<ul style="list-style-type: none"> <li>• Stable: processus identifié et dont la modélisation des constituants n'évolue pas ou peu</li> <li>•Évolutif : processus identifié dont certains élément évoluent périodiquement</li> <li>•Instable : processus identifié mais dont les constituants évoluent sans périodicité et relativement souvent</li> </ul>
Généricité	<ul style="list-style-type: none"> <li>•Instanciable : processus pouvant être instancié et exécuté dans un contexte (au niveau de données) particulier. Plusieurs instances d'un même processus peuvent s'exécuter simultanément</li> <li>•Unique : processus non instanciable en de multiples instances. Une seule instance peut s'exécuter à la fois</li> </ul>



Mesurabilité	<ul style="list-style-type: none"> <li>• Mesurable : processus dont certains objectifs ou résultats peuvent être mesurés au moyen d'indicateurs</li> <li>• Non Mesurable : Processus dont les objectifs ou les résultats sont trop subjectifs pour être mesurés. Ils nécessitent un savoir-faire particulier. Les changements d'états de ces processus se font généralement sur l'initiative d'un des acteurs du processus</li> </ul>
--------------	---

Tableau 1: Critères / Caractéristiques des processus

*Les opérateurs d'activités*

Les opérateurs d'activités permettent de relier entre elles les différentes activités d'un processus. Ils sont utilisés dans une approche de décomposition et d'ordonnancement. Classiquement nous retrouvons, les opérateurs de composition qui permettent décomposer une activité en sous-activités, les opérateurs d'enchaînement (Séquence, Parallèle, Branchement, Synchro, Boucle) qui permettent d'ordonnancer les activités. Ces opérateurs sont présents dans tous les modèles: IDEF3, CIMOSA, UEML, etc. Les systèmes de workflow propose par ailleurs des opérateurs événementiels. Ils sont essentiellement utilisés pour réagir à des évènements extérieurs, non liés aux états ou aux sorties des activités concernées.

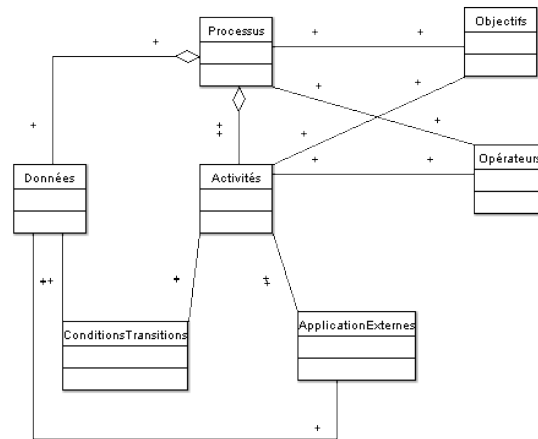


Figure 2 Diagramme des Processus

Nous proposons d'enrichir les opérateurs événementiels des workflow par trois opérateurs spécifiques :

- L'opérateur *Exception* permet notamment de gérer les exceptions et les erreurs sur une activité.

- L'opérateur *Délai* déclenche une action (une activité) au bout d'un certain délai.
- L'opérateur *Arbitrage* intervient dans le cadre d'une régulation. Cet opérateur peut être utilisé avec une activité dont les acteurs appartiennent à des organisations industrielles distinctes. Il s'agit donc d'une activité coopérative.

### **Organisation des structures industrielles**

La composante organisationnelle d'un système PLM a vocation à organiser tous les acteurs qui contribuent de près ou de loin aux processus de développement du produit. Dans le cas des systèmes PLM, elle doit répondre à trois contraintes spécifiques majeures : diversité, sécurité, adaptabilité.

En effet, une entreprise doit pouvoir organiser son personnel et ses modes de fonctionnement dans la plupart des schémas organisationnelles en vigueur. D'ailleurs, il n'est pas rare de trouver plusieurs modes de fonctionnement en même temps au sein d'une même entreprise. La sécurité des données et des processus devient vitale dès que l'entreprise a des partenaires. La structure organisationnelle doit faciliter les collaborations entre partenaires en contrôlant la mise à disposition de l'ensemble ou d'une partie du patrimoine technique de chacun des partenaires [Noe ]. Enfin, dans le cas de structures industrielles de type PME/PMI, elle doit pouvoir s'adapter aux réorganisations et aux missions sur des projet en cours

Afin de répondre aux contraintes précédentes nous définissons une organisation des structures industrielles à partir autour de composants collaboratif dénommés *Espace Collaboratif Produit* (ECP) dont l'objectif est de permettre la mise en oeuvre, le maintien et l'évaluation d'un travail coopératif en interne ou avec des partenaires extérieurs à l'entreprise. L'ECP est donc le lieu virtuel où se déroule des activités coopératives orientées vers le développement des produits. Il regroupe des acteurs, des données techniques, des processus, des règles d'usage et divers autres éléments qui contribuent tous à la mise en oeuvre et à la bonne marche d'une coopération au sein d'un PLM. Ce composant est constitué en trois unités distinctes dont chacune assure certaines fonctionnalités : les unités d'organisation, les unités de connexion, les unités de régulation

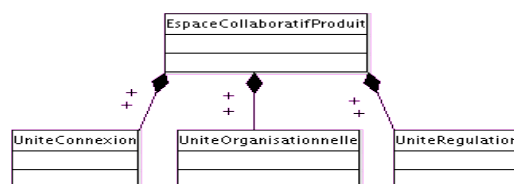


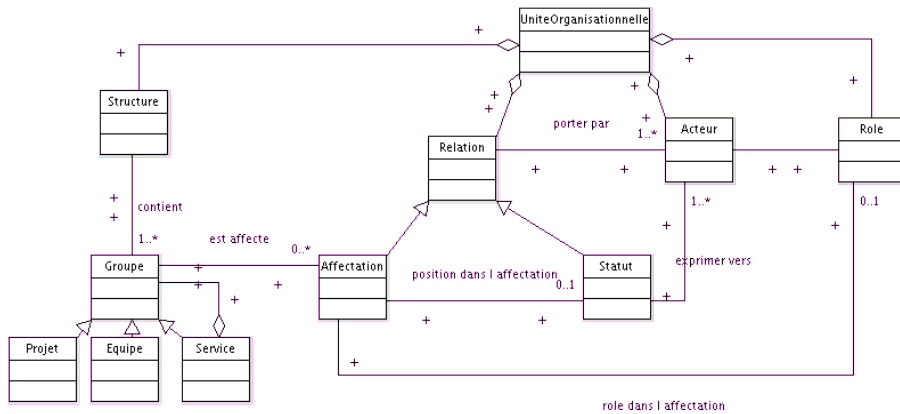
Figure 3 Diagramme des ECP

*Les unités organisationnelles*

Les *Unités Organisationnelles* (UO) sont des éléments permettant de regrouper les acteurs ainsi que leurs capacités d'action sur le SIP. Elles définissent la structure de l'organisation, soit à partir d'un découpage traditionnel (structure matricielle, divisionnelle, projet, etc. ), soit à partir d'une définition spécifique des relations entre acteurs. De plus, elles gèrent la politique d'accès aux données techniques par l'intermédiaire des vues et des rôles associés aux acteurs ou aux éléments de la structure.

Les *relations* sont les éléments qui permettent de positionner de façon effective les acteurs dans les structures. Les relations d'affectation sont utilisées pour établir des regroupements par rapport à une configuration de travail. Il faut noter que l'appartenance à l'un des composants de la structure (service, équipe, projet) est toujours une relation dynamique et paramétrable. Les relations statutaires regroupent l'ensemble des positions d'un acteur par rapport aux autres acteurs. Elle caractérise généralement les relations hiérarchiques et les liens d'autorités entre les individus.

Figure 4 Diagramme des UO



*Les unités de connexion*

Les *Unités de Connexion* (UC) sont composées de connecteurs dont la fonction est d'établir un canal de communication entre les différents ECP/UO. Ils doivent garantir la continuité des flux (données ou contrôles) de manière transparente entre plusieurs ECP ou UO, tout en veillant aux règles de restriction imposées par leur configuration. Cette continuité du flux se complète par une continuité sur les processus. Par exemple, l'état d'une activité s'exécutant dans un ECP doit pouvoir être modifiée par une action réalisé dans un ECP lié.

Les connecteurs peuvent être considérés comme des filtres sur les flux d'informations. Chaque gestionnaire d'ECP a la possibilité de définir ses connecteurs. Comme l'indique la figure à chaque filtre, on associe des Ayants droit qui représentent les groupes, les rôles ou les acteurs concernés par le filtre. Un filtre permet donc d'associer un ensemble d'ayant droits avec des éléments de flux entrant ou sortant.

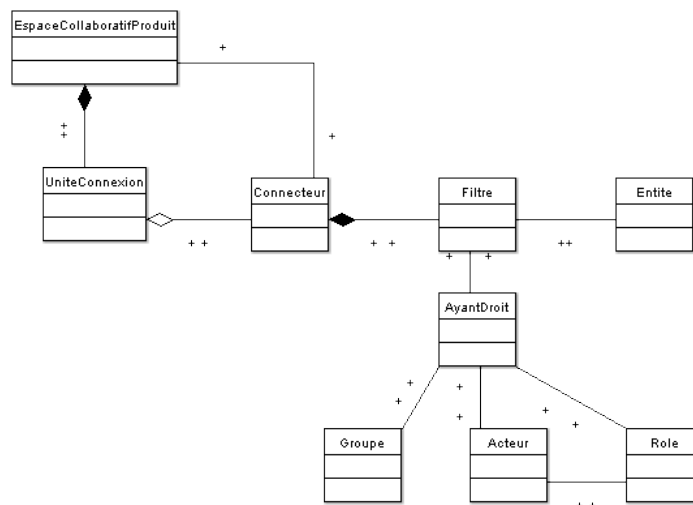


Figure 5 Diagramme des UC

### Les unités de régulation

Les *Unités de Régulation* sont introduites afin de garantir, la réelle mise en oeuvre d'une activité coopérante, ainsi que sa pérennité dans le temps. La régulation des interactions entre les différents acteurs contribue à maintenir l'équilibre de la coopération par l'acceptation et la mise en oeuvre de règles et de protocoles au niveau du fonctionnement du groupe.

Avant même de définir les protocoles qui vont gérer les interactions, les *positions* permettent de formaliser la perception par les acteurs de leurs activités au sein du groupe. Les positions dans l'unité de régulation sont des positions subjectives. Un acteur exprime un ensemble de positions qui concernent, soit son environnement de travail, soit la façon dont il se perçoit dans cet environnement. Ainsi, les positions d'utilisation (des ressources) sont utilisées pour exprimer les préférences d'un acteur sur un des éléments du SIP. Les positions de compétence représentent le savoir-faire d'un acteur dans le cadre du SIP. La compétence est généralement exprimée sous trois aspects (le savoir, le savoir-faire, le savoir être) [Har 00]. Pour notre part, nous ne nous intéresserons qu'au deux premiers aspects.

Les positions de compétence sont utilisées pour créer des regroupements cohérents ou complémentaires. Elles peuvent être, par exemple, utilisés pour définir des rôles similaires permettent de pallier une absence effective d'acteurs sur un rôle déterminé.

La formalisation des protocoles qui vont régir les interactions est établie dans notre modèle à partir des règles d'usage qui correspondent à une réponse en rapport avec les interactions des acteurs et/ou du système. On distingue alors les règles causales (exemple: le changement d'affectation d'un responsable projet (la cause) implique l'envoi d'un message à tous les membres (l'effet), et les règles d'arbitrage qui décrivent les protocoles à mettre en oeuvre en cas de conflit ou de problèmes.

### **3.Implémentation d'une modélisation intégrée Produit/ Processus/ Organisation**

Dans ce paragraphe, nous présentons un langage de modélisation intégré construit sur XML (eXtensible Markup Language). Cette démarche présente deux intérêts majeurs. Tout d'abord, elle permet de représenter et de structurer tous les éléments (article, nomenclature, activité, organisation, etc.) de façon globale avec un formalisme unique. Le second intérêt est lié à l'utilisation de XML qui est de plus en plus utilisé comme un standard d'échange [BUR 01] [EBX 01], confortant ainsi une démarche de compatibilité avec les standards actuels. Le langage proposé (Product Information System Markup Language) [PER 02] [PER 03] permet de formaliser les différents éléments des trois axes (PPO) présentés dans le paragraphe précédent. Il permet de décrire d'un part les objets génériques permettant de concevoir un système d'information PLM, d'autres part les instances d'objets circulant dans le PLM

#### ***Référencement des éléments***

PISML manipule des objets dont certains sont en fait, soit des classes, soit des instances de ces classes. Dans tous les cas, il est indispensable de pouvoir les référencer explicitement. Les références en PISML utilisent une notation proche des URL (Unified RessourcesLocation). La structure d'une référence contient divers éléments :

- un type qui caractérise la classe d'objets désignée par la référence (une entité technique, un objet technique, un acteur, un domaine, ...),
- une localisation qui permet de situer une entité dans l'espace organisationnel de l'entreprise ou d'un partenaire,
- des entités désignant l'identifiant de l'élément à caractériser,

- des expressions (facultatives) qui permettent de formuler des conditions éventuelles de restrictions sur l'entité concernée.

PISREF :<type>//<localisation> <entite>(<expression>) (1)

### *Structure grammaticale*

Tous les éléments du modèle PPO sont définis dans le langage PISML sous forme de balises (<ClassItem> </ClassItem>, <ClassLink> </ClassLink>, ...). La grammaire du langage est définie sous la forme de DTD (Document Type Definition). Ces DTD (voir Tableau 2) permettent de définir les règles syntaxiques du langage et donc de construire des documents XML valides.

```

<!ELEMENT ClassItem (ClassObject | ClassLink | ClassState)>
<!ELEMENT ClassLink (Inputs, Items?)>
<!ATTLClassState ClassLink
    idname ID #REQUIRED
    lib CDATA #IMPLIED
>
<!ELEMENT Inputs (Library+ | Variant+ | ConstraintLink+)>
<!ELEMENT Input (In, Out,Items? )>
<!ATTLClassState ConstraintLink
    card CDATA #REQUIRED
    required (YES | NO) #IMPLIED
>
<!ELEMENT In (PismlRef+)>
<!ELEMENT Out (PismlRef+)>
<!ELEMENT ClassObject (Items)>
<!ATTLClassState ClassObject
    idname ID #REQUIRED
>
<!ELEMENT Items (Item+)>
<!ELEMENT Item (Lib?,(Domain|DomainRef))>
<!ATTLClassState Item
    idname ID #REQUIRED
    required (YES|NO) #IMPLIED
    formatted CDATA #IMPLIED

```

Tableau 2: Extrait de la DTD des Entités

Nous définissons trois DTD pour décrire les éléments génériques (classe) de notre modèle PPO ainsi que trois autres pour les flux d'informations (instances de classe) [PER 02]. C'est à partir de ce langage que nous définissons dans le paragraphe suivant, certains éléments des connaissances métier concernant un procédé de type usage.

#### 4. Vers l'intégration des connaissances métier

Nous avons rappelé en introduction les différentes hypothèses concernant les connaissances métiers traitées. L'objectif de ce paragraphe est de montrer, au travers d'un exemple, comment définir les caractéristiques nécessaires à la mise en oeuvre de connaissances métiers au sein d'un système PLM.

L'exemple traité concerne une méthode développée pour évaluer les dispersions d'usinage dans une démarche de conception et de fabrication de pièces mécaniques. Nous appelons dispersions d'usinage  $\Delta_i$  (en mm) les variations géométriques et dimensionnelles obtenues sur une série de pièces réelles pour une gamme de fabrication et un usinage donné. Les origines et les causes même des dispersions d'usinage sont multiples : flexion des outils, asservissement, géométrie de la machine-outil [ANS 03]. La résultante de ces origines possibles et les connaissances relatives aux lois qui les régissent apparaissent donc difficilement formalisables sans l'appui des experts métier. Il s'agit donc de quantifier ces dispersions d'usinage en fonction de paramètres produit/process définis par l'expert et d'intégrer ces connaissances à un système PLM en vue de son utilisation dans un processus global de développement d'un nouveau produit.

##### *Description des connaissances à intégrer*

La modélisation des connaissances métiers est basée sur l'usinage d'une pièce type de tournage comme le montre la Figure 6. Nous avons défini cinq types de dispersions qui prennent en compte les spécifications géométriques suivant les axes X et Z de la machine.

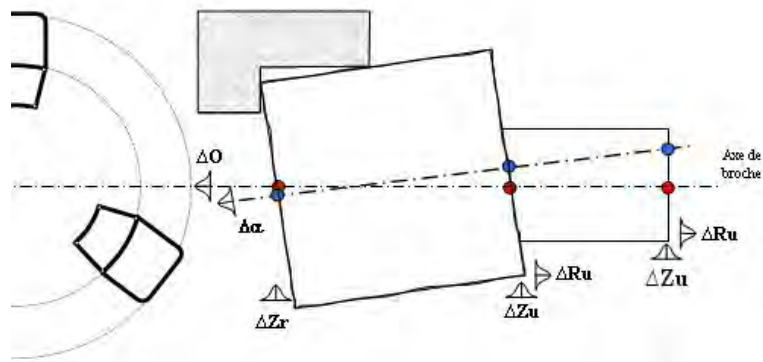


Figure 6 Dispersion d'usinage  $\Delta_i$  suivant les axes X et Z

*Définition du modèle*

Les modèles obtenus sont du type :

$\Delta_{machine} = f_i(p_i)$  où  $\Delta_{machine}$  représente l'une des cinq dispersions  $\Delta_i$  et  $p_i$  un ensemble de paramètres produit/process discret ou continu. Les fonctions de régression linéaire  $f_i$  sont issues d'une modélisation de type plans d'expériences.

Les paramètres d'entrées  $p_i$ , proposés par les experts et permettant l'évaluation des dispersions, sont donnés dans le tableau suivant :

Paramètres $p_i$		type	modalités	
nuance plaquette	N	discret	P15	P35
rayon (mm) de bec	$R_\epsilon$	discret	0.40	0.80
vitesse (m/min) de coupe	Vc	continu	150	280
matériau de la pièce usinée	M	discret	A60	XC38
Avance (mm/tour) f	f	continu	0.10	0.30

Compte tenu des critères de choix des plans (orthogonalité et nombre de degrés de liberté), la table  $L_{16}(2^{15})$  de Taguchi a été choisie pour mener les expérimentations.

A titre d'illustration, suite à une campagne d'essais pour évaluer la dispersion  $\Delta Ru$  sur le petit diamètre de la pièce tournée notée  $\Delta Ru25$ , nous obtenons le modèle suivant :

$$\Delta Ru25 = \alpha + \beta N + \chi R_\epsilon + \delta Vc + \epsilon M + \phi f + \gamma NV - \eta Nf - \varphi VcM + \kappa Mf \quad (1)$$

avec pour coefficients:

$\alpha$	0.02789	$\beta$	0.01822	$\chi$	0.00637	$\delta$	0.00008	$\epsilon$	0.01187
$\phi$	0.01941	$\gamma$	0.00004	$\eta$	0.02873	$\varphi$	0.00010	$\kappa$	0.06160

Tableau 3: Coefficients de l'équation de régression

*représentation sous forme de cartes de savoir-faire*

Pour des raisons d'utilisation et d'accessibilité des connaissances modélisées et afin de proposer des zones de faisabilité technique, les fonctions  $f_i$  sont représentées sous la forme de carte de savoir-faire [Lef01].



Dans le modèle (1), seuls deux paramètres sont continus : la vitesse de coupe  $V_c$  et l'avance  $f$ . La représentation en deux et trois dimensions est donc possible.

L'équation (1), de la forme  $\Delta Ru_{25} = C_1 + \alpha_1 x_1 + \beta_1 x_2$  où  $C_1$ ,  $\alpha_1$  et  $\beta_1$  sont des constantes et  $x_1$  et  $x_2$  les paramètres continus  $V_c$  et  $f$ , est celle d'un plan représenté sur la figure

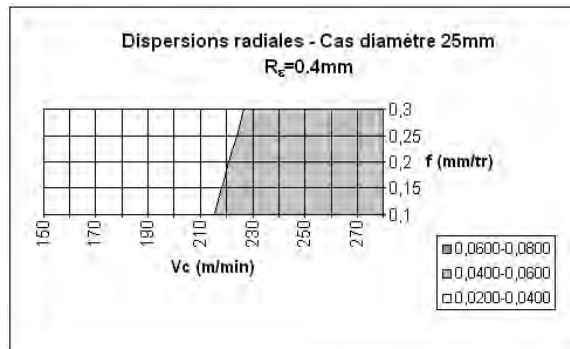


Figure 7 : Carte de dispersions  $\Delta Ru_{25}$  en fonction de  $f$  et de  $V_c$

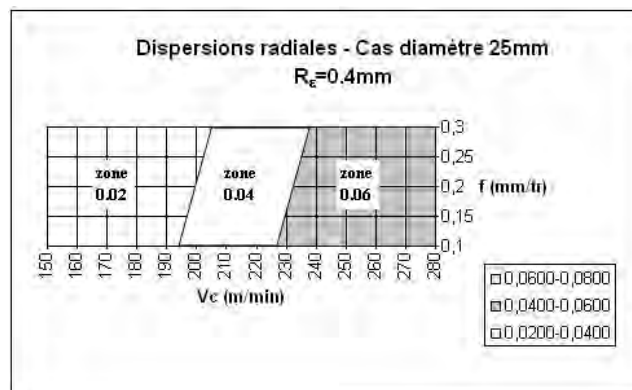


Figure 8 : Carte de dispersions expertisée  $\Delta Ru_{25}$

La frontière entre les deux zones est donc le lieu où  $\Delta Ru_{25}$  est égale à 0.04. L'équation (1) devient alors :  $0.04 = C_1 + \alpha_1 x_1 + \beta_1 x_2$ . Cette équation est celle d'une droite. Nous considérons donc que les points situés proches et sur la droite commune sont inclus dans une troisième zone de dispersions « 0.04 », dont l'amplitude est définie par l'expert. La Figure 7 présente la carte validée par l'expert qui permet de définir des zones de faisabilité technique.

***Elément de modélisation du PLM***

La méthode de formalisation des dispersions d'usinage présentée dans le paragraphe précédent caractérise des connaissances métier utilisables au sein d'un processus d'ingénierie simultanée de conception-fabrication. L'objectif d'intégration de ces connaissances au sein d'un système PLM implique la définition des principaux concepts suivant éléments de modélisation PPO. Le tableau suivant présente les différents éléments PPO, leur description et leur type.

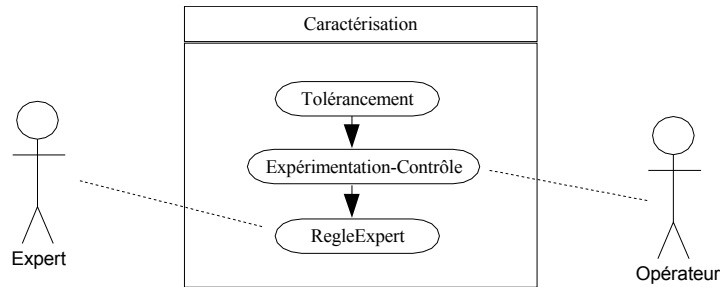
<b><i>Elément PPO</i></b>	<b><i>Description</i></b>	<b><i>Type</i></b>
RessourceMachine	Caractéristiques : nombre d'axes, vitesse de broche, puissance, avance	Entité
Pièce	Caractéristiques : modèle géométrique, spécifications dimensionnelles, spécifications géométriques, matériau	Entité
Outil	Caractéristiques : rayon de bec, vitesse de coupe	Entité
CarteSavoirFaire SF	$\Delta_{machine}$	Entité
Expérimentation-contrôle	Réaliser la pièce et calculer le lien dispersion : usinage et détermination des fonctions fi	Activité
Tolérancement	Caractériser les fonctions de la pièce : Spécification dimensionnel, et spécification géométrique	Activité
RegleExpert	Valide les carte de savoir-faire : Prend en entrée une carteSF et retourne une carteSF exploitable	Activité
Expert	Acteur possédant l'expertise pour valider les cartes de savoir faire	Organisation
Opérateur	Acteur qui va réaliser les cartesSF	Organisation
Dispersion	RessourceMachince, Pièces, CarteSF	Lien

Caractérisation d'une machine	Processus de caractérisation d'une machine composé des trois activités précédentes : voir Figure 9	Processus
-------------------------------	--	-----------

Tableau 4: Eléments de connaissance du modèle PPO

Figure 9 : Description du processus de caractérisation

Description PISML



la description des éléments de connaissance dans le langage PISML permet de décrire les concepts manipulés au sein du PLM. Les figures suivantes présente quelques-uns des éléments du tableau précédent

```

<!DOCTYPE Entite SYSTEM "Entite.dtd">
<ClassItem>
<ClassObject idname="RessourceMachine" level="">
  <Items>
    <Item idname="N" required="yes" formatted="NONE">
      <lib>vitesse de broche</lib>
    </Item>
    <Item idname="NB_axe" required="yes" formatted="NONE">
    </Item>
    <Item idname="Puissance" required="yes" formatted="NONE">
    </Item>
    <Item idname="Avance" required="yes" formatted="NONE">
    </Item>
  </Items>
</ClassObject>
</ClassItem>

```

Tableau 5: l'entité RessourceMachine

```

<!DOCTYPE Entite SYSTEM "Entite.dtd">
<ClassItem>

```

```

<ClassObject idname="CarteSavoirFaire" level="">
  <Items>
    <Item idname="D_Alpha" required="yes" formatted="NONE"></Item>
    <Item idname="D_O" required="yes" formatted="NONE"></Item>
    <Item idname="D_Zu" required="yes" formatted="NONE"></Item>
    <Item idname="D_Zr" required="yes" formatted="NONE"></Item>
    <Item idname="D_Ru" required="yes" formatted="NONE"></Item>
  </Items>
</ClassObject>
</ClassItem>

```

Tableau 6: l'entité CarteSF

```

<!DOCTYPE Entite SYSTEM "Entite.dtd">
<ClassItem>
  <ClassLink idname="Dispersion" lib="Dispersion">
    <Inputs>
      <Input card="1" required="YES">
        <In>
          <PismlRef pref="PISREF:ClassItem//.@./ResourceMachine()"/>
        </In>
      </Input>
      <Library>
        <In>
          <PismlRef pref="PISREF:ClassItem//.@./Piece()"/>
          <PismlRef pref="PISREF:ClassItem//.@./Outils()"/>
        </In>
        <Out><PismlRef pref="PISREF:ClassItem//.@./CarteSF()"/> </Out>
      <Items>
        <Item idname="FonctionRégrétion">
          <Domain idname="EquationLineaire" >
            <Equation >
              <Constant>C1</Constant>
              <Constant>Alpha 1</Constant>
              <Constant>Beta 1</Constant>
              <Var>D_Ru25</Var>
              <Var>X1</Var>
              <Var>X2</Var>
              <Expression>D_Ru25 = C1 + Alpha 1 *X1+Beta 1 *X2</Expression>
            </Equation>
          </Domain>
        </Item>
      </Items>
    </Library>
  </Inputs>
</ClassLink>
</ClassItem>

```

Tableau 7: le lien Dispersion

```

<Process idname="caracterisation">

```

```

<!--
=====
Caracteristique descriptive du processus
=====
-->
<Description>
  <Name>Ce processus évalue les caractéristique d'une machine<Name>
</Description>
<!--
=====
Caractéristiques typologiques du processus
=====
-->
<Typology>
  <Time type="borne">
    <InitState />
    <FinalState />
  </Time>
  <Sturdyness type="stable" />
  <Measurable type="mesurable"/>
  <Generic type="instanciable" />
</Typology>
<!--
=====
Description des activités du processus
=====
-->
<Activities>

  <!-- ++++ OPERATEUR ENCHAINEMENT +++++ -->
  <LinksOperator>

    <Sequence>
      <From>
        <RefActivity idref="Tolerancement" />
      </From>
      <To>
        <RefActivity idref="ExperimentationControle" />
      </To>
    </Sequence>
    ...

  </LinksOperateur>

  <!-- ++++++ ++++++ ++++++ ++++++ ++++++ -->
  <Activity id="Tolerancement">
  <!-- ++++++ ++++++ ++++++ ++++++ ++++++ -->

  <lib>Définir les spécifications dimensionnelles et géométrique</lib>
  <Objectif type="DISCRET" is="IDENTIFIABLE">
  <Lib>l'objectif de cette activité est de concevoir

```

```

    une maquette correspondant à une demande
  </lib>
  <ReachState type="MANUAL"></ReachState>
  <UnreachState type="MANUAL"></UnreachState>
</Objectif>

<Actors>
  <Pismref pref="PISREF:Actor/.@*/equipeBE()">
</Actors>

....
</Activities>ion
</Process>

```

Tableau 8: Extrait du processus « *Caractérisation d'une machine* »

Les éléments PISML présentés dans les tableaux précédents vont donc permettre de concevoir un système PLM intégrant des connaissances métier relatives aux dispersions d'usinage. Ainsi l'entreprise pourra bénéficier des outils de capitalisation inhérents aux PDM (stockage des données, traçabilité, accès sécurisé, requête sur des connaissances techniques,...)

## 5. Conclusion

Nous avons présenté au travers de ce chapitre une démarche pour intégrer des connaissances métiers au sein d'un système PLM. Pour ce faire, nous avons utilisé un modèle Produit / Processus / Organisation adapté au système PLM et basé sur :

- Une description produit orientée méta-article
- Une description des processus de type workflow complété d'objectif et d'opérateurs spécifiques
- Une description des organisations suivant des unités fonctionnelles

A partir de ces éléments nous avons traité un exemple d'intégration de connaissance métiers dans un système PLM. L'exemple de connaissance permet de caractériser une dispersion d'usinage. Il est le résultat d'un processus d'expertise et d'expérimentation associé à des résultats de calcul. Les éléments de calcul sont donc modélisés sous forme d'entité et le processus d'expertise et de d'acquisition sous forme de Workflow. Une analyse détaillée nécessiterait de caractériser plus en détail au (niveau organisationnel) la compétence d'expertise. L'identification des objets de connaissances nécessite l'expertise au niveau du SI. Il nous semble intéressant de proposer avec le modèle une méthode d'identification de ces objets par des non-spécialistes des SI

## 6. Bibliographie

- [ANS 03] ANSELMETTI B., *Cotation de fabrication et métrologie*, Vol.3, Hermès, Paris, 2003.
- [BAU 03] BAUMBERGER C., PULM U., LINDEMANN U., « Coordination and controlling of distributed product development processes », *ICED 2003*, Stockholm, Suède, 2003.
- [BOC 04] BOCK C., « Systems Engineering in the Product Lifecycle », *International Journal of Product Development*, Special Issue on Product Lifecycle Management, 2004.
- [BOO 94] BOOCH G., *Object-Oriented Analysis and Design with Applications*, Addison-Wesley, 1994.
- [BUR 01] BURKETT W. C., « Product data markup language: a new paradigm for product data exchange and integration », *Computer-Aided Design*, Vol. 33, No. 7, p. 489-500, 2001.
- [CIM 01] CIMdata, « Product Lifecycle Management (PLM) Definition », <http://www.cimdata.com/PLM/plm.html>, 2001.
- [COA 92] COAD P., YOURDON E., *Object-Oriented Analysis*, Prentice Hall, 1992.
- [EBX 01] EbXML, « technical architecture specification », <http://www.ebxml.org>, 2001.
- [EYN 04] EYNARD B., GALLET T., NOWAK P., ROUCOULES L., « UML based specifications of PDM product structure and workflow », *Computers in Industry*, Vol. 55, N°3, p.301-316, 2004.
- [FIL 00] FILSON A., LEWIS A., « Cultural issues in implementing changes to new product development process in a small to medium sized enterprise (SME) », *Journal of Engineering Design*, Vol. 11, N°2, p.149-157, 2000.
- [GAU 03] GARUD R., KUMURASWAMY A., LANGLOIS R. N., *Introduction, In Managing in the modular age: architectures, networks and organizations*, Blackwell Publishers, 2003.
- [GIR 01] GIRAUDIN J.P., CHABRE-PECCOUD M., RIEU D., SAINT-MARCEL C., *Modèles de spécification pour l'ingénierie des systèmes d'information*, *Ingénierie des systèmes d'information*, Hermes Science Publications, 2001.
- [GZA 00] GZARA L., Les patterns pour l'ingénierie des systèmes d'information produit, de doctorat, Institut National Polytechnique de Grenoble, 2000
- [GRU 00] GRUNDSTEIN M., « Repérer et mettre en valeur les connaissances cruciales pour l'entreprise », 10th International AFAV congress, Paris, 7-8 Novembre 2000.
- [IPP 05] IPPPOP, <http://www.opencascade.org/ippop>, 2005.
- [ISO 94] ISO 10303-1: Industrial automation systems and integration -- Product data representation and exchange -- Part 1: Overview and fundamental principles. Genève, 1994.
- [LEF 01] LEFEBVRE A., Méthodologie et outils de capitalisation des connaissances métier pour l'élaboration de gammes d'usinage. Application à l'usinage des culasses automobiles, Thèse de doctorat, Université Blaise Pascal, Clermont II, 2001.
- [MER 04] MERLO C., GIRARD P., « Information System Modelling for Engineering Design Coordination », *Computers in Industry*, Vol. 55, N° 3, p.317-334, 2004

- [NOE 03] NOËL F., BRISSAUD D. « Dynamic data sharing in a collaborative design environment », *International Journal of Integrating Manufacturing*, Vol. 16, p. 546-556, 2003.
- [PER 02] PERNELLE P., Système d'Information Produit pour les PME/PMI : modélisation multi-niveaux d'entreprises engagées dans un travail collaboratif, Thèse de doctorat, Université de Savoie, 2002.
- [RAL 01] RALYTE, Ingénierie des méthodes à base de composants, Thèse de doctorat, Université Paris I, 2001.
- [ROL 01] ROLLAND G., GROSZ, *De la modélisation conceptuelle à l'ingénierie des besoins, Ingénierie des systèmes d'information*, Hermes Science Publications, 2001.
- [SAI 01] SAIKALI K., Flexibilité des workflows par l'approche objet: 2Flow: un framework pour workflows flexibles, Thèse de doctorat, Ecole Centrale de Lyon, 2001.
- [SUD 05] SUDARSAN R., FENVES S., SRIRAM R. D., WANG F., « A Product Information Modeling Framework for Product Lifecycle Management » *Computer-Aided Design*, 2005.
- [WFM 05] WORKFLOW MANAGEMENT COALITION. « Overview , reference, model and glossary » . <http://www.wfmc.org/> , 2005.