UNIVERSITÉ DE LORRAINE

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

Nancy-Université
Université Henri Poincaré

FACULTE DES SCIENCES & TECHNIQUES

U.F.R. Sciences et Techniques Mathématiques, Informatique et Automatique
Ecole Doctorale IAEM Lorraine
Département de Formation Doctorale Automatique

Thèse

présentée pour l'obtention du titre de

Docteur de l'Université Henri Poincaré, Nancy-I

en Automatique, Traitement du Signal, Génie Informatique

par **Souheïl ZINA**

Proposition d'un cadre de modélisation pour les applications PLM Application à la gestion de configurations

Soutenue à huis-clos le 14 novembre 2007

Membres du jury :

Rapporteurs :	M. Abdelaziz BOURAS	Professeur, Université Lumière Lyon 2
	M. Benoit EYNARD	Professeur, Université de Technologie de Compiègne
Examinateurs :	M. Michel BIGAND	Maître de Conférences, Habilité à Diriger des Recherches, Ecole Centrale de Lille
	M. Pascal LHOSTE	Professeur, Institut National Polytechnique de Lorraine
	Mme. Muriel LOMBARD	Maître de Conférences, Habilité à Diriger des Recherches, Directeur de thèse, UHP – Nancy I
	M. Gabriel RIS	Professeur, UHP – Nancy I
Invité industriel :	M. Dominique PIOLLE	Directeur Général Adjoint, LASCOM – Bièvres

Centre de Recherche en Automatique de Nancy
Faculté des Sciences & Techniques - 54500 Vandœuvre-lès-Nancy

A mes parents

A mes grands-parents

Remerciements

Mes remerciements s'adressent tout particulièrement à Madame **Muriel LOMBARD** pour avoir accepté de diriger ce travail. Son expérience, ses grandes compétences et sa disponibilité ont permis l'accomplissement de cette thèse.

Je remercie chaleureusement Monsieur **Gabriel RIS,** pour m'avoir permis d'entreprendre cette thèse au sein de son équipe et pour son soutien bien chaleureux pour mes travaux.

Je remercie très sincèrement Monsieur **Dominique PIOLLE** pour sa confiance, son soutien, ses conseils et sa participation au jury de soutenance. Qu'il trouve ici les marques de ma reconnaissance.

Je tiens à exprimer toute ma gratitude à Monsieur **Abdelaziz BOURAS** et Monsieur **Benoit EYNARD**, pour l'honneur qu'ils m'ont fait d'avoir accepté d'être rapporteurs de cette thèse.

Je tiens également à remercier Monsieur **Pascal LHOSTE** qui m'a fait l'honneur de présider le jury, ainsi que Monsieur **Michel BIGAND** pour avoir accepté d'examiner ce travail avec bienveillance et de faire partie du jury.

Je voudrais également remercier vivement Monsieur **Luc LOSSENT** pour la façon efficace et amicale d'accompagner ce travail.

Un merci plus particulier à mes amis et collègues de **LASCOM**, avec lesquels j'ai passé de très bons moments durant ce travail de thèse. Je les remercie pour la bonne humeur dans laquelle ce travail a été accompli.

Merci à mes amis et collègues du laboratoire **CRAN** et de l'**AIP Lorrain** pour leur attention.

Enfin, mes pensées vont à ma famille pour m'avoir, malgré l'éloignement, soutenu et encouragé tout au long de ce travail.

Note : Pour des raisons de confidentialité, certaines parties de ce mémoire ont été retirées.

Table des matières

Introduction générale 3

Chapitre 1 : Modélisation du produit et Gestion du cycle de vie 11

1 La gestion du cycle de vie du produit 13

1.1 La gestion des processus 14

1.2 La gestion des données techniques 16

1.2.1 La structuration des données du produit 17

1.2.2 Aspect multi-point de vue 18

1.2.3 La configuration du produit 21

1.3 La gestion de configuration 22

2 La modélisation du produit 25

2.1 Approche par méta-modélisation 25

2.2 Approches par graphes de données 29

2.2.1 Approche issue du Web sémantique 30

2.2.2 Approche issue des réseaux sémantiques 32

3 Conclusion 35

Chapitre 2 : Extraction et formalisation des concepts 37

Chapitre 3 : Formalisation et déploiement des nouveaux concepts 65

Chapitre 4 : Conclusion et perspectives 85

1 Conclusion 85

2 Perspectives 86

Bibliographie 91

Annexes 97

Introduction générale

La maîtrise de l'information technique se situe au centre des préoccupations actuelles des entreprises. Les réglementations de plus en plus contraignantes et la concurrence forte demandent plus de rigueur et une réactivité accrue pour répondre aux demandes des clients. L'amélioration de la qualité des produits et la réduction des cycles et des coûts passent par la mise en œuvre de règles et de moyens de gestion des données techniques.

La gestion de celles-ci a démarré au début des années quatre-vingt-dix pour répondre aux besoins de gestion des données importantes de conception issues de la CAO (Conception Assistée par Ordinateur). Cantonnés au bureau d'étude, les outils de gestion des données techniques (PDM ou *Product Data Management*) ont permis de disposer d'une infrastructure de développement des produits. Aujourd'hui, la prise en compte de la gestion toute entière du cycle de vie du produit (PLM ou *Product Lifecycle Management*) vise à étendre ce périmètre d'action aux disciplines qui gravitent autour du produit. Le PLM permet ainsi de fédérer l'ensemble des outils de création et de gestion des données techniques et les moyens de l'entreprise ; il ne

regroupe pas seulement les éléments ayant un lien avec la conception et la gestion documentaire, mais s'attache aussi à tous les composants du système d'information permettant d'assurer le suivi du produit lors de sa fabrication, sa commercialisation jusqu'à sa disparition et éventuellement son recyclage.

Le PLM est une stratégie d'entreprise qui s'articule autour de la gestion des données techniques et la gestion des processus. Les processus assistent l'utilisateur pour la création, la consultation et la diffusion des données. L'utilisateur dispose alors d'une vision simplifiée et dynamique de la gestion des données complexes. Ainsi, la principale caractéristique d'un système PLM réside dans sa capacité à fédérer et à intégrer plusieurs aspects liés au développement et à la gestion d'un produit et de ses processus tout au long de son cycle de vie.

Sa mise en place reste un exercice difficile. En effet, l'ingénierie des systèmes PLM commence par l'analyse des besoins des entreprises pour aboutir à une solution logicielle basée sur un système technologique donné. Cependant, la diversité des applications PLM (diversité liée aux spécificités des clients), la complexité croissante des données et le besoin de systèmes évolutifs et flexibles (réagir et anticiper les changements) font qu'il n'existe pas un modèle produit universel capable de couvrir les besoins des différents clients. Le grand nombre des modèles produit proposés dans la littérature témoigne de cette diversité.

Les problématiques auxquelles les éditeurs et les intégrateurs des applications PLM doivent faire face, proviennent du caractère spécifique inhérent aux projets clients, même si la plupart des besoins fonctionnels sont souvent génériques. De par cet aspect spécifique des besoins des entreprises, la mise en place d'une solution PLM nécessite des investissements lourds notamment en charge de développement. Ces développements engendrent des délais de mise en œuvre et des ressources importants. Ces difficultés constituent actuellement des points névralgiques dans le processus de développement et de déploiement des projets PLM.

Ce mémoire présente nos travaux menés au sein de l'équipe projet « NUMérisation, COnception COopérative Produits-Processus » (NumCOP2) du groupe thématique « SYstèMes de Production Ambiants » (SYMPA) du « Centre de Recherche en Automatique de Nancy » (CRAN) et de la société LASCOM. Ce travail a été réalisé dans le cadre d'une convention CIFRE.

Les activités de recherche de l'équipe projet NumCOP2 portent sur le problème de la représentation complexe, distribuée et hétérogène des informations produit multi-points de vue. Plus particulièrement, l'action « Gestion intégrée des connaissances en conception collaborative de produit et de processus organisationnel » a pour objectif la construction d'un modèle intégré représentant complètement le produit et exploitable par tous les métiers impliqués dans son cycle de vie.

La société LASCOM est un éditeur de solutions de gestion du cycle de vie de produit. Elle se concentre sur les domaines de l'Ingénierie industrielle et du Transport, de l'Aérospatial et de la Défense, de l'Industrie Manufacturière et de l'Alimentaire-Pharma. Ses solutions, basées sur la gestion de configuration et de processus, permettent de gérer et d'assurer l'échange et le suivi de l'ensemble des informations techniques critiques dans les domaines de la gestion des projets, des systèmes et installations et des produits.

Nos travaux se placent à la frontière de l'ingénierie des systèmes d'information produit et du génie logiciel. Ils visent à définir un cadre de modélisation qui permet aux éditeurs et intégrateurs de spécifier, concevoir, réaliser et faire évoluer rapidement les applications PLM en prenant en compte les spécificités des clients. Ce travail vise à :

- capitaliser les savoirs et savoir-faire pour une meilleure maîtrise des délais de spécification et de mise en œuvre,
- faciliter l'évolution des applications et limiter les développements spécifiques,
- réduire les coûts de développement et de maintenance,
- augmenter la qualité et améliorer les performances,
- fournir une plus grande autonomie aux utilisateurs.

Il s'agit alors de formaliser les concepts de modélisation adaptés aux applications PLM et de définir un cadre pour la spécification et le déploiement adapté aux applications PLM, permettant ainsi une implémentation adaptative dans le contexte métier des utilisateurs.

Notre objectif est de permettre de spécifier et d'implémenter rapidement des applications PLM répondant à des besoins divers dans des secteurs différents en limitant au strict minimum les développements spécifiques.

Au cours de ce travail de thèse nous avons exploré plusieurs directions dans le but de cerner les problématiques et les domaines de recherche concernés par la modélisation multi-points de vue des données techniques relatives au produit et plus particulièrement sur la gestion de configuration permettant de les organiser. Ainsi le **Chapitre 1** s'appuie sur une étude bibliographique permettant de présenter les concepts nécessaires à la définition du contexte ainsi que l'étude de formalismes possibles pour répondre aux besoins d'évolution des applications PLM.

Après étude de l'architecture actuelle du progiciel Advitium™, ainsi que des techniques utilisées par la société LASCOM pour la spécification et le déploiement de son progiciel Advitium™, nous avons mis en place un cadre méthodologique pragmatique de recherche. La Figure 1 présente la démarche méthodologique suivie basée sur une approche relationnelle privilégiant les données.

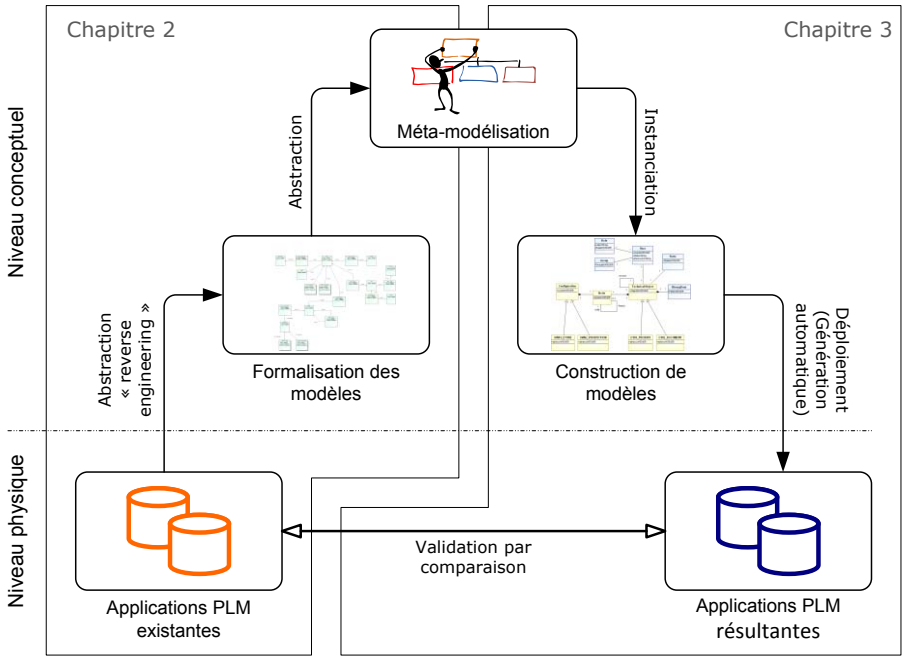

Figure 1. Démarche de notre étude

Une première étape de notre étude effectuée chez LASCOM a consisté, à partir d'un ensemble d'applications PLM existantes, à mettre en place une démarche de « reverse engineering » qui a permis de formaliser les concepts utilisés dans Advitium™ au sein d'un méta-modèle UML. Le déroulement de cette étape est présenté au **Chapitre 2**. Ainsi, il s'agit d'introduire progressivement les concepts

constituants le méta-modèle du progiciel Advitium™ en montrant également les relations entre eux ainsi que leur utilisation au travers de diagrammes d'états. Pour ce faire, nous avons développé dans un premier temps un outil de rétro-modélisation des applications PLM existantes. Un extrait du méta-modèle obtenu est présenté à la Figure 2. Sa dérivation automatique a permis de le comparer puis de le valider en regard des concepts manipulés au niveau du schéma relationnel du progiciel Advitium™ permettant ainsi de valider la boucle de rétroaction mise en place.

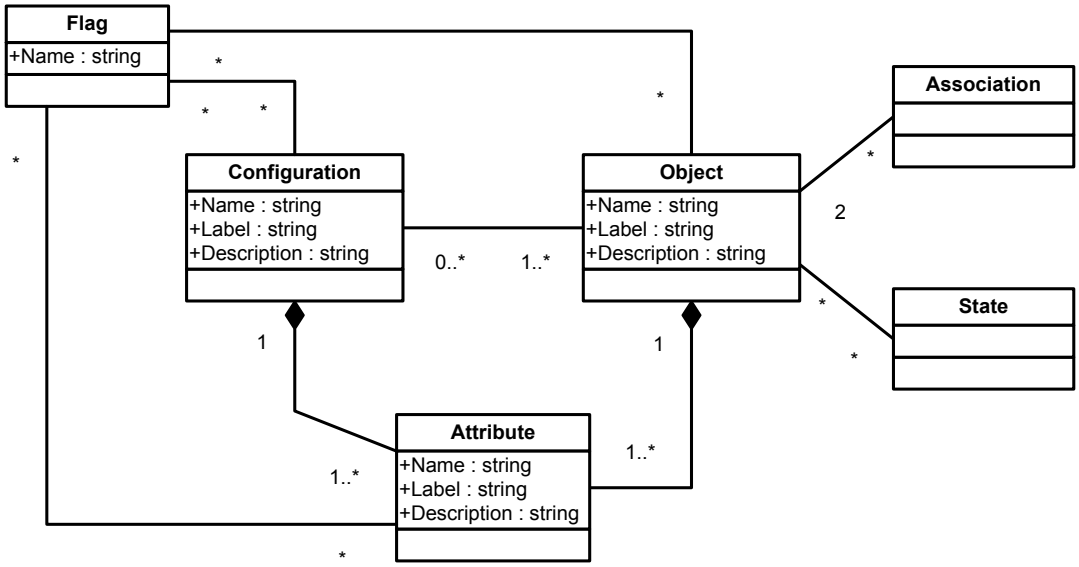

Figure 2. Extrait du méta-modèle UML des applications PLM

Néanmoins, la structure ainsi obtenue présente des incomplétudes en regard de la formalisation sémantique nécessaire des objets techniques et liens existants entre eux pour répondre à :

- un souci de présentation des objets techniques à l'utilisateur,
- une facilité d'accès lors de recherches et une optimisation des traitements sur les éléments de configuration
- une optimisation des liens existants dans la base de données en regard de l'étude sémantique menée au niveau conceptuel à partir du cahier des charges du client.

Aussi, la seconde étape présentée au **Chapitre 3** est centrée sur les spécificités liées à l'organisation des données techniques au travers de la gestion de configuration de produits (structure hiérarchique conditionnelle) et les liens entre

elles au niveau physique. La mise en correspondance des modèles conceptuels avec les informations du niveau physique nous a permis de cerner les manques dans la définition des concepts utilisés et d'optimiser les structures relationnelles en regard des concepts utilisés et de leurs traitements associés.

La Figure 3 présente un exemple de configuration montrant l'ensemble des objets techniques constituant la configuration d'un produit et qui peut être représenté à l'aide d'une structure de graphe (DAG ou *Directed Acyclic Graph*). Chaque graphe est constitué par un ensemble de nœuds (objets techniques), les nœuds sont reliés entre eux par des liens. Les liens comportent des propriétés (type du lien, dates de début et de fin de validité, etc.) permettant de décrire plus précisément les relations entre les objets et autorisant, ainsi, la traçabilité des évolutions des configurations du produit.

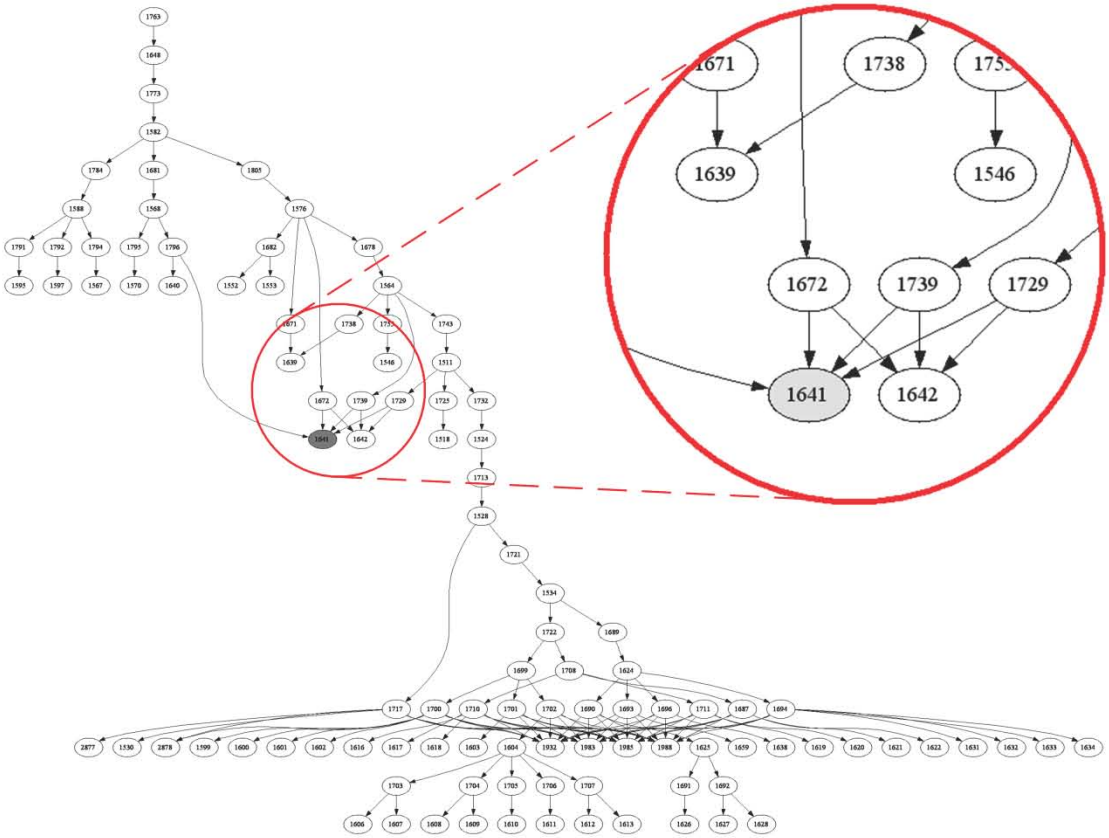

Figure 3. Exemple d'un graphe de configuration

En regard de cette problématique, le **Chapitre 3** présente également les évolutions proposées et développées au sein du méta-modèle du progiciel Advitium™ ainsi que le déploiement nécessaire de ce dernier pour aboutir à l'application souhaitée en

regard des spécifications du client. Il s'agit de prendre en compte la notion de contexte dans la gestion de configuration, mais aussi, la définition d'un formalisme permettant d'adapter les formulaires d'objets techniques aux besoins exprimés par les utilisateurs.

Enfin, le **Chapitre 4** présente des conclusions et perspectives ouvertes par nos travaux, d'un point de vue conceptuel, technique et méthodologique. En effet, ce travail constitue une première étape dans la définition sémantique d'un cadre de modélisation d'applications PLM à destination des développeurs du progiciel. Il convient par la suite d'envisager un accompagnement au travers de méthodologies de déploiement à destination des concepteurs de manière à faciliter leur travail par réutilisation de tout ou partie d'applications métiers génériques existantes de manière à simplifier les développements et à raccourcir les délais de livraison des applications PLM.

CHAPITRE 1

Modélisation du produit et Gestion du cycle de vie

La maitrise des données de l'entreprise a toujours été une préoccupation essentielle. Il s'agissait dans un premier temps d'organiser les données et de les gérer dans un système informatisé. Puis, la prise en compte des changements organisationnels de l'entreprise basée sur un processus synchrone puis asynchrone synchronisé de ses activités a nécessité d'envisager le cycle de vie du produit non plus comme un processus séquentiel mais comme un processus concourant [Lombard, 1994]. Les concepts du CIM (*Computer Information Manufacturing*) s'attachant à l'intégration des données puis de CIME (*Computer Information Manufacturing Engineering*) permettant d'envisager sa mise en œuvre dans les années 80, puis une

interprétation du CIM comme *Corporate Information Management* dans les années 90 ont positionné les bases et la problématique de la modélisation du produit dans le cadre plus général de la maitrise globale des informations présentes dans la gestion du cycle de vie [Lombard, 2006].

Apparues tout d'abord dans les secteurs de l'aéronautique et de la défense pour répondre à la nécessité de gérer des volumes de données croissants dans des environnements toujours plus complexes, la gestion du cycle de vie du produit (PLM ou *Product Lifecycle Management*) s'étend aujourd'hui à des domaines très variés. En effet, la traçabilité dans l'industrie agro-alimentaire, ou encore les problèmes de compatibilité et de respect des règles d'homologation dans l'industrie pharmaceutique, constituent autant de problématiques dont la réponse passe aujourd'hui par la maitrise de l'ensemble des informations qui décrivent le produit tout au long de son cycle de vie. Si les liens sont étroits avec les Systèmes de Gestion des Données Techniques (SGDT), le PLM a néanmoins un caractère d'ouverture à l'extérieur du bureau d'études. En effet, le PLM ne s'arrête pas à la phase de conception et ne se limite pas aux seules données techniques telles que plans et schémas. Les définitions de produit et par conséquent les données industrielles, ne partent pas seulement du bureau d'études mais aussi des services en amont et en aval. Le problème de la gestion du cycle de vie du produit ne se résume pas à extraire des informations du bureau d'études pour l'ensemble de l'entreprise, mais de disposer d'une infrastructure technique transversale à l'entreprise permettant de gérer plus efficacement la masse de données associées au produit et ainsi de recouvrer l'intégralité de son cycle de vie.

D'un point de vue structurel, le PLM est souvent défini comme un ensemble de fonctions et procédures qui permettent de gérer et d'exploiter les données définissant à la fois les produits et les processus mis en œuvre pour leurs développements. En effet, la gestion du cycle de vie de produit par une solution PLM permet d'englober tous les éléments nécessaires à sa traçabilité, comme la modélisation, la gestion documentaire, l'analyse numérique, la capitalisation des savoir-faire, etc., mais aussi tous les composants du système d'information permettant d'assurer le suivi du produit lors de sa fabrication, sa commercialisation jusqu'à sa disparition et éventuellement son recyclage. Mais au-delà des systèmes informatiques qui visent plus efficacement la masse des données de l'entreprise, le concept inclut surtout

toutes méthodologies permettant de travailler de façon simultanée sur un produit ou un projet, de faire collaborer des équipes distantes, d'assurer la diffusion d'une information valable et à jour à tous les intervenants (bureau d'études, bureau d'industrialisation, fabrication, après vente, voire fournisseurs et clients) où qu'ils se trouvent géographiquement. Au cours des dernières années, les nouvelles Technologies de l'Information et de la Communication (TIC) ont eu des répercussions considérables sur les outils et systèmes PLM, notamment par le grand nombre des utilisateurs et l'environnement distribué des entreprises. De plus, les technologies Web ont contribué fortement à étendre le périmètre d'utilisation du PLM.

Ainsi, ce chapitre présente la gestion du cycle de vie du produit et plus particulièrement porte sur la modélisation du produit. Il est organisé en deux parties, une première partie présente l'historique et l'évolution de la modélisation du produit et de la gestion de son cycle de vie, une deuxième partie aborde différentes approches novatrices pour sa modélisation en tenant compte des multiples points de vue des acteurs impliqués dans son cycle de vie, dans l'objectif de mieux gérer les configurations du produit. Après avoir défini les besoins en termes de modélisation de configuration, nous présentons certaines approches et pratiques industrielles de manière à mettre en exergue certains formalismes. De manière à mieux gérer les configurations, nous nous plaçons dans un cadre de méta-modélisation. En réponse à notre préoccupation, nous choisissons le formalisme sNets, basé sur les réseaux sémantiques, car il intègre un système de typage et une gestion de la modularité le rendant réflexif. Néanmoins, nous complétons cette réponse au chapitre 3 par l'extension de ce formalisme pour la prise en compte des multi-points de vue.

1 La gestion du cycle de vie du produit

Les outils de gestion des données techniques (PDM ou *Product Data Management*) ont permis jusqu'à présent de disposer d'une infrastructure de développement des produits [Stark, 2005]. Le PLM vise maintenant à étendre ce périmètre d'action aux disciplines qui gravitent autour du produit. En effet, la gestion des données techniques a démarré au début des années 90 pour répondre aux besoins de gestion des données de conception issues de la CAO. Ensuite le besoin des entreprises est passé de la gestion des données à la définition d'un produit durant tout son cycle de

vie ; puis l'approche technique a été dépassée pour couvrir l'ensemble des enjeux stratégiques de l'entreprise.

Une des définitions communément admise pour le PLM est celle donnée par CIMdata[1] « *une approche stratégique d'entreprise, qui applique un ensemble de solutions pour soutenir dans un mode collaboratif la création, la gestion, la dissémination et l'utilisation de l'information de définition des produits, en entreprise étendue, du concept à la fin de vie, en intégrant les personnes, les processus, les systèmes et l'information* » [CIMdata, 2002]. Le périmètre du PLM esquissé à travers cette définition est très large. Il inclue ainsi une approche stratégique du développement de produit et de ses processus, les outils mis en œuvre dans la création ou la diffusion de données, le travail collaboratif, etc.

Il est à noter que le terme PLM n'inclut que la notion de gestion (*management*) des données et non leur création qui relève d'activités et d'outils métier spécifiques. Ainsi, dans cette étude, nous abordons brièvement la présentation de la gestion des processus pour mieux nous concentrer sur notre problématique portant sur la maîtrise des concepts de bases et traitements nécessaires pour implémenter et gérer les données dans les applications PLM dédiées aux besoins des clients.

1.1 La gestion des processus

La gestion du cycle de vie de produit s'articule autour de la gestion des données techniques et la gestion des processus. A l'instar d'Advitium™ qui génère des interactions fortes entre ces deux composantes permettant leur enrichissement mutuel : « *les processus fabriquent les données et les données font avancer les processus* ». Ce postulat usité par LASCOM permet de situer l'intégration des données et des processus et illustre l'impact qu'ils peuvent avoir entre eux.

Un processus permet, à partir d'un modèle de processus, de guider le déroulement d'une suite de tâches métiers (validation de données, modification de produit, gestion de conflits, etc.), de coordonner les actions à réaliser et de capitaliser les différentes actions menées. Cette capitalisation est nécessaire pour pouvoir réutiliser des savoir-faire et garder trace des divers choix effectués par les acteurs sur les produits tout au long de leur cycle de vie. Plusieurs typologies de systèmes processus ont été

[1] Cabinet d'analyse et de conseil international spécialisé en PLM (http://www.cimdata.com).

décrites dans la littérature [Van der Aalst et Van Hee, 2004]. C'est à partir de la définition des modèles de processus et de produit réalisés par les concepteurs qu'est proposée aux utilisateurs une implémentation physique d'une application PLM (Figure 4).

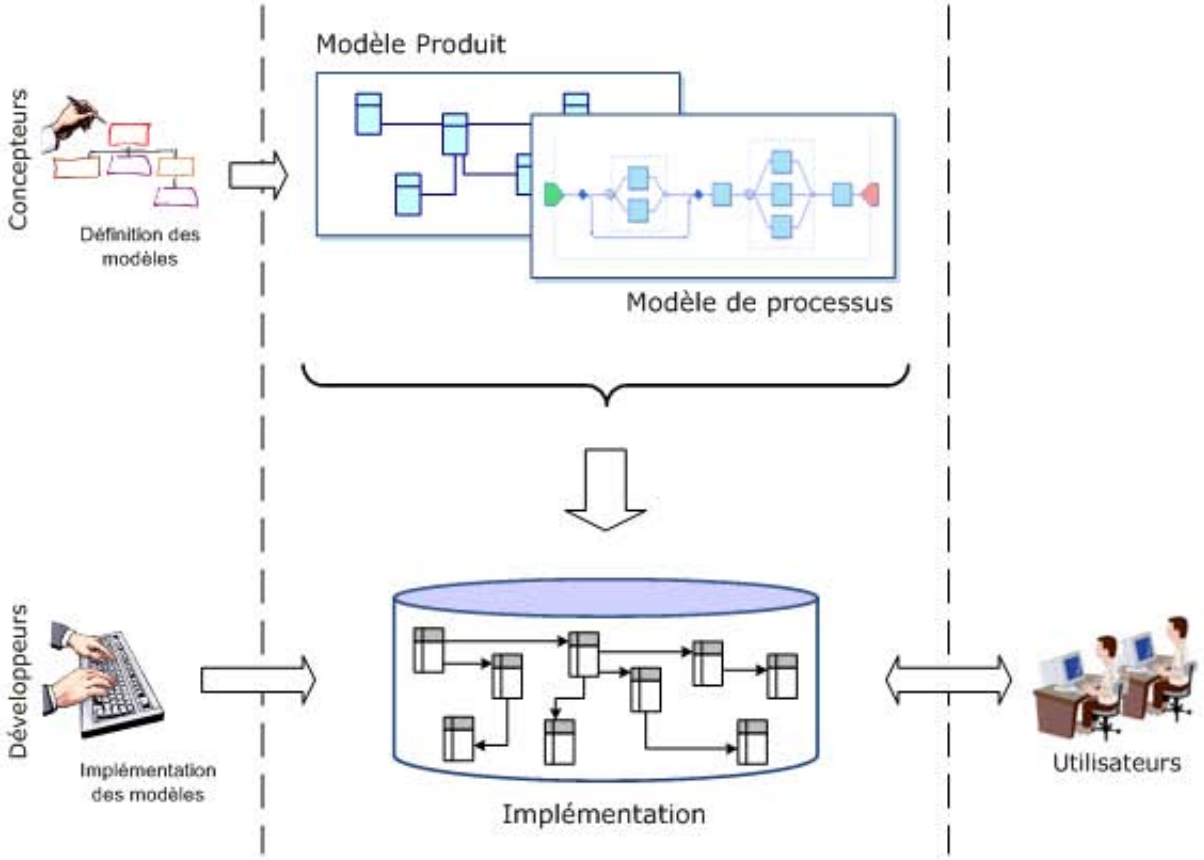

Figure 4. Modèle Produit et Modèle de Processus

Des travaux tels que ceux de [Harani, 1997] proposent un modèle de processus associé à un modèle de produit. La modélisation ainsi proposée s'appuie sur quatre concepts principaux : le concept processus (description des différentes étapes de développement du produit en identifiant les tâches et leur mode d'enchainement), le concept tâche (de type élémentaire ou composite permettant la représentation du déroulement des étapes composant le processus), le concept d'état (concernant le produit, le processus et les tâches) et le concept ressource (matériel ou humain). Un couplage entre le modèle de processus et le modèle de produit (gestion de l'ensemble des données techniques) est également proposé afin d'assurer la cohérence de l'approche proposée.

1.2 La gestion des données techniques

Le terme de donnée technique est utilisé pour designer un élément d'information servant de base à un raisonnement sur lequel les traitements sont effectués (Figure 5). Les données techniques peuvent être relatives à la conception, la fabrication, la maintenance, l'utilisation, le recyclage et le marketing [Bernard, 2000]. Un objet technique est généralement défini comme un objet d'entreprise sur lequel sont exprimés des besoins en gestion de données qui sont relatifs à la gestion et à la manipulation des objets complexes.

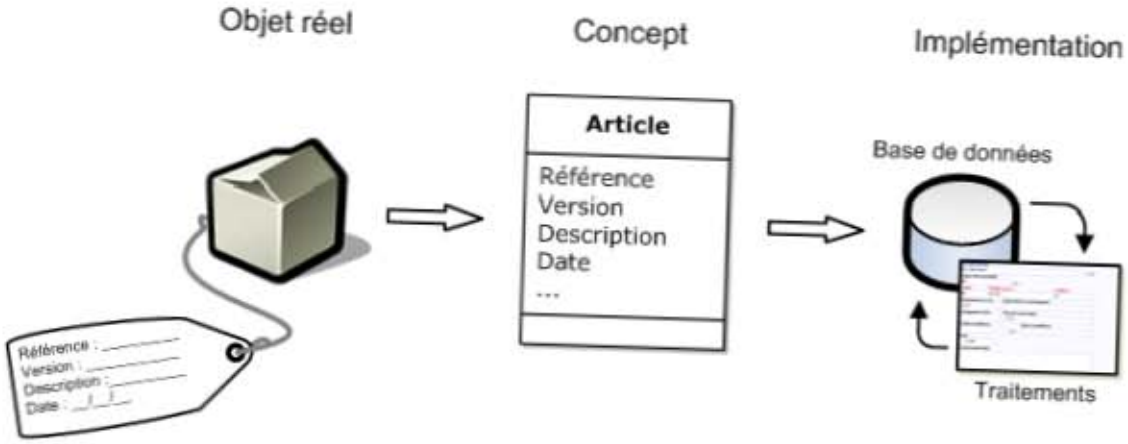

Figure 5. Le concept d'objet technique

La gestion des objets techniques se fait au travers d'un Système de Gestion de Données Techniques (SGDT) qui est un outil permettant de gérer les données et les processus de développement de produit. De plus, il conserve et hiérarchise toutes les données et informations nécessaires à la définition du produit.

Pour répondre à cette problématique de gestion, deux grands courants co-existent, l'un est issu du domaine de la CAO où les outils proposés sont bâtis autour des spécificités des données issues des activités de conception, l'autre se situe autour de la gestion des données techniques au sens large où les outils proposés s'attachent à la gestion des données et non à leur création qui relève d'outils experts métier. Ainsi, le terme produit s'applique aussi bien aux composants élémentaires (ex. : un boulon ou une vis, etc.) qu'aux entités très compliquées (ex. : un yaourt, un lanceur, un satellite, etc.). Le boulon ou la vis posent peu de problèmes, ils sont loin d'atteindre la complexité des données associées à un yaourt ou à un satellite. En

effet, la fabrication d'un produit laitier comprend la prise en compte des ingrédients de base, mais elle exige aussi et surtout le respect des règles de sécurité, d'hygiène, de traçabilité, des législations liées à l'alimentation, des descriptions d'allergènes. Ici, pas de CAO, mais une très grande précision dans la description des produits, qui doit pouvoir servir de preuve légale dès que nécessaire.

1.2.1 La structuration des données du produit

Un des intérêts majeurs des SGDT réside dans leur capacité à structurer et gérer l'évolution des différentes données tout au long de leur cycle de vie, tout en maintenant la cohérence et l'intégrité des informations [Eynard, 2005]. Les SGDT sont caractérisés par les fonctionnalités de gestion de documents et de gestion des structures du produit. Un SGDT gère des objets de plusieurs types et les fichiers associés à ces objets (modèles CAO, fichiers bureautiques, etc.). Chaque type d'objet est identifié par son propre jeu d'attributs.

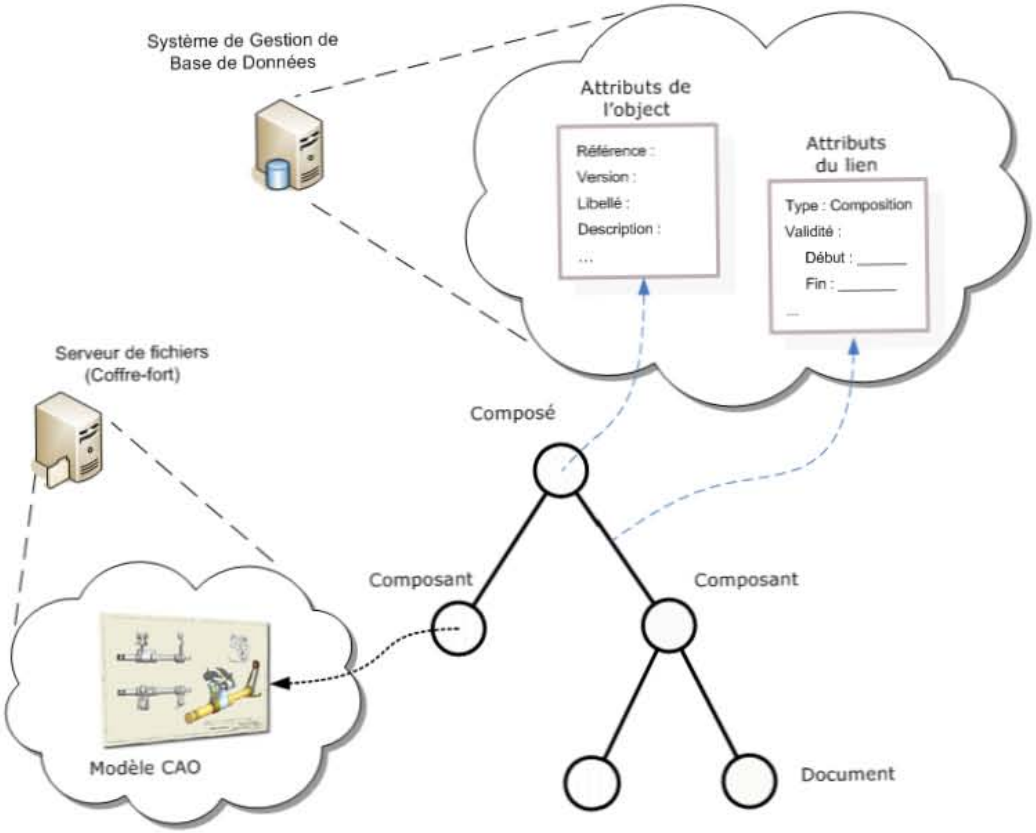

Figure 6. Structuration des données dans un SGDT

La structure du produit est modélisée sur la base de graphes. Ces graphes sont composés de nœuds d'objets métiers (documents, articles, pièces, etc.). Ces objets

sont reliés entre eux par des liens de différents types et porteurs d'informations (type du lien, dates de validités, etc.). Un lien de composition, par exemple, sert à lier un objet à ses constituants (Figure 6). Ce type de lien permet d'établir et de gérer une nomenclature de produits [Maurino, 1993].

Les documents sont considérés comme des objets techniques identifiés par un ensemble d'attributs caractéristiques comme le nom du document, sa version, son statut dans le cycle de vie, le nom de l'auteur etc. [Eloranta et *al.*, 2001]. Ces documents, relevant d'outils expert métier, sont stockés dans un coffre-fort. L'accès à ces objets techniques est contrôlé par des fonctionnalités de base telles que la gestion des évolutions des objets (mécanismes d'extraction et de rangement) et donc de leurs fichiers associés, la gestion des versions et des états, la gestion des liens entre objets. La gestion des données techniques s'appuie également sur des Systèmes de Gestion de Base de Données Relationnelle (SGBDR) standards qui assurent un stockage centralisé des données du produit. Cette centralisation permet de conserver l'unicité, l'accessibilité et la sécurité des informations.

A partir de ces éléments, il est possible de décomposer le produit selon une structure fonctionnelle, organique, technique, etc. [Bigand, 2005]. Bien que la décomposition puisse être arborescente, la création de liens transversaux est envisageable [Maurino, 1993]. Ces liens peuvent permettre de maîtriser la propagation d'une modification par exemple.

1.2.2 Aspect multi-point de vue

La prise en compte de différentes expertises métier par la définition de différentes structures produit induit la nécessité de pouvoir proposer à chaque utilisateur les informations relevant de son domaine d'intérêt. De manière à pouvoir gérer ces multiples structures, les concepts de vue et de point de vue ont été introduits par des travaux en implémentation de bases de données puis des travaux en modélisation de systèmes d'information et plus particulièrement autour de la modélisation de produit [Harani, 1997] [Million, 1998] [Nöel, 2005].

La notion de point de vue, classiquement utilisée dans la littérature, a pour principal objectif la description d'une entité complexe ayant plusieurs facettes. Les points de vue permettent de structurer l'information à partir de différents critères, soit métier, soit produit, pour la rendre plus représentative, plus compréhensive et donc

concourent à la fourniture de vues externes particulières du produit, qui correspondent à des besoins spécifiques : vue d'étude, vue de production, vue maintenance, etc.

L'approche multi-points de vue est basée sur l'élaboration d'un modèle unique à partir de différents besoins en modélisation. Ce modèle unique peut être construit par itération successive ou par fédération de modèles partiels. Ce modèle unique est accessible selon plusieurs points de vue, ce qui permet alors de centraliser les données du modèle. La notion de point de vue associée aux utilisateurs donne la possibilité de voir le modèle selon plusieurs points de vue. Ceci revient à donner des « éclairages » différents sur la même information (Figure 7), ce qui permet à l'utilisateur de se focaliser sur une partie (ou sous-modèle) du modèle global spécifique à son besoin. L'avantage majeur de l'utilisation d'un modèle unique est que les modifications apportées à un sous-modèle sont répercutées dans les autres sous-modèles. Par conséquent, les problèmes d'incohérence dus au partage des données entre les modèles partiels sont évités [Nassar, 2005]. Cette approche est très utilisée dans les travaux sur la représentation de produit dans les systèmes CAO [Derigent et Lombard, 2006].

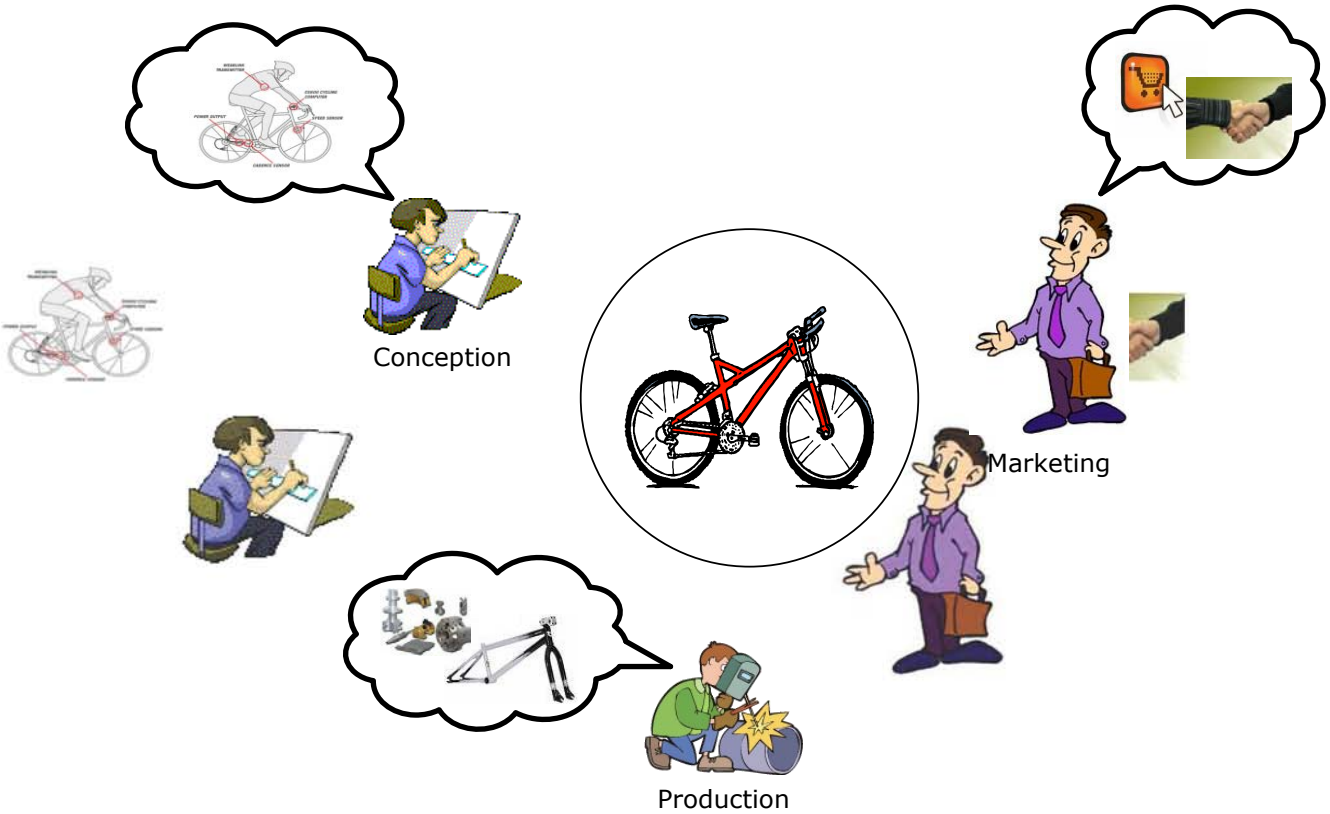

Figure 7. Notion multi-points de vue sur le produit

Les travaux de [Million, 1998] portent sur la problématique de la conception de systèmes d'information techniques dans un contexte industriel multi-acteurs. La méthode proposée, appelée VIM (*Viewpoints Information Modelling*), permet de construire un modèle final porteur de l'ensemble des points de vue métier par ajustements successifs du modèle initial centré sur l'objet technique en regard des points de vue considérés.

Dans le cadre de nos travaux sur les approches de modélisation de produit multi-points de vue, nous retenons en particulier les travaux de [Harani, 1997] qui reposent sur le concept de méta-modélisation, et qui visent à représenter et regrouper toutes les informations de conception définissant et caractérisant le produit. Dans le modèle proposé Figure 8, le produit possède un certain nombre d'attributs qui le caractérisent, mais aussi des « point-de-vue », « état », « paramètre » et « description ». Chaque point de vue est modélisé comme un graphe hiérarchisé avec des liens (de composition, de spécialisation ou d'appartenance) entre les nœuds. Des liens peuvent exister entre nœuds de graphes différents (association d'une fonction à un composant structurel).

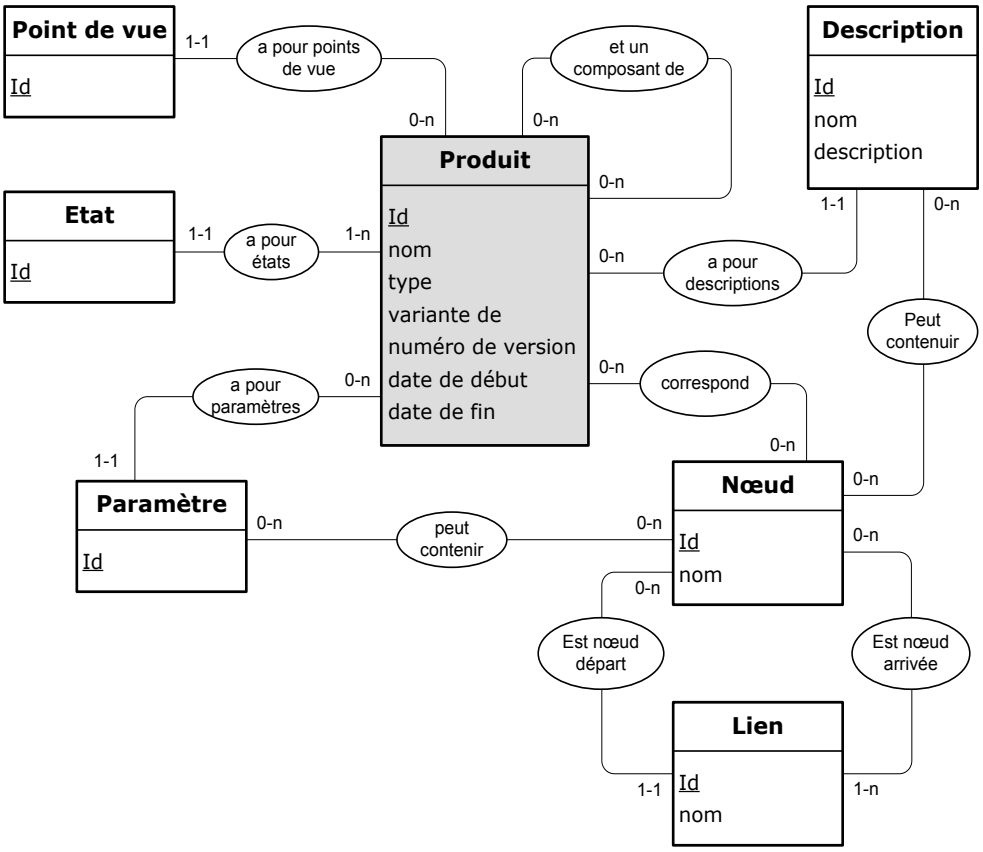

Figure 8. Méta-modèle du concept produit [Harani, 1997]

Dans ce modèle, si le produit peut posséder plusieurs points de vue, il n'est pas aisé de répondre aux problèmes de cohérence entre les points de vue. En effet, un objet dans un point de vue peut être décomposé ou résulter des traitements dans un autre point de vue. Cet aspect de propagation entre points de vue est abordé dans les travaux de [Bettaieb, et *al.*, 2003] pour les modèles CAO.

1.2.3 La configuration du produit

Les structures arborescentes définies précédemment s'apparentent à des configurations de produit telles que proposées dans le progiciel Advitium™.

Une configuration consiste à produire différents arrangements des données constituant le produit selon différents besoins. L'équipe Product Data Management Group[2] propose un modèle de configuration attaché à une structure de produit générique [Männistö et Sulonen, 1999]. Ce modèle décrit les composants qui peuvent être inclus dans la configuration. Le modèle de configuration comprend également les règles de combinaison de ces composants. Une configuration représente donc une description précise du produit, qui satisfait les besoins et qui est validée par le respect des règles de combinaison définies dans le modèle.

Un processus de configuration permet de fournir une structure de produit spécifique en s'appuyant sur la structure générique et les exigences définies dans le modèle de configuration. Plusieurs méthodes, basées notamment sur les techniques d'intelligence artificielle, offrent aujourd'hui des pistes pour la résolution des problèmes de configuration. Il existe aussi plusieurs outils, appelés configurateurs, tel que *EngCon* [Hollmann et *al.*, 2000] et *WeCoTin* [Asikainen et *al.*, 2004] permettant d'obtenir des configurations précises du produit. Ces outils peuvent être intégrés dans les applications PLM pour assister l'utilisateur dans la configuration de produit [Ackermann, 2005]. En effet, dans les applications PLM, les données techniques sont organisées au sein de configurations. La gestion de configuration est utilisée pour des raisons à la fois de complexité des produits et de diversité des connaissances issues de divers métiers de l'entreprise. Le nombre croissant des utilisateurs des applications PLM, le volume des données techniques et les diverses évolutions associées à ces données exigent :

[2] http://www.soberit.hut.fi/pdmg/

- la maîtrise et le contrôle de la cohérence, l'unicité et la sûreté des données échangées,
- la prise en compte des évolutions des données et de l'ensemble de leurs répercussions sur le produit et ses constituants.

C'est pourquoi, la gestion de la configuration s'impose comme composante fondamentale dans les applications PLM, permettant de mieux maîtriser et gérer la complexité liée aux données.

L'intérêt majeur de la gestion de configuration est sa capacité à superviser la totalité du cycle de vie d'un produit. Elle permet de formaliser et de présenter de manière claire et complète la configuration du produit à un instant et à un état d'accomplissement donnés. Dans nos travaux, nous nous intéressons à la gestion de configuration, qui consiste à gérer les informations qui décrivent le produit, en particulier sa décomposition en sous-ensembles et pièces élémentaires mais aussi toutes les informations et caractéristiques fonctionnelles et physiques liées au produit.

A l'instar des outils de configuration de produit, les outils de gestion de configuration sont aussi basés sur des modèles produit permettant de formaliser les objets, les règles et les liens et qui doivent êtres gérés et tracés tout au long du cycle de vie du produit

1.3 La gestion de configuration

La gestion de configuration est une discipline de gestion qui consiste à appliquer des règles techniques et administratives au développement, à la production et à la maintenance, dans tout le cycle de vie du produit. Elle consiste à gérer la description technique d'un système (et de ses divers composants), ainsi qu'à gérer l'ensemble des modifications apportées au cours de l'évolution du système. En effet, les outils de gestion de configuration intègrent des fonctions et des mécanismes permettant d'auditer et de contrôler toutes les actions effectuées sur la configuration d'un produit. Ces fonctions permettent, par exemple, de gérer les demandes de modification du produit à faire évoluer, elles permettent aussi de mettre en correspondance les demandes de modification avec les changements apportés au produit. Ainsi, toute modification de configuration suit une procédure formelle impliquant notamment des approbations préalables.

La norme ISO 10007 [ISO, 2003a] présente des recommandations pour l'utilisation de la gestion de la configuration dans l'industrie. Elle fournit le processus, l'organisation et les procédures détaillées de gestion. Selon cette norme, la gestion de la configuration fait partie intégrante du PLM, elle permet de fournir une vision claire de l'état d'une configuration, d'un produit ou d'un projet, et de leurs évolutions en garantissant une traçabilité totale. Les outils de gestion de configuration intègrent des fonctions et des mécanismes permettant d'auditer et de contrôler toutes les actions effectuées sur la configuration du produit.

Dans le secteur de l'aérospatial, la gestion de la configuration a pour objectif de formaliser et de présenter de manière claire et complète la configuration du produit à un instant donné et l'état d'accomplissement des exigences physiques et fonctionnelles. Elle a également pour objectif d'aider les acteurs impliqués dans l'affaire ou le projet, tout au long du cycle de vie du produit, à disposer d'une documentation correcte et exacte.

Le système de gestion de la configuration doit permettre aux acteurs de :

- connaître à tout instant, la description technique du système et de ses constituants, par des informations contenues dans des documents approuvés, pour maîtriser la conception, qualifier la définition et pour être en mesure de mettre à disposition les données de définition nécessaires pour réaliser les produits attendus,
- contrôler que cette documentation et les produits qu'elle décrit sont en permanence compatibles entre eux pendant toute la durée de l'affaire ou du projet,
- identifier la configuration de référence et maîtriser de manière efficace et continue les évolutions de cette description technique et leur cohérence (approbation des évolutions, application sur les produits, introduction dans la documentation), afin de limiter les risques techniques et les impacts sur les coûts et les délais,
- identifier la configuration applicable et la configuration réalisée pour être en mesure de traiter les écarts décelés lors de la réalisation et à la livraison des produits.

L'élaboration d'un produit nécessite l'implication de plusieurs métiers et l'intervention de plusieurs acteurs de spécialités très différentes (Client, Commercial, Concepteur, Fabricant, etc.). En effet, dans une collaboration multidisciplinaire, les acteurs ont besoin d'intégrer leurs points de vues sur le produit tout au long de son cycle de vie [Geryville et *al.*, 2006]. Chaque acteur manipule une vue particulière du produit qui correspond à ses besoins spécifiques. Le produit est perçu selon différents points de vue qui déterminent la nature des éléments de structuration et de configuration du produit. Plusieurs configurations et structures différentes sont associées au produit.

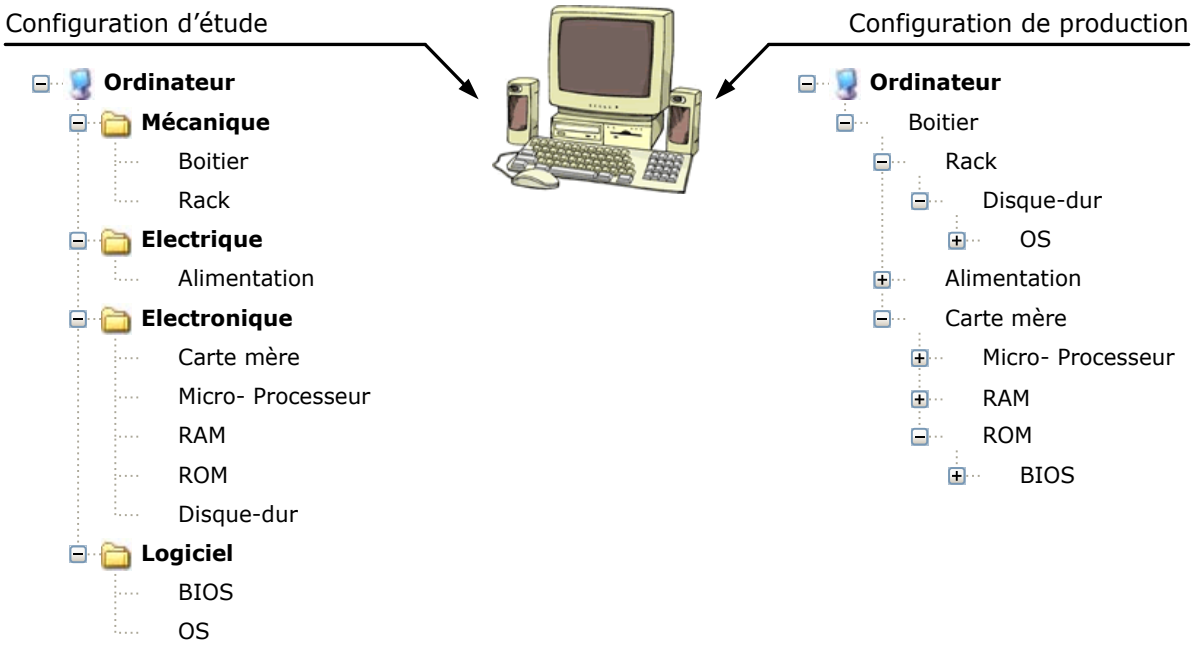

Figure 9. Exemple de configuration d'études et de production.

La Figure 9 présente un exemple de configuration d'un ordinateur perçu selon les points de vue « Etude » et « Production ». Nous parlons donc de « Configuration d'étude » regroupant les éléments qui le constituent, ceux-ci étant regroupés par familles technologiques, de « configuration de production » structurant les éléments en fonction. Dans les deux arborescences, les liens sont porteurs d'informations et contribuent à la définition du produit.

A tous les niveaux de ces arborescences produit, la gestion de configuration permet aux différents acteurs, de connaître en permanence la définition, les cas d'emploi de chaque élément du produit et, en cas d'anomalies, les exemplaires concernés.

2 La modélisation du produit

Dans les travaux sur les approches de modélisation, Lemesle [Lemesle, 2000] présente la modélisation comme « une activité qui consiste à transformer des descriptions informelles de la réalité en des descriptions formelles, appelées modèles, afin de pouvoir opérationnaliser les connaissances. Le modèle est une représentation abstraite de la réalité au sens où il simplifie cette réalité en vue d'une utilisation précise ». Le modèle est donc une vue particulière mais pertinente de la réalité. Il permet de simuler le fonctionnement de l'élément étudié, l'information modélisée pouvant être de n'importe quelle nature.

2.1 Approche par méta-modélisation

De manière à garantir la conformité des modèles, il faut se référer à un vocabulaire et une grammaire regroupés dans un méta-modèle. En effet, un modèle possède une sémantique précise qui est définie dans un autre modèle, de niveau d'abstraction plus élevé, appelé méta-modèle. Le MOF (*Méta-Object Facility*) [Gardner, et al. 2003] permet de définir 3 niveaux d'abstraction (Figure 10), chaque niveau d'abstraction étant décrit à l'aide du niveau supérieur. Le niveau le plus bas (M0) correspond à celui des instances, soit le niveau physique.

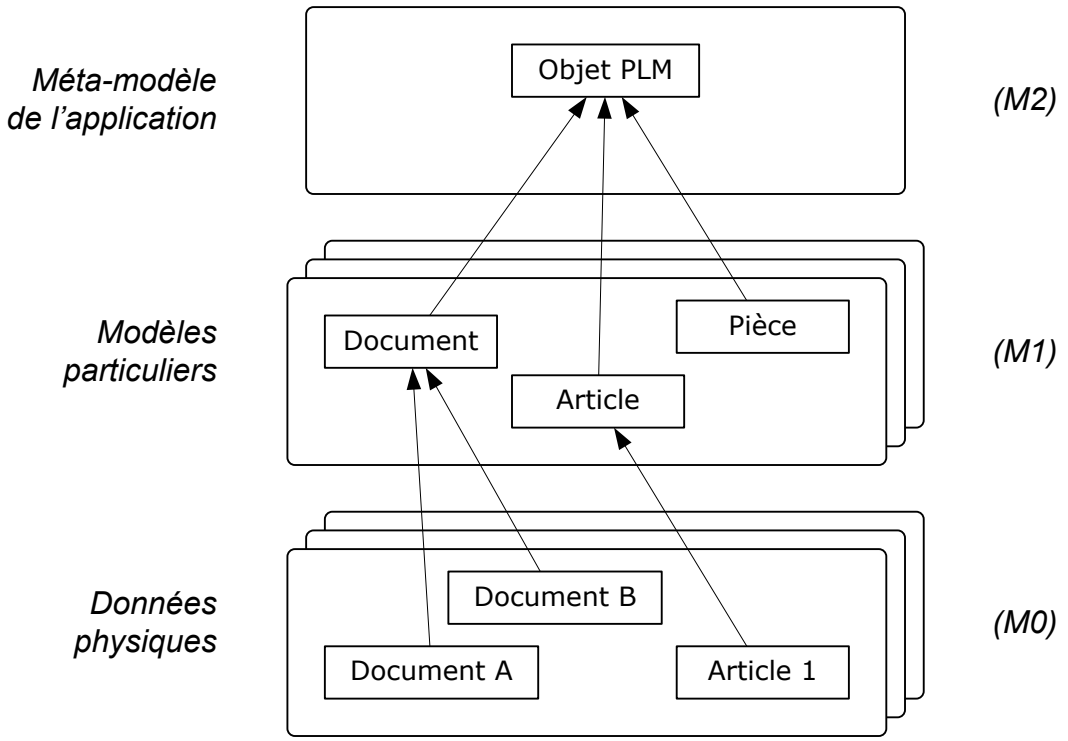

Figure 10. Le MOF (Meta-Object Facility)

Si la Figure 10 présente bien les différents niveaux d'abstraction ainsi que les différents types objets pouvant s'y trouver, il demeure à expliciter les différents liens et types de liens pouvant exister entre ces niveaux de manière à garantir aux concepteurs d'applications la conformité et la pérennité nécessaires à leurs applications.

Une utilisation particulière de ce cadre de méta-modélisation intégrant la notion de multi-points de vue nécessaire pour la modélisation de produit dans les applications PLM est présenté dans les travaux de [Nassar, 2005] qui s'attache à développer, à partir du méta-modèle UML (M2) [Booch et *al.*. 1998], une modélisation par profils (M1).

Les profils UML permettent de spécialiser, en les étiquetant par des « stéréotypes », les concepts existants dans le méta-modèle (M2). Ainsi, dans ses travaux visant à modéliser un système logiciel par une approche combinant objets et points de vue, [Nassar, 2005] propose une extension d'UML, appelée VUML (View based Unified Modeling Language). Cette extension propose le concept de classe multi-vues associée au stéréotype view. Cette classe est une unité d'abstraction et d'encapsulation qui permet de stocker et restituer l'information en fonction du profil de l'utilisateur. Elle offre des mécanismes de gestion des droits d'accès aux informations, de changement dynamique de points de vue et de gestion de la cohérence entre les vues dépendantes. Une classe multi-vues est composée d'une classe de base et d'un ensemble de vues. Ces vues sont reliées à la classe de base via une relation de dépendance particulière (extension) et peuvent être également reliées à d'autres vues par des relations de dépendances classiques (pour gérer la cohérence). Chaque vue est associée à un acteur unique.

La Figure 11 décrit un exemple de modèle VUML simplifié d'un concessionnaire de voitures. Dans ce diagramme de classes, il y a 3 classes multi-vues : Voiture, Constructeur, Accident. La classe Voiture est dotée de 3 vues correspondant aux points de vue du Maintenicien, du Commercial et du Client. Pour la classe Constructeur, seules les vues associées au Maintenicien et au Commercial sont pertinentes. Un patron de génération de code pour le profile VUML à été proposé dans [Crégut et *al.*, 2005].

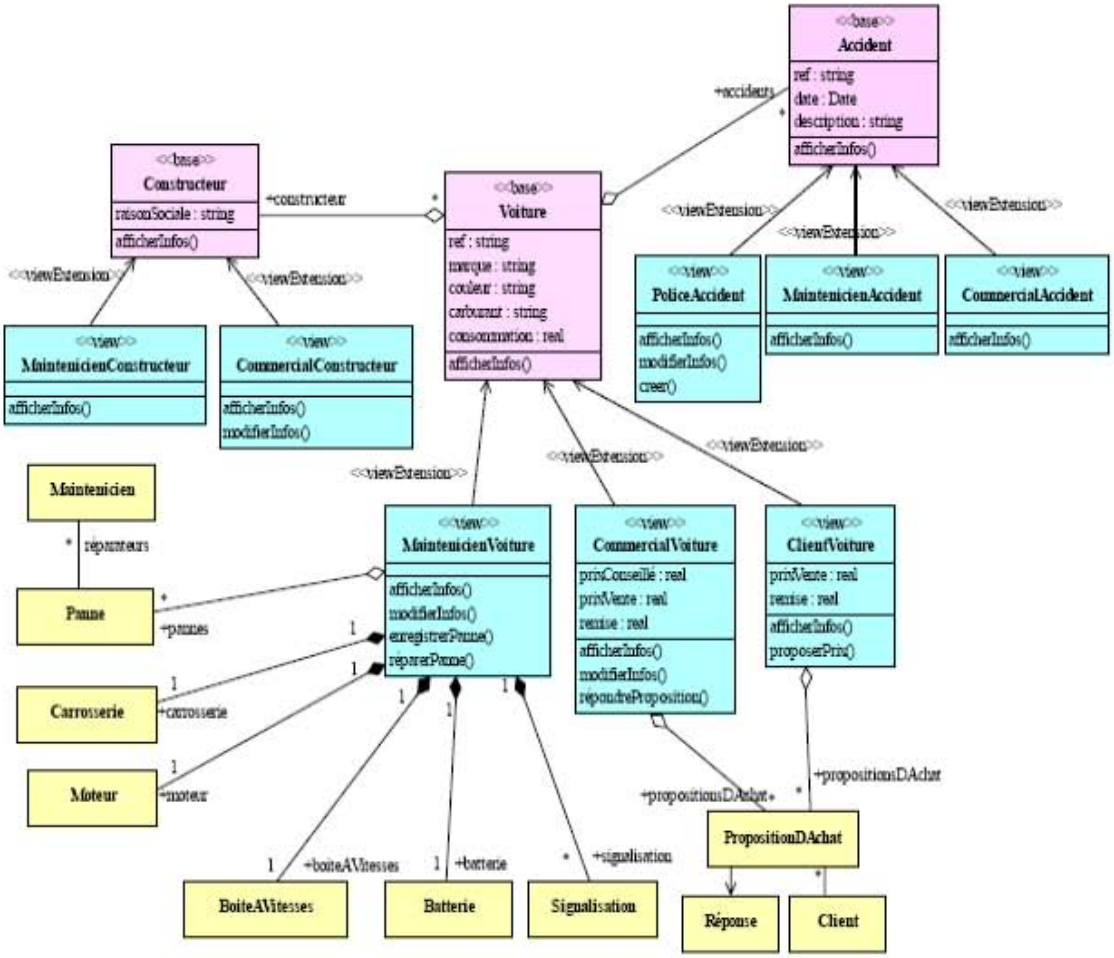

Figure 11. Exemple de modèle VUML « Concessionnaire de voitures » [Nassar, 2005].

Si, des extensions telles que proposées précédemment existent, il n'en demeure pas moins vrai qu'UML n'intègre pas explicitement les relations entre les deux niveaux d'abstractions différents. Le problème majeur est qu'une modification dans un modèle (M1) induit, dans une démarche classique, la nécessité de le dériver à nouveau, cette procédure induisant une inadéquation des développements réalisés. Aussi, dans un souci de pérennité des applications, d'autres voies doivent être envisagées permettant d'encapsuler les niveaux (M2) et (M1) dans les applications comme proposé dans le formalisme de modélisation sNets [Lemesle, 2000] sur lequel nous nous appuyons dans le chapitre 3. De plus, il est à noter également le souci de pouvoir enrichir le méta-modèle (M2) en regard de nouvelles problématiques jusqu'alors non envisagées. Cela suppose que la définition du méta-modèle puisse être récursive (auto-définition des concepts).

Dans la littérature plusieurs travaux visent à modéliser et implémenter des systèmes PDM avec des approches orientées objet en utilisant UML [Eynard et *al.*, 2004]. Des ateliers de génie logiciel (ex. MEGA, Rational Rose, etc.) peuvent être utilisés. Ils permettent d'une part de dériver automatiquement le modèle conceptuel objet (1) en modèle relationnel (2) et d'autre part de générer le squelette du modèle de traitement à implémenter (3), ce dernier étant complété par les spécifications de la logique de traitement de ces objets (4) (ex. algorithme, etc.) (cf. Figure 12).

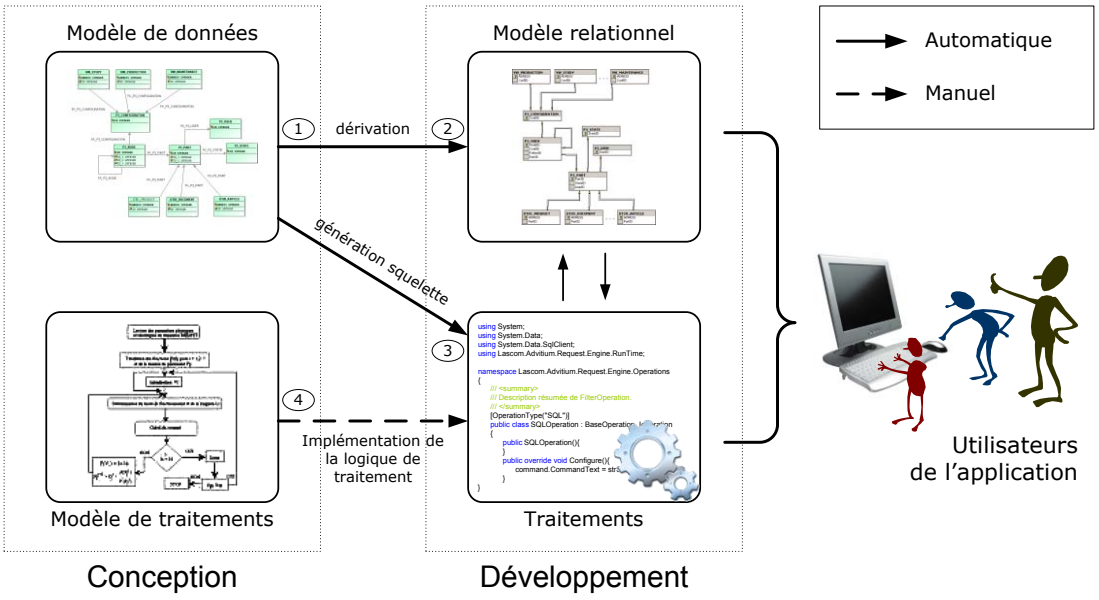

Figure 12. Démarche classique de déploiement des applications

Cependant, le contexte des systèmes d'informations produit et plus particulièrement des applications PLM implique des besoins différents que ceux présents dans les applications de gestion traditionnelles. « *Les données gérées sont plus complexes (graphiques, texte, etc.), les modèles de données sont plus riches (structures de données hiérarchisées), à forte sémantique et impliquent une dynamique riche à plusieurs niveaux d'abstraction (instance, classe, modèle)* » [Gzara, 2000] mais nécessite aussi de disposer de systèmes évolutifs et flexibles pour anticiper et réagir aux évolutions des besoins des utilisateurs. Dans ce contexte, les approches orientées objet présentent plusieurs avantages pour la conception et le déploiement des applications PLM notamment en terme de réutilisation de code à partir de spécifications identiques. Cependant, dans la pratique, l'évolution des modèles

conduit souvent à repenser les traitements et les codes déjà développés pour implémenter les variations des modèles conceptuels.

Par ailleurs, les éditeurs et les intégrateurs des applications PLM, sont confrontés au caractère spécifique inhérent aux projets clients, cet aspect spécifique des besoins constitue un goulet d'étranglement dans le processus de déploiement des solutions PLM, et conduit à des investissements lourds notamment en charge de développement. En effet, ces derniers engendrent des délais de mise en œuvre et des ressources importants. Mais, malgré le caractère spécifique des projets clients, la plupart des besoins fonctionnels sont souvent génériques.

Il est ainsi nécessaire de définir un cadre de modélisation et de nouveaux concepts capables de couvrir la diversité des applications PLM et autoriser leurs évolutions.

2.2 Approches par graphes de données

En regard de la problématique issue de la modélisation des données techniques sous forme de configuration ou de graphe, il demeure que les données en soit ne constituent pas l'information essentielle dans une application PLM, mais que ce sont les interactions entre les données et donc les liens qui leurs donnent tout leur sens. Ainsi, la complexité n'est pas issue uniquement des informations techniques du produit ni de son cycle de vie, mais également de l'impact d'un évènement sur l'ensemble des données qui définissent le produit.

De manière à mettre en évidence graphiquement ces structures complexes de liens entre les objets dans des applications PLM existantes, nous avons utilisé la bibliothèque GraphViz[3] [Gansner et North, 2000] pour le fait qu'elle permet la représentation de graphes très denses comprenant un très grand nombre de nœuds grâce à des algorithmes très évolués. Elle est très rapide à l'exécution et le rendu (généré par routage) est optimisé afin que les liens ne recouvrent pas les nœuds et qu'ils ne se croisent pas.

La Figure 13 présente un aperçu de liens de configurations dans une application mettant en évidence des relations d'un objet à un objet élémentaire mais également d'un objet à un objet complexe car pouvant être lui-même en relation avec un autre

[3] Conçue par une équipe des laboratoires de recherche de AT&T (http://www.graphviz.org/)

objet. La nature de ces relations est définie au sein de la configuration est peut être de type « décomposition », « compatibilité », « documentation », etc.

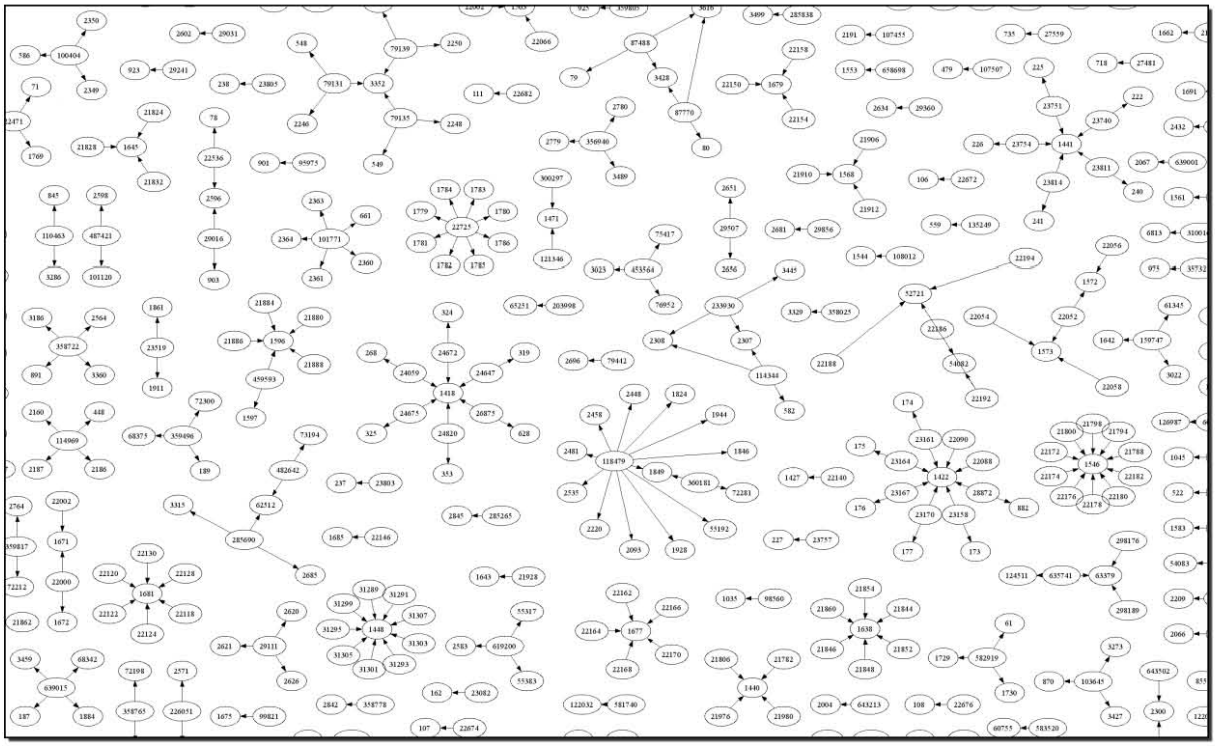

Figure 13. Aperçu d'un graphe représentant les liens entre les instances d'objets dans une application PLM.

Notre problématique autour du PLM est double, en effet il s'agit de modéliser les applications PLM pour d'une part, formaliser les concepts nécessaires pour modéliser des configurations et des points de vue traités précédemment et, d'autre part, permettre l'organisation, la gestion et la navigation dans les graphes de données. Aussi, nous présentons dans ce qui suit les différentes approches significatives en termes de représentation d'information permettant de structurer les données. Nous retenons le formalisme sNets en réponse à nos préoccupations par le fait qu'il permet d'encapsuler dans les modèles des applications PLM sa vue génétique ou méta-modèle.

2.2.1 Approche issue du Web sémantique

Dans la littérature, d'autres travaux s'intéressent à la modélisation de produit, avec des méthodes et outils du Web Sémantique [Boccon-Gibod, 2006] constituant une approche moins traditionnelle que la précédente. Ainsi, des approches basées sur les Topic Maps [Bergström et *al.*, 2004] ou l'utilisation de RDF (*Resource Description*

Framework) [Jun et *al.*, 2005] visent également à modéliser et gérer les informations produit tout au long de son cycle de vie.

En effet, bien que le Topic Maps [ISO, 2003b] représente un standard de notation qui permet de définir des vues multiples et concourantes d'un ensemble d'information (essentiellement des documents), sa portée a dépassé le cadre strictement documentaire se présentant comme une alternative aux langages de méta-données. Ce standard disposent aussi d'une représentation XML (*eXtensible Markup Language*) appelée spécification XTM (*Xml Topic Maps*) [Pepper et Moor, 2001].

A l'instar des Topic Maps, le format RDF [Klyne et Carroll, 2004], développé par le W3C (*World Wide Web Consortium*), est un cadre de méta-données ; il propose donc un modèle abstrait permettant d'attribuer certaines propriétés à certaines ressources. Il propose également une syntaxe basée sur XML permettant de représenter les objets de ce modèle. RDF possède un modèle de base très simple, ce modèle est basé sur la notion de graphe qui permet de représenter des relations étiquetées entre ressources d'information (les nœuds), des propriétés étant attribuées aux nœuds du modèle. Les étiquettes des relations représentent les types de propriété et les points d'arrivée des arcs sont les valeurs des propriétés.

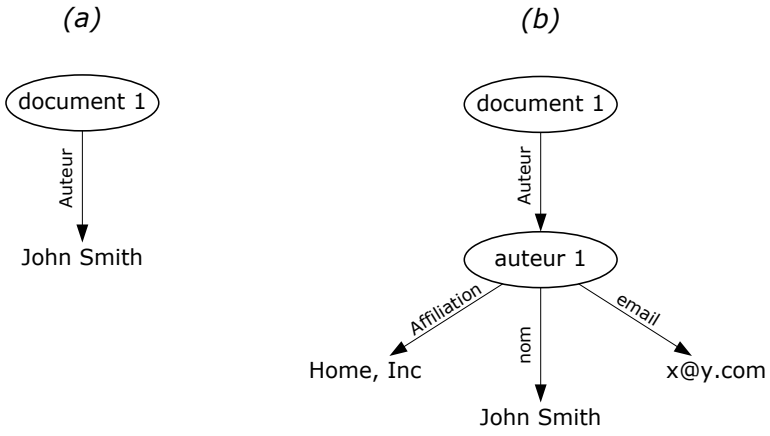

Figure 14. Spécification d'une propriété en RDF.

Les propriétés sont représentées par des arcs, dont l'étiquette correspond au type de la propriété que l'on désire attribuer, le point de départ est la ressource à laquelle on désire attribuer la propriété, et le point d'arrivée est la valeur que l'on veut donner à la propriété. La Figure 14.a montre une propriété énonçant que « John Smith » est l'auteur de la ressource « document 1 ».

Dans cet exemple, la valeur de la propriété est atomique (la chaîne « John Smith ») ; il est cependant possible que la valeur de la propriété soit structurée. Pour ce faire, il suffit d'utiliser comme point d'arrivée de l'arc une autre ressource (un autre nœud du graphe), laquelle possédera l'ensemble des propriétés désirées. La Figure 14.b illustre cette possibilité.

La norme RDF prévoit également l'attribution de propriétés à des propriétés. L'attribution d'une propriété à une propriété permettrait d'exploiter la provenance d'une méta-donnée comme critère de recherche d'une ressource. Le format RDF constitue la base de plusieurs initiatives de méta-modélisation.

Ainsi, l'utilisation d'une représentation sous forme d'un schéma RDF permet, à l'aide des techniques de recherche basées sur le Web sémantique, de proposer une navigation aisée à l'aide de mots clés spécifiés par les nœuds d'un graphe complexe. Celle-ci peut être envisagée pour améliorer les performances de la navigation dans les structures de configuration produit par l'utilisateur. Cette navigation permet par exemple de rechercher des informations en regard de l'expression de critères sur les liens ou sur les objets, ou d'effectuer des mesures d'impact, etc.

Cette approche reste néanmoins générique par le fait qu'elle ne propose pas de mécanisme de typage de nœuds permettant ainsi de proposer un plus fort pouvoir d'expression au niveau conceptuel. Pour palier à ce manque, la section suivante présente les apports du formalisme sNets basé sur les réseaux sémantiques.

2.2.2 Approche issue des réseaux sémantiques

Le formalisme de réseaux sémantiques introduit en 1968 par Ross Quillian [Quillian, 1968] avait pour but la reconstitution d'un modèle de la mémoire humaine par un système de représentation dans lequel les concepts acquerraient leur signification par les liens qui les unissaient à d'autres concepts.

Les réseaux sémantiques visent à représenter graphiquement des concepts et des relations entre ces concepts. La connaissance est décrite et représentée par des nœuds dont les interrelations sont établies par des liens étiquetés. Les nœuds sont les différents types d'information en mémoire. L'étiquette associée au lien indique quel est le type de relation entre deux nœuds. L'avantage de ce formalisme est qu'il permet la représentation de tout type d'information tandis que son inconvénient

majeur est l'absence de modularité et de structuration de la sémantique de ces informations.

Une étude des différents formalismes de modélisation est présentée par Richard Lemesle [Lemesle, 2000]. Il a de plus proposé le formalisme sNets basé sur les concepts de base des réseaux sémantiques auxquels a été ajoutée la notion de typage. sNet présente un formalisme minimal au sein duquel tout peut se ramener à un ensemble de noeuds et de liens. Un exemple de représentation graphique du formalisme sNet est proposé à la Figure 15 mettant en évidence le typage des nœuds et des liens par les notions de sous-typage en classe/sous-classe (Cercle et Carré sont des particularisations de Figure), l'association de méthode (*AfficherSur*) avec paramètre (*Ecran*) et la notion d'attribut (*Position*) relié à un objet permettent une représentation atomique de celui-ci.

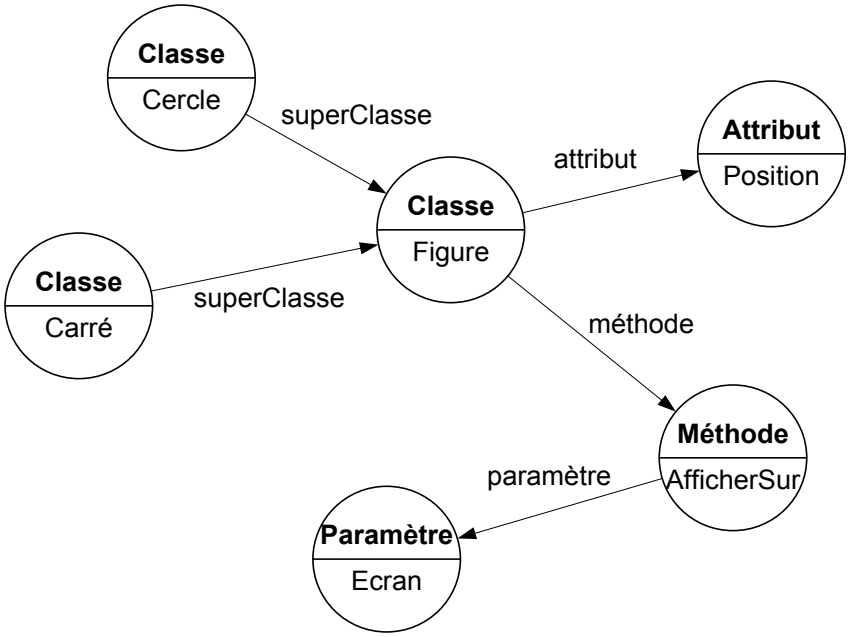

Figure 15. Extrait d'une représentation graphique typée d'un sNet [Lemesle, 2000]

Dans le formalisme sNets, chaque concept est défini au niveau méta-méta modèle correspondant à l'univers du formalisme. Le niveau méta-modèle, s'appuie sur ce formalisme en respectant des règles de conformité ou règles sémantiques pour sa définition. Le niveau modèle englobe la définition du méta-modèle par l'ajout de « Classe » sur Figure, ou « Attribut » sur Position, etc. Ces deux niveaux M1 et M2 (au sens MOF) sont empaquetés dans la définition du modèle comme présenté à la Figure 16. Ceci constitue un changement quant à l'organisation de l'architecture des

modèles où classiquement chaque niveau est géré séparément avec des relations entre chacun des niveaux n et n-1 alors que dans l'architecture proposée pour le formalisme sNets il y a agrégation des niveaux supérieurs vers le niveau inférieur.

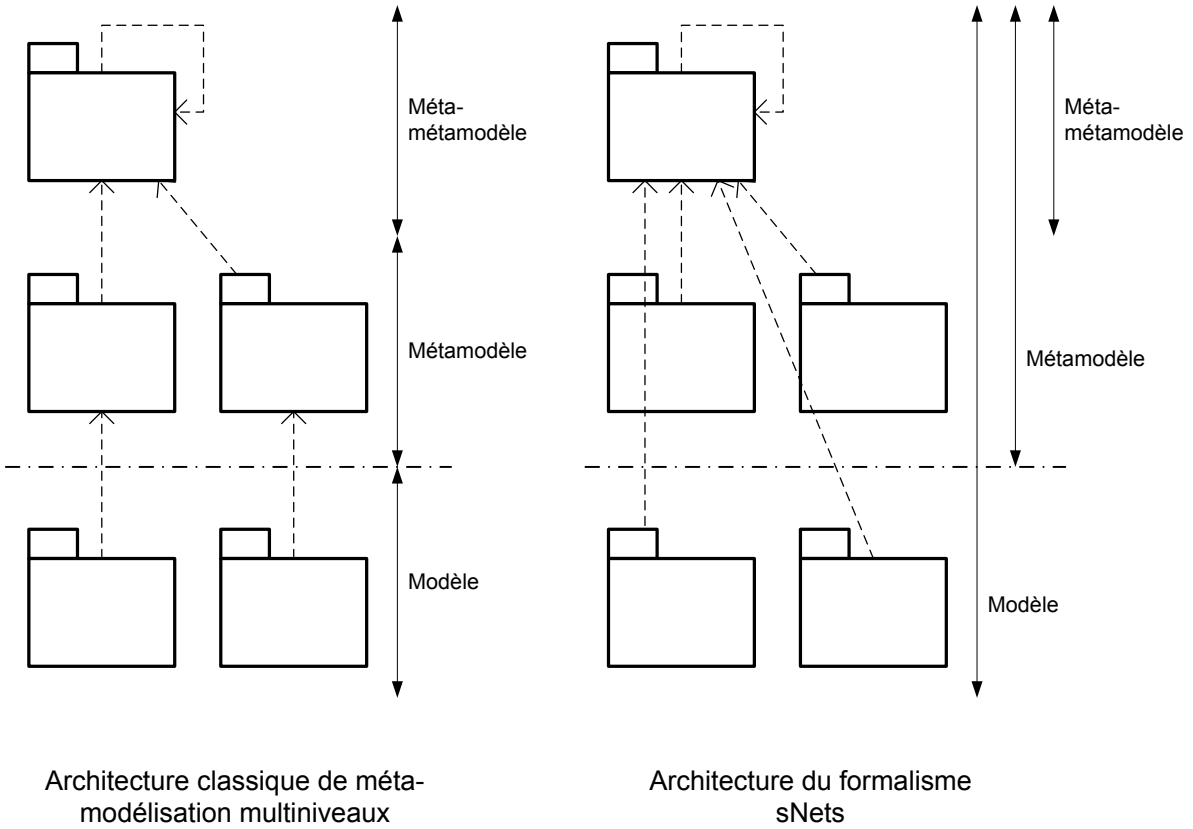

Figure 16. Comparaison entre le formalisme sNets et l'architecture classique multi-niveaux [Lemesle, 2000].

De plus, dans le formalisme sNets, chaque niveau de modèle est en relation avec son niveau supérieur au travers d'une relation d'abstraction (méta) et au travers d'une relation sémantique (*sem*). L'architecture MOF propose un langage de définition de méta-modèles et un langage de définition de modèles, néanmoins ces éléments sont distincts de sorte qu'il n'est pas possible de lier explicitement les modèles à leurs méta-modèles. La relation méta existant entre les niveaux de modèles est présente et explicitée dans les méthodologies de modélisation mais n'est pas formalisée. Aussi, la relation *sem*, proposée par le formalisme sNets, permet de relier sémantiquement deux niveaux d'abstraction définissant ainsi la vue génétique du niveau inférieur.

Aussi, nous retenons l'approche proposée par [Lemesle, 2000] basée sur le formalisme sNets visant à encapsuler la vue génétique dans le modèle correspondant à l'encapsulation de la vue méta-modèle dans le modèle. Ceci permet d'avoir des modèles pouvant évoluer par le fait qu'ils contiennent les concepts nécessaires à leur propre définition.

3 Conclusion

La diversité des applications PLM (diversité liée à la spécificité des clients), la complexité croissante des données et le besoin de systèmes évolutifs et flexibles (complexité due à l'évolution des besoins au fil du temps) fait qu'il n'existe pas un modèle d'application universel, capable de couvrir l'ensemble des besoins des différents clients.

Aussi, nous avons montré au travers de notre étude bibliographique qu'il existe plusieurs modèles produit. De plus, selon Muriel Lombard [Lombard, 2006], la modélisation de produit est loin d'avoir la couverture escomptée du cycle de vie produit. « *Il n'existe pas de modèle de produit universel mais une pléthore de travaux répondant à des aspects très précis d'une problématique*».

De plus, la problématique à laquelle les éditeurs et intégrateurs de solutions PLM sont confrontés provient du caractère spécifique des besoins utilisateurs nécessitant à chaque fois des développements particuliers. La demande d'évolution des applications proposées nécessitant une implication forte des éditeurs, il convient d'envisager des progiciels pouvant supporter l'évolution des modèles sans impacter les développements déjà réalisés voire même de développer des applications indépendantes du modèle final implémenté dans le progiciel.

Dans ce contexte, nous retenons le formalisme sNets qui intègre un système de typage, et une gestion de la modularité le rendant réflexif (auto-défini). La récursivité permet ainsi de limiter le nombre de niveaux d'abstraction. Le modèle réflexif se valide lui-même, les outils et algorithmes applicables au niveau du modèle sont applicables aussi au niveau des instances de ce modèle. Ceci facilite l'adaptation aux extensions ou les évolutions et modifications futures.

Ce dernier point est fondamental car il peut ainsi permettre à un éditeur d'implémenter rapidement des applications PLM spécifiques aux besoins des clients

sans avoir à redévelopper à chaque fois les fonctions et mécanismes nécessaires à la gestion du produit tout au long de son cycle de vie.

Nous présentons dans le chapitre 2 les concepts du progiciel Advitium™ au travers de leurs méta-modélisations. Cette étude nous permet de mettre en évidence certains manques ou incomplétudes pour lesquels nous apportons des réponses et modifications au chapitre 3.

CHAPITRE 2

Extraction et formalisation des concepts

CHAPITRE 3

Formalisation et déploiement des nouveaux concepts

CHAPITRE 4

Conclusion et perspectives

1 Conclusion

L'étude effectuée chez LASCOM sur leur progiciel Advitium™ a consisté, à partir d'un ensemble d'applications PLM existantes, à mettre en place une démarche de « Reverse Engineering » qui a permis de formaliser les concepts nécessaires à la gestion de configuration au sein d'un méta-modèle UML. Cette étape fondamentale a permis de valider au niveau conceptuel les notions manipulées, puis de mettre en avant les avantages de travailler à ce niveau de modélisation en vue d'une capitalisation au plus tôt des études réalisées. Nous nous sommes concentrés ensuite sur les spécificités liées à la gestion de configurations de produits (structure hiérarchique conditionnelle). Nous avons développé des outils pour visualiser les liens entre les instances des objets techniques au sein des configurations. Nous

considérons que toute instance de configuration peut être représentée à l'aide d'une structure de graphe (DAG ou Directed Acyclic Graph). Les traitements liés aux configurations peuvent donc s'inspirer des techniques et méthodes issues des théories des graphes.

Il n'est plus à démontrer, en regard des approches de modélisation de base de données, qu'une bonne organisation de données (modèles cohérents et adaptés aux besoins) permet de simplifier considérablement les traitements et ainsi d'améliorer les performances des applications concernées.

Aussi, notre travail est réalisé dans le but de formaliser les objets techniques et les liens existants entre eux pour répondre à :

- un souci de présentation des objets techniques à l'utilisateur et une facilité d'accès lors de recherche dans les configurations,
- une optimisation des liens existants dans la base de données en regard de l'étude sémantique menée au niveau conceptuel à partir du cahier des charges du client,
- une standardisation des types de configuration dans un souci de capitalisation des savoir-faire en déploiement de solutions PLM.

2 Perspectives

Dans l'objectif d'aider les développeurs d'applications PLM, ce travail s'est situé dans un premier temps au niveau des étapes amont de définition d'un cadre formel de modélisation d'Advitium™. Il s'agit désormais d'envisager de s'attacher aux étapes avales de définition de modèles de référence centrés métiers ainsi que de méthodologie de déploiement ad hoc de manière à faciliter le développement d'application PLM non plus par une difficile réutilisation de toute ou partie d'applications existantes mais par une réelle particularisation de solution générique d'entreprise.

Ce souci de capitalisation des savoirs et savoir-faire déjà utilisés dans des projets industriels antérieurs est une préoccupation de la société LASCOM pour optimiser

les temps de développement projet de manière à leur permettre de se focaliser sur l'évolution du produit, sur de nouveaux projets et applications à forte valeur ajoutée.

Aussi, à partir des conclusions énoncées précédemment (partie gauche de la Figure 17), il s'agit d'envisager un nouveau travail de recherche autour de la formalisation de modèles de référence centrés métiers jusqu'à leurs implémentations au niveau physique (partie droite de la Figure 17).

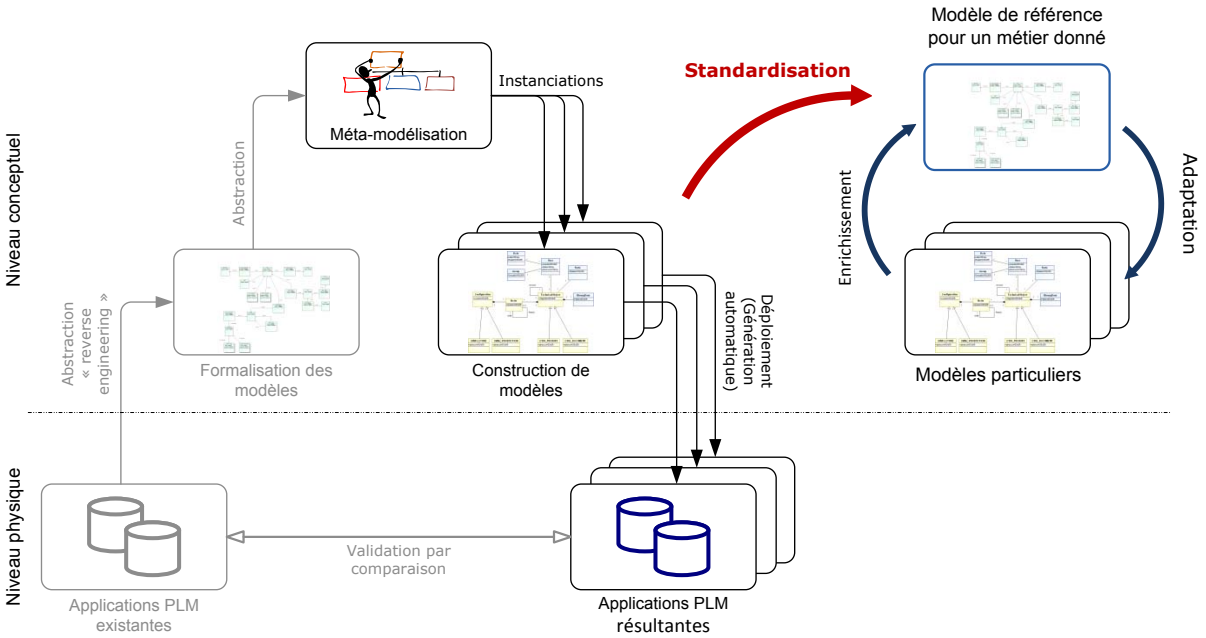

Figure 17. Processus d'instanciation de modèle

Il s'agit d'analyser les problèmes posés par la réutilisation des modèles, de les formaliser pour mieux les comprendre et, dans un souci de réutilisation, de pouvoir les modéliser avec un objectif de généricité. Il s'agit aussi de définir et de mettre en œuvre des méthodes de déploiement de ces modèles génériques en regard d'un contexte particulier, la définition même de ce contexte concourant à garantir avec succès sa particularisation future mais aussi sa bonne caractérisation à des fins de réutilisation.

Les méthodologies de déploiement devraient être à aborder dans deux contextes de développement différents :

- pour des applications de type re-conception où il s'agit de mettre en place une solution « client » à partir d'une application « dérivée » d'une existante. Celle-

ci obtenue à partir d'un cycle de reverse engineering - capitalisation – modèle générique – déploiement de modèle.

- pour des applications nouvelles (nouveau métier, nouveau domaine, exemple de verticaux métiers comme agro-industrie, microélectronique, conception/production mécanique, etc.).

Ces méthodologies pourraient être utilisées chez LASCOM ou tout autre éditeur dans tout le cycle de vie des solutions PLM, en partant de la phase commerciale, l'avant-vente, le développement de l'application, les tests, l'installation, la formation et le suivi de l'application avec ses modifications ultérieures (gestion d'affaires, maintenance, évolution, etc.). Une phase de capitalisation des informations générées par ce déploiement devra être envisagée afin d'enrichir les modèles de référence voire même le méta-modèle.

Les objectifs attendus d'un point de vue industriel pourraient porter sur :

- une réduction du temps de conception d'application par réutilisation et adaptation de concepts génériques,
- une meilleure adéquation de l'offre au besoin client (optimisation des traitements de données, connaissances des structures de données – processus – organisation, etc.),
- une maintenance optimisée (généricité des concepts de maintenance appliquée, temps de réponse du service en ligne, etc.),
- une évolution et une adaptation de l'offre corrélées aux besoins des produits et des services clients,
- une robustesse de l'application / différents intervenants dans les cycles de maintenance – évolution (changement de personnes, connaissances des applications).

Les objectifs attendus d'un point de vue recherche pourraient porter sur :

- la définition d'un nouveau métier en charge de la création et de l'enrichissement des modèles particuliers en regard de la similarité entre plusieurs applications. Il s'agit de proposer une capitalisation efficace du savoir-faire des développeurs et paramétreurs d'applications,

- une méthodologie de recherche et de codification efficace, à savoir comment trouver le bon modèle de référence en regard d'un cahier des charges ? ceci implique donc comment affecter des caractéristiques aux modèles de référence, en tout ou partie, de manière à pouvoir le plus simplement possible récupérer les parties existantes intéressantes pour les applications futures,
- enfin, la définition d'un cadre semi-automatisé de particularisation des modèles de référence en utilisant la métaphore du systémographe de J.L. Le Moigne [Le Moigne, 1990] reprise dans les travaux de [Lombard, 2006] proposant de construire le modèle final à partir d'une projection effectuée sur les points de vue à prendre en compte dans la modélisation à rapprocher dans notre cas de la notion de contexte.

Bibliographie

Ackermann, P., 2005. *PLM-integrated Configurators for Machine and Plant Construction*, in 19th International Joint Conference on Artificial Intelligence (IJCAI'05). Edinburgh – Scotland.

Asikainen, T., T. Männistö, et T. Soininen. 2004. *Using a Configurator for Modelling and Configuring Software Product Lines Based on Feature Models*. in Software Variability Management for Product Derivation. Boston, MA, USA.

Bergström, P., P.A. Ling, et L.M. Garshol, 2004. *Topic Maps in the PDT Domain*. White paper from Eurostep and Ontopia (http://www.eurostep.com/files/EurostepOntopiaWhitePaper.pdf)

Bernard, A. 2000. *Modèles et approches pour la conception et la production intégrées*. Productique, méthodes et outils. Hermes Science.

Bigand, M., 2005. *Intégration de modèles pour l'ingénierie des systèmes de conception*, Sciences physiques. Université des sciences et technologies de Lille.

Boccon-Gibod, H., 2006. *Application de méthodes et outils du Web sémantique pour la gouvernance d'un système d'information industriel*. Semaine de la Connaissance (SdC2006), Nantes

Booch, G., J. Rumbaugh, et I. Jacobson, 1998. *The Unified Modeling Language User Guide*. 1st edition. Addison-Wesley Professional.

Bettaieb, S. F. Noël, et S. Tickiewitch. 2003. *Interface between CAD/CAM software and an integrative design environment*. International CIRP Design Seminar: Methods and Tools for Co-operative and Integrated Design.

Celko, J., 2004. *Joe Celko's Trees and Hierarchies in SQL for Smarties*. Morgan Kaufmann

CIMdata, 2002, "Product Lifecycle Management: Empowering the Future of Business". CIMdata, Inc.

Chen, Y., 2003. *On the computation of recursion in relational databases. In Effective Databases For Text & Document Management*, S. A. Becker, Ed. Idea Group Publishing, Hershey, PA, pp. 263-277.

Conallen J., 2000. Concevoir des applications Web avec UML, Editions Eyrolles.

Crégut X., S. Ebersold, M. Nassar et B. Coulette, 2005. *Un patron de génération de code pour le profil VUML*. Dans actes Workshop OCM adossé à la conférence sur les Langages et Modèles à Objets (LMO 2005).

Derigent, W. et M. Lombard, 2006. *Contribution à la conception de système d'information Produit : méthode pour une approche par modélisation multi-points de vue*, Ingénierie de la conception et cycle de vie des produits. Hermes Lavoisier Paris. pp. 343-370.

Eloranta, E., A.P. Hameri, et L. Lahti, 2001. *Improved project management through improved document management*. Computers in Industry, 2001. 45(3). pp. 231-243.

Eynard, B., 2005. *Gestion du cycle de vie des produits et dynamique des connaissances industrielles en conception intégrée*. Université technologique de Compiègne.

Eynard, B., T. Gallet, P. Nowak et L. Roucoules,2004. *UML based specifications of PDM product structure and workflow*. Computers in Industry, Volume 55, Issue 3, pp. 301-316.

Gansner, R., E. et S. C. North, 2000. *An open graph visualization system and its applications to software engineering*. Software: Practice and Experience. Volume 30, Issue 11: pp. 1203-1233.

Gardner T., Griffin C., Koehler J. et Hauser R., 2003. *A review of OMG MOF 2.0 Query / Views / Transformations Submissions and Recommendations towards the final Standard*. OMG 2003 (http://www.omg.org/docs/ad/03-08-02.pdf).

Carnduff T.W. et J.S. Goonetillake, 2004. *Configuration management in evolutionary engineering design using versioning and integrity constraints*. Advances in Engineering Software, Volume 35, Issues 3-4, March 2004, pp. 161-177.

Geryville, M.H., A. Bouas, Y. Ouzrout et N. Sapidis, 2006. *Collaborative Product and Process Model: Multiple Viewpoints Approach*. 12th International Conference on Concurrent Enterprising Center for Concurrent Enterprise, Nottingham University Business School ISBN 978-0-85358-228-1, pp. 391-398. Milan Italie.

Gzara, L., 2000. *Les patterns pour l'ingénierie des systèmes d'information produit*, Génie Industriel, Institut National Polytechnique de Grenoble.

Harani, Y., 1997, *Une approche multi-modèles pour la capitalisation des connaissances dans le domaine de la conception*. Institut National Polytechnique de Grenoble.

Hollmann, O., T. Wagner, et A. Güenter. 2000. *EngCon – A Flexible Domain-Independent Configuration Engine*. European Conference on Artificial Intelligence (ECAI'2000).

ISO, 2003a. *Quality management systems - Guidelines for configuration management*: 10007:2003.

ISO, 2003b *Information technology - SGML applications - Topic maps*. 2nd edition, ISO/IEC 13250:2003.

Lee, C.K.M., G.T.S. Ho, H.C.W. Lau and K.M. Yu, 2006. *A dynamic information schema for supporting product lifecycle management*. Expert Systems with Applications Volume 31, Issue 1: pp. 30-40.

Jun, H.B., D. Kiritsis, et P. Xirouchakis, 2005. *Product lifecycle information modeling with RDF*. International conference on Product Lifecycle Management. (PLM'05). Lyon.

Klyne, G. et J.J. Carroll, 2004. *Resource Description Framework (RDF): Concepts and Abstract Syntax*, World Wide Web Consortium, Recommendation REC-rdf-concepts-20040210.

Lemesle, R., 2000. *Techniques de Modélisation et de Méta-modélisation*, Sciences de l'Ingénieur, Automatique et Informatique Appliquée. Université de Nantes.

Le Moigne J.L., 1990. *La modélisation des systèmes complexes*. Dunod, Paris

Lombard, M., 2006, « Contribution de la modélisation informationnelle aux Processus de Conception et Réalisation de Produits Manufacturiers : vers une Ontologie Métier ». Habilitation à Diriger des Recherches. Université Henri Poincaré - Nancy I (09/10/2006).

Lombard, M., 1994, « Contribution au génie productique : prototypage d'une architecture d'ingénierie concourante des systèmes intégrés de fabrication manufacturière ». Thèse de doctorat de l'université de Nancy I, France.

Männistö, T. et R. Sulonen. 1999. *Evolution of Schema and Individuals of Configurable Products*. in ECDM'99 of - Workshop on Evolution and Change in Data Management. Versailles, France.

Maurino, M., 1993. *La gestion des données techniques. Organisation industrielle*. Edition Masson, Paris.

Million, O., 1998. *De l'intégration des métiers par les données techniques vers la maîtrise de la modélisation conceptuelle : la méthode V.I.M. (Viewpoints Information Modelling)*. Université Henri Poincaré Nancy.

Müller, E., P. Dadam et M. Feltes, 2003. *Efficient Assembly of Product Structures in Worldwide Distributed Client/Server Environments*, Proc. BTW 2003, pp. 364-383, 2003.

Nassar, M., 2005. *Analyse/conception par points de vues : le profil VUML*. Institut National Polytechnique de Toulouse.

Nöel, F., 2005. *A Dynamic multi-view product model to share the product behaviours among designers*. International Conference on Product Lifecycle Management. pp. 113-120.

Quillian R., 1968. *Semantic memory*. Semantic Information Processing, Minsky editions, MIT press, Cambridge Mass.

Peltonen H., 2000. *Concepts and an Implementation for Product Data Management*. Department of Computer Science and Engineering, vol. Ph.D. thesis. Scandinavica: Helsinki University of Technology.

Pierra G., D. Hondjack, N. Negue, M. Bachir, 2005. *Transposition relationnelle d'un modèle objet par prise en compte des contraintes d'intégrité de niveau instance*. Inforsid'05, Grenoble, p.455-470.

Pepper, S. et G. Moore, 2001. *XML Topic Maps (XTM) 1.0*., TopicMaps.Org Specification. (http://www.topicmaps.org/xtm/1.0/xtm1-20010806.html).

Stark, J., 2005, "Product Lifecycle Management: 21st century paradigm for Product Realisation". 2ed. Decision Engineering. Springer.

Van der Aalst, W. and K. Van Hee, 2004, "Workflow Management: Models, Methods, and Systems". The MIT Press.

Zina, S., M. Lombard, L. Lossent, C. Henriot, 2006a. *Generic Modeling and Configuration Management in Product Lifecycle Management*. International Journal of Computers, Communications & Control (IJCCC). Vol. I, No. 4, pp. 126-138, 2006.

Zina, S., M. Lombard, L. Lossent, C. Henriot, 2006b. *Generic modeling and configuration management in Product Lifecycle Management*. Computational Engineering in Systems Applications (CESA'2006). Beijing, China, 4-6 October 2006.

Zina, S., M. Lombard, L. Lossent, 2006c. *Multi-view model for contextual configuration management in PLM applications*. International Conference on Product Lifecycle Management (PLM'06). Bangalore, Karnataka, India, 10-12 July 2006.

Zina, S., M. Lombard, L. Lossent. 2006d *Integration of contextual views in configuration management for PLM applications*. 9th IFAC Symposium on Automated Systems Based on Human Skill and Knowledge (ASBoHS'06). Nancy, 22-24 May 2006.

Annexes