

Méthodes d'évolution de modèle produit dans les systèmes du type PLM

Seyed Hamedreza IZADPANAHI

Table des matières

1. Introduction	2
2. Approche « Ingénierie Dirigée par les Modèles » (I.D.M.)	5
3. Les processus d'évolution	7
3.1. Les Scénarios	7
3.2. Similarité	9
3.3. D.S.M.L. : Domain Specific Modeling Language	10
3.4. Comparaison des modèles	13
4. Exemple d'évolution : du produit spécifique au produit générique	15
4.1. Généricité de produit	15
4.2. Approche d'évolution proposée	17
5. Conclusion	25
6. Bibliographie	26

Introduction

La maîtrise des données techniques des produits industriels est un enjeu crucial pour la pérennité de l'entreprise. Des produits toujours plus sophistiqués et diversifiés, des collaborations plus étendues, des phases de conception et de fabrication plus délocalisées obligent les entreprises à s'investir davantage dans la mise en place de systèmes d'information qui garantissent un partage plus rapide et mieux sécurisé des données et des connaissances [1].

Les systèmes d'information de type P.L.M. (Product Lifecycle Management) ont vocation à collecter, organiser, gérer, sécuriser et partager les informations liées aux produits, pendant leur cycle de vie [2] [3]. Ces informations sont générées tout au long du cycle de vie des produits, à partir des phases amont de conception jusqu'aux phases aval de fin de vie et de recyclage, en passant par les phases de fabrication, de vente, de maintenance et de service aux clients. La maintenance et la compatibilité des informations au long de ces phases posent de nombreux problèmes de pérennité et de cohérence [4].

Le P.L.M. est donc devenu un composant stratégique de l'entreprise, qui peut apporter un support aux activités internes, ainsi qu'aux collaborations externes d'une manière centralisée et cohérente [5] [6]. Pour bien répondre à ces besoins, les outils logiciels du P.L.M. doivent pouvoir s'adapter aux particularités de l'entreprise (structures de produits, traçabilité, structures des processus, modes de gestion documentaire etc.).

Du point de vue de l'architecture logicielle, les outils P.L.M. sont décomposables en deux parties distinctes : une partie standard correspondant au noyau du système et une partie customisée correspondant à la couche spécifique à l'organisation qui l'exploite. Le noyau offre les fonctionnalités de base : une structure de base des produits (BOM), une structure documentaire, une gestion des états et des droits d'accès, un moteur de workflow, un méta-modèle des données à gérer, un système de gestion de fichiers, un système de gestion de base de données. C'est en s'appuyant sur ce noyau que chaque entreprise construit la couche spécifique, constituée des modèles représentant ses propres produits, ses processus ou son organisation : ils sont construits conformément au méta-modèle du noyau, ainsi que des interfaces homme-machine [7][8]. La figure 1 illustre le noyau et la partie paramétrable : le noyau comporte une base de données qui archive tous les objets, ainsi qu'une infrastructure de modélisation et un moteur de workflows qui permet de modéliser et d'exécuter les processus dynamiques de l'entreprise.

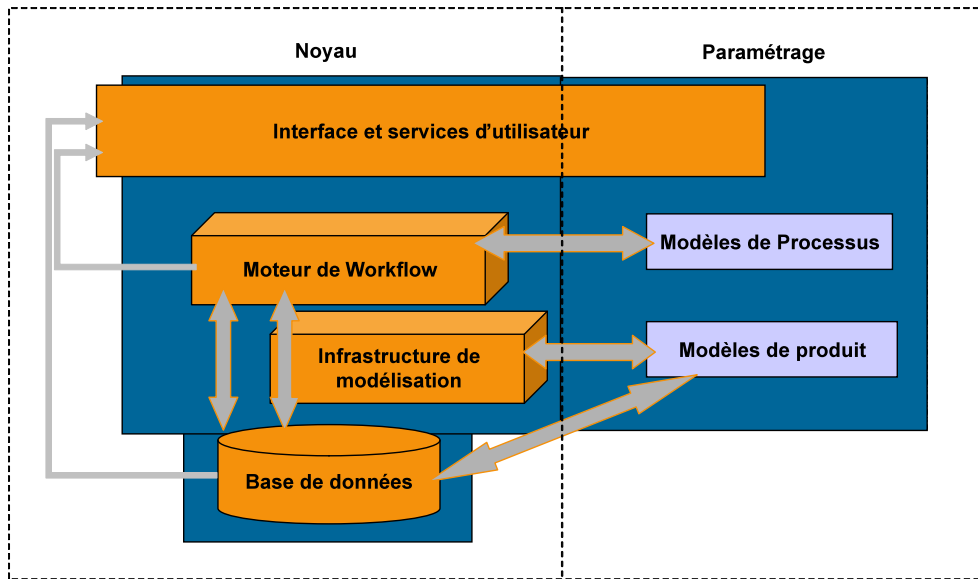


Figure 1 - Architecture logicielle d'un système PLM

De part la particularité de chaque métier et l'inéluctable évolution des besoins au fil du temps, l'organisation et la structuration des informations relatives aux produits ne peuvent être identiques d'une entreprise à l'autre et d'une période à l'autre. La question des migrations et des évolutions du système P.L.M. se pose alors.

Pour aborder les problèmes d'évolution des systèmes P.L.M., cette thèse propose une démarche d'analyse en deux temps :

- analyse des causes métiers des évolutions, pour identifier et préciser le besoin industriel à l'origine des évolutions envisagées. Ce besoin peut concerner les produits - processus - organisation à gérer dans le système P.L.M., mais également les interfaces homme-machine à proposer aux utilisateurs ou les traitements divers à mettre en place sur les données techniques.
- analyse des conséquences techniques de l'évolution, pour identifier la partie de l'outil P.L.M. qui sera impactée par l'évolution et définir la solution technique à mettre en place. Ces conséquences sont de granularité variable : cela va de la modification de quelques éléments de l'outil P.L.M. (modification d'un modèle, modification d'un méta-modèle, modification des interfaces, etc.) jusqu'au remplacement ou au changement total de l'outil lorsque celui-ci s'avère inadapté aux évolutions souhaitées.

Dans ce travail de thèse, nous nous focalisons sur les évolutions relatives aux produits. Ce type d'évolution impacte les parties du système P.L.M. qui permettent de construire les modèles de produits. Ces modèles jouent un rôle central dans les systèmes P.L.M.. Toutes les données techniques relatives aux produits sont organisées selon ces modèles [9]. Un modèle produit permet de représenter la structure du produit, i.e. sa constitution en termes de composants, matières, sous-ensembles, assemblages..., en d'autres termes : la nomenclature. Il décrit également les diverses informations associées au produit et à ses constituants [10], telles que les documents, les outils (de fabrication, de contrôle etc.) et les projets associés aux divers

constituants. Un modèle produit peut représenter le produit à divers niveaux d'abstraction (le produit physique fabriqué, le produit type selon lequel divers produits physiques sont fabriqués, le produit générique représentant une gamme ou une famille de produits types). Il peut également représenter le produit selon plusieurs points de vue, associés aux divers métiers de l'entreprise (bureau d'étude : as designed, bureau de méthode : as built, SAV : as maintained...). La cohérence entre ces points de vue doit être maintenue dans un système P.L.M. [11] [12]. Dans la littérature, plusieurs termes sont utilisés pour désigner le modèle produit tels que structure produit, architecture produit et configuration produit. Ces termes ont une connotation métier et désignent plutôt le contenu et la composition d'un produit. D'un point de vue système d'information, le terme « modèle produit » est plus approprié pour parler de la structure de données support à la représentation du produit dans le système P.L.M..

Selon l'infrastructure et l'ouverture du système P.L.M. considéré, les modifications nécessaires à l'évolution des produits peuvent être effectuées au niveau de la couche spécifique, mais parfois des évolutions du noyau (dans le méta-modèle) s'avèrent nécessaires. Dans la plupart des cas, ces modifications ne sont possibles que dans la couche spécifique (en modifiant les modèles, voire en introduisant un niveau de modélisation intermédiaire entre le méta-modèle du P.L.M. et les modèles produit manipulés).

La première question de recherche posée par cette thèse se formule comme suit :

« Suite à un changement métier, quelle est la procédure à suivre pour faire évoluer les modèles produit et les applications informatiques associées ? »

Par changement métier, il faut entendre l'évolution de l'architecture d'un produit (modification de la constitution d'un type ou d'une gamme de produit), l'évolution de l'offre (introduction de nouveaux produits), la mise en place d'un nouvel outil P.L.M. (avec une infrastructure différente de l'outil existant), la fusion de systèmes d'information (lors du rachat d'une entreprise, le regroupement de sites industriels, etc.), l'adoption d'une nouvelle logique de représentation et de décomposition des produits, etc. Le tableau 1 ci-dessous résume les types de changement métier à l'origine d'une évolution du système P.L.M..

Par procédure, nous désignons une démarche de réingénierie de système d'information qui permet d'obtenir un système P.L.M. répondant aux besoins induits par le changement. Cette démarche est destinée aux intégrateurs de systèmes P.L.M., qu'ils soient internes ou externes (prestataires) à l'entreprise. Pour aborder l'évolution de modèles produit, nous nous appuyons sur une approche prometteuse : l'ingénierie dirigée par les modèles (I.D.M.). Avant de développer les éléments de la démarche d'évolution proposée (section 3), nous présentons dans la section 2 les principes de l'approche I.D.M.

Cause relatives au produit :

Produit Physique	Extension de l'offre produit
	Changement du type de composant de produit
Informations de Produit enregistré dans le système	Ajout/Suppression des concepts dans le modèle produit
	Changement du niveau de modélisation (ex : du spécifique au générique)/ changement de la structure de la nomenclature

Causes relatives à l'organisation :

Changement Inter-Organisationnel	Adaptation du PLM à ceux des partenaires
	Fusion du PLM dans celui d'un partenaire
Changement Intra- Organisationnel	Adaptation du PLM aux autres outils de l'entreprise

Causes relatives aux processus métier :

Changement des processus suite à la re-décomposition du produit ou à la modification de sa représentation

Causes relatives aux logiciels PLM :

Modification/ Mise à jour du logiciel
Intégration/ fusion d'outils PLM différents
Remplacement du logiciel par un autre

Tableau 1 – Causes métiers des évolutions du système PLM

1. Approche « Ingénierie Dirigée par les Modèles » (I.D.M.)

L'ingénierie Dirigée par les Modèles (I.D.M.) est une approche récente en informatique, initiée dans les années 2000 [13] [14]. Elle a été développée afin de répondre aux besoins du génie logiciel, en réduisant la complexité de la conception et du développement des outils informatiques, et en facilitant la collaboration des différents acteurs pendant les phases de développement. Le concept principal de cette approche est de séparer le monde de la conception de logiciels de celui de l'implémentation. Elle propose ainsi des modèles à divers niveaux : des modèles indépendants des plateformes sur lesquels les modèles s'exécutent (PIM pour Platform Independent Models) qui sont ensuite enrichis avec des modèles prenant en compte les particularités des plateformes (PDM : Platform Dependand Models) et deviennent enfin des modèles exécutables et dépendants de la plateforme (PSM pour Platform Specific Models). Cette approche prometteuse en génie logiciel permet de concevoir, modifier et faire évoluer des modèles abstraits sans entrer dans les spécificités techniques des plateformes d'exécution [15].

Le principe d'I.D.M. pourrait être employé dans d'autres domaines car il peut faciliter la conception des modèles. Cette facilité vient du concept de méta-modèle qui est particulièrement développé dans l'I.D.M. [16]. Ce concept met en exergue une hiérarchie de modélisation qui distingue le langage de modélisation des modèles eux-mêmes ; les modèles décrits dans un langage sont lisibles et utilisables dans un autre langage grâce à la technique de transformation.

La notion de langage a, dans IDM, une définition plus large qu'un simple formalisme pour schématiser les modèles ; le langage définit tous les concepts et les règles avec lesquels l'utilisateur peut construire les modèles. Dans l'approche IDM, le mécanisme clé pour définir ce langage est le méta-modèle. Un méta-modèle est un ensemble de règles, concepts et contraintes, utilisées pour définir les modèles dans un contexte précis. Ce principe de méta-modélisation permet de formaliser et généraliser le processus le plus important en I.D.M., celui de la transformation entre PIM /PSM. La modélisation est alors réalisée en plusieurs niveaux : les méta-modèles qui comprennent le langage utilisé pour décrire les modèles, les modèles et les données. Les modèles sont dits *conformes* aux méta-modèles et les données sont les instances des modèles. PIM et PSM se situant tous les deux au niveau modèle (M1) ; ils sont conformes à leur méta-modèle au niveau M2. La figure 2 illustre ces niveaux de modélisation.

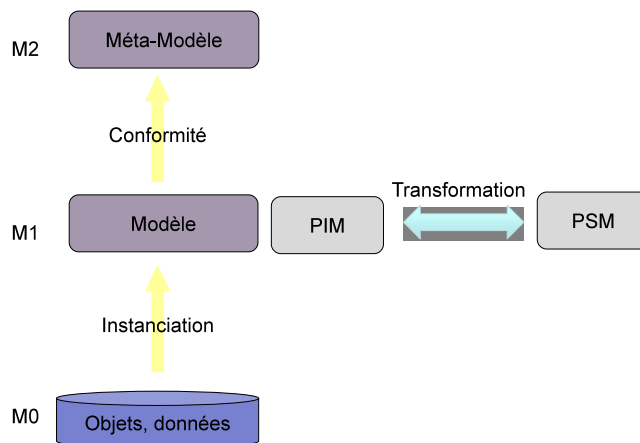


Figure 2 –Les divers niveaux de modélisation en I.D.M.

Le processus de transformation comprend : 1) l'identification des concepts similaires et des correspondances entre les méta-modèles cible et source, 2) l'exécution de la transformation en instanciant les concepts du modèle cible à partir de cette correspondance. Dans la première partie du processus, tous les concepts existants dans les deux méta-modèles sont comparés en prenant en compte les contraintes associées à la transformation, et qui varient selon la problématique d'évolution considérée. En fonction des différents cas possibles de similarité, pour chaque concept, un correspondant est identifié. Dans le cas où il n'existe pas de correspondant, différentes solutions sont proposées. Le détail de ces propositions est présenté dans les sections suivantes.

Cette thèse s'inscrit résolument dans une approche de type I.D.M. pour résoudre la problématique d'évolution des P.L.M. ; ce choix a été dicté par les mécanismes offerts par IDM, dont notamment la transformation. L'évolution peut être considérée comme un type de transformation. Les méthodes de transformation proposées dans l'approche IDM peuvent aider à définir la démarche d'évolution P.L.M. visée.

L'approche IDM présente plusieurs avantages :

- IDM est une approche validée, standardisée et acceptée dans le monde du logiciel.
- les méthodes et principes de l'IDM sont applicables aux systèmes d'information. Ils sont valables pour les systèmes qui utilisent des structures de méta-modèles similaires à ceux

adoptés en IDM. Dans le cas de la modélisation produit, les structures adoptées dans le domaine du PLM se prêtent bien aux principes de l'IDM.

- la transformation des modèles est facilitée grâce au concept de Méta-modèle. Il est possible d'automatiser ou factoriser les démarches de transformation pour tous les modèles conformes à un même méta-modèle.
- l'architecture des systèmes P.L.M. avec deux parties, noyau et couche spécifique, se prête bien à l'esprit de l'approche IDM qui est basée sur deux niveaux de modélisation. Le fait d'avoir deux niveaux de modélisation, un comprenant les modèles et l'autre comprenant les méta-modèles, aide les concepteurs à construire, modifier, réutiliser et implémenter leurs modèles dans différents systèmes, selon différentes conditions et pour répondre à des besoins variés des entreprises.

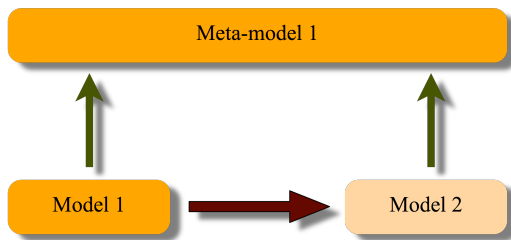
2. Les processus d'évolution

Afin de répondre à la problématique formulée dans la première partie et en utilisant l'approche I.D.M., une méthode générale pour l'évolution des modèle produit est présentée dans cette section. Les scénarios possibles d'évolution au niveau des systèmes PLM sont préalablement présentés. Ensuite, un cadre de similarité qui explique les démarches de comparaison des modèles produit est présenté. Enfin, les mécanismes de transformation de modèles sont expliqués.

2.1. Les Scénarios

Les besoins métiers à l'origine de l'évolution d'un système PLM sont souvent exprimés par les utilisateurs du système en langage naturel avec des termes métiers. L'interlocuteur en charge du projet d'évolution du PLM devra par la suite traduire ces besoins en une spécification plus formelle qui sera compréhensible par les informaticiens en charge de l'implémentation de la solution. Selon le contexte de l'évolution, la solution informatique à mettre en place pour traduire les besoins métiers est différente mais il s'agit dans tous les cas d'une transformation de modèles, comme précisé dans la section 2. Nous distinguons quatre scénarios de transformation selon le contexte d'évolution qui se présente. (Dans les figures de cette partie, les carrés foncés représentent les entités connues et structurées au début du processus de la transformation, tandis que les carrés clairs sont des entités inconnues, lesquelles doivent être identifiées en fonction du processus)

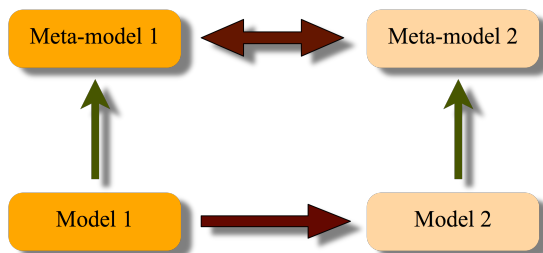
Premier Scénario



Ce cas, le plus simple, peut se produire lorsque l'évolution se fait dans un seul système PLM (sans modification de méta-modèle) ou entre deux systèmes PLM qui possèdent un méta-modèle identique. Par exemple, dans le cas de la modification du produit ou sa représentation dans le PLM, parce que le méta-modèle restera intact, la transformation se loge dans ce scénario. Comme un exemple, nous pouvons constater tous les changements des composants de produit, sans ajouter des nouveaux concepts ou entités au modèle existant.

Traiter ce scénario suit les démarches classiques de transformation dans l'approche IDM.

Deuxième Scénario

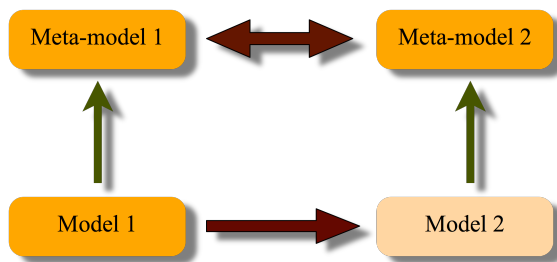


Ce scénario est l'un des cas les plus fréquents de l'évolution de modèle produit, et plus généralement, des systèmes PLM. Dans ce cas, le méta-modèle cible n'est pas connu. Ce scénario correspond au cas de migration vers un nouvel outil PLM ou encore au cas d'une modification majeure apportée à l'offre produit de façon à ce que tout le méta-modèle du système PLM utilisé doive être modifié.

Dès le début du projet d'évolution, le méta-modèle cible doit être construit à partir des exigences et besoins de l'entreprise. Pour permettre cette construction, il est nécessaire de disposer d'un DSML (Domain Specific Modeling Language) qui n'est autre qu'un langage de modélisation dédié à un domaine spécifique, en l'occurrence le PLM. Nous revenons plus tard sur le concept de DSML (§ 3.3).

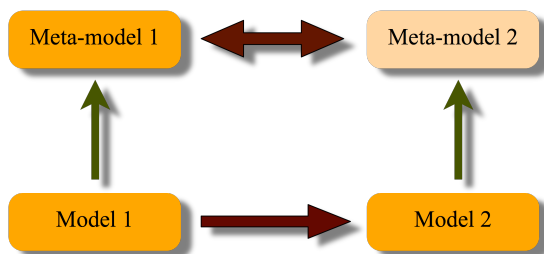
Le processus de la transformation présenté dans cette thèse essaie de focaliser sur ce cas d'évolution.

Troisième Scénario



Ce scénario correspond au changement de système PLM où le méta-modèle cible (du nouveau système) est connu. C'est le cas par exemple d'une entreprise qui veut adapter son système PLM avec ceux de ses partenaires ; dans ce cas, le méta-modèle de produit est figé, et l'évolution se fait vers un système identifié.

Quatrième Scénario



Ce scénario ne s'inscrit pas tellement dans la philosophie de l'approche IDM parce que nous ne pouvons pas avoir des modèles sans leur méta-modèle. Pourtant, ce cas peut se présenter dans le monde industriel. Il s'agit d'un cas dans lequel nous connaissons le système cible que nous souhaitons avoir (le modèle), mais son méta-modèle n'est pas identifié. Le processus de transformation sera un cas particulier qui sera présenté dans la section suivante.

2.2. Similarité

Les scénarios identifiés dans la partie précédente correspondent aux divers cas rencontrés lors d'un problème d'évolution. Quelque soit le scénario, une fois le méta-modèle cible identifié et précisé, le processus de transformation est quasiment identique. Nous allons présenter ce processus général de transformation et détailler ensuite les particularités propres à chaque scénario.

La transformation commence par la comparaison des méta-modèles (cible et source) pour identifier leurs similarités et leurs divergences ; on parle alors de *cadre de similarité* [17].

La comparaison de modèles (et de méta-modèles) est une thématique abordée dans de nombreux travaux de recherches, notamment en génie logiciel [18] [19] [20]. Il faut toutefois noter que les recherches académiques et les outils développés dans ce domaine sont généralement dédiés à la comparaison de modèles. Nous réutiliserons les résultats de ces travaux pour les appliquer à la comparaison de méta-modèles, partant du fait que les méta-modèles sont des modèles avant tout.

La plupart de ces travaux sont consacrés à la comparaison textuelle ou structurelle des modèles afin de déterminer les différences entre les modèles. Toutefois, les modèles considérés dans ces recherches ne sont pas foncièrement très différents ; dans la plupart des cas, il y a beaucoup de concepts communs entre les modèles. Or, les cas d'évolution de PLM rencontrés dans l'industrie peuvent présenter des différences significatives entre les méta-modèles cibles et les méta-modèles sources où les concepts ne sont pas similaires. Pour identifier les concepts correspondants, des démarches plus sophistiquées sont nécessaires. Contrairement à la plupart des travaux traitant cette thématique, les méta-modèles considérés dans le cas de systèmes P.L.M. ne sont pas facilement comparables ; la mise en place d'un processus permettant la comparaison est nécessaire. Ce processus est présenté dans les sections suivantes. De plus, la similarité entre les modèles dépend énormément du contexte d'application, car les sémantiques des modèles imposés par le contexte d'application et les définitions particulières des concepts jouent énormément dans la comparaison des modèles. Il est alors nécessaire de développer un cadre de similarité adapté à chaque contexte.

Pour répondre à cet objectif, la démarche proposée consiste à définir une DSML relative aux modèles produit. Cette DSML sert à la fois à clarifier la sémantique des méta-modèles en question, et aussi à construire les méta-modèles dans les cas du deuxième scénario. Cette DSML doit enrichir le modèle produit avec des sémantiques propres au contexte de modélisation de produits industriels. Nous présentons l'approche DSML dans le paragraphe suivant.

2.3. D.S.M.L. : Domain Specific Modelling Language

Les entités manipulées dans les méta-modèles peuvent être des concepts représentant les composants du produit, les relations entre ces composants ainsi que les contraintes et les règles qui régissent la dynamique des modèles produit. Ces entités appartiennent à deux couches : la première couche est relative au langage de modélisation ; la deuxième couche est relative aux particularités du domaine et qui sont représentées dans une DSML pour enrichir les concepts du domaine d'étude. Dans un contexte de système d'information, les concepts produit sont modélisés à l'aide de classes et d'attributs ; les relations entre concepts sont modélisées à l'aide d'associations, compositions/agrégations et généralisations/spécialisation. Ces entités (classes, attributs, associations, compositions ...) peuvent être accompagnées de contraintes (ou règles) qui régissent, soit la construction du modèle produit (contraintes statiques) telles que les contraintes garantissant la conformité et la faisabilité du modèle, soit la dynamique du modèle telles que les règles définissant le processus d'évolution d'un modèle pendant son cycle de vie.

La comparaison des modèles nécessite une description précise de la sémantique de chaque entité manipulée dans les méta-modèles. Cela est nécessaire pour pouvoir identifier toutes les propriétés des concepts et les comparer précisément. Cette sémantique est de deux types : sémantique langage et sémantique métier.

- La sémantique liée au langage de modélisation, ici le schéma UML, contient la définition générale de l'entité en UML, indépendamment du domaine d'application considéré. Elle comprend la description des concepts, des relations et des contraintes en UML.

L'utilisation du langage UML est privilégiée du fait qu'UML est un langage normalisé et de plus en plus utilisé dans le monde industriel afin de modéliser les systèmes d'information.

- La sémantique métier est associée aux particularités du domaine d'application. Elle contient des informations propres au domaine et définit les concepts devant exister dans un modèle de domaine ainsi que les règles régissant leur construction et leur évolution. Notre objectif est d'identifier la sémantique métier des modèles produit manipulés dans le PLM.

Par ailleurs, il faudra prendre en compte la granularité des entités au cours de la comparaison des modèles. Nous avons considéré les entités, comme des nœuds reliés par différents types d'arcs. Avec cette approche, il n'y a aucune hiérarchie entre les entités. Cet aspect est développé dans la section suivante.

2.3.1. Sémantique des entités

Cette section présente les principaux concepts gérés dans la configuration de produit. Cette liste n'est pas exhaustive, et contient les éléments les plus fréquents dans les modèles produit étudiés dans la littérature ainsi que dans les études de terrain. Les concepts listés dans la suite correspondent soit à des classes, soit à des attributs de classe :

- Nom : une expression textuelle désignant l'appellation du concept (article / produit / document, etc)
- Référence/identifiant : une combinaison unique de caractères et/ou de chiffres, qui identifie précisément et sans ambiguïté un objet
- Dimension : une série de chiffres, représentant les mesures (longueur, largeur, diamètre...) d'un article ou d'une combinaison d'articles, avec une échelle de mesure donnée.
- Masse : un chiffre, représentant la masse d'un article ou d'un ensemble d'articles, avec une échelle de mesure.
- Matériau : liste des matériaux entrant dans la composition des articles.
- Article élémentaire : il s'agit d'un article qui n'est pas lui-même composé d'autres articles.
- Article composé : il s'agit d'un article qui est composé d'au moins deux autres articles (élémentaires ou composés à leur tour).
- Article générique : c'est un type d'article, qui a au moins un paramètre non-valué (plusieurs choix/valeurs sont possibles pour cet article)
- Article spécifique : c'est un type d'article issu d'un article générique en fixant certaines valeurs des paramètres de l'article générique (suite à des choix techniques effectués, par exemple : fixer la couleur). Au niveau de l'article spécifique, certains paramètres peuvent rester non valués ; la différence par rapport à l'article générique c'est que l'article spécifique possède au moins un paramètre valué de plus par rapport à l'article générique associé.

- Article exemplaire : c'est un type d'article issu d'un article spécifique en fixant la valeur de l'ensemble des paramètres de l'article spécifique. Il correspond à une réalisation physique de l'article spécifique.
- Document externe : ce type de document est créé en dehors de l'environnement PLM (comme les fichiers CAO).
- Document interne : ce type de document est créé à l'aide des outils existants dans l'outil PLM.
- Etat : l'état d'un objet (article, combinaison d'articles, document) est déterminé par les valeurs prises par les paramètres caractérisant cet objet. Il correspond à un stade du cycle de vie de l'objet.
- Version : un code (en général) qui permet de distinguer et de garder trace des modifications successives d'un objet.
- Outil : libellé d'un outil utilisé à un moment donné du cycle de vie d'un objet (article, combinaison d'articles, document). Evidemment, la spécificité qui influe les caractéristiques de l'objet associé est présente dans cette information.
- Fournisseur : le fournisseur d'un article, d'un matériau, d'un document, d'un outil...
- Dates d'effectivité : les caractéristiques techniques rattachées au produit peuvent avoir des dates de validité non permanentes. Dans ce cas, les dates d'effectivité préciseront le début et la fin de validité d'une caractéristique (une caractéristique est une qualité intrinsèque à un concept ou une qualité associée à une relation entre concepts)
- Situation de production : pour caractériser la provenance d'un objet (fabriqué / acheté / sous-traité / co-conçu ...)
- Utilisateur : le nom ou les références d'une personne qui utilise le système PLM, avec certains droits et autorisations.
- Auteur : un utilisateur particulier avec un rôle de création d'objets (article, combinaison d'articles, document). Outre l'auteur, divers utilisateurs peuvent intervenir sur l'objet mais pour d'autres fins (valider, consulter, modifier l'objet).

En ce qui concerne les relations entre concepts, les principaux types de relation sont :

- la composition / agrégation
- la spécialisation / généralisation
- l'association

Selon le type de relation, les caractéristiques suivantes peuvent être rattachées aux relations :

- Nom, type de la relation et sens de lecture s'il s'agit d'une association, les rôles joués par les éléments liés
- Les terminaisons de part et d'autre d'une relation : type de terminaison (composition, agrégation, association...), multiplicité, rôle
- les attributs (regroupés alors dans une classe association)

2.3.2. Les contraintes

Nous distinguons 2 types de contraintes :

- Les contraintes statiques, liées à :
 - la faisabilité du produit
 - la modélisation. Par exemple, pour un modèle générique, un article spécifique doit être associé à un article générique
 - la conformité entre le modèle et son méta-modèle
- Les contraintes dynamiques relatives :
 - au temps : elles permettent de gérer l'évolution du modèle de produit dans le temps et aident les concepteurs à réaliser l'évolution.
 - au cycle de vie : elles permettent de gérer la cohérence du modèle entre les différentes phases (conception, fabrication, etc.)

Dans le cadre d'une démarche de type IDM, la transformation des contraintes est effectuée après la transformation des concepts et des relations.

2.4. Comparaison des modèles

Suite à la définition de la sémantique des méta-modèles cible et source, ces deux modèles sont comparés afin d'identifier les correspondances [21] [22]. Pour cela, un concept est proposé ; le degré de la similarité qui permet de comparer les concepts des méta-modèles d'un point de vue syntaxique ou sémantique. La similarité entre un concept de MM1 et un concept de MM2 est évaluée selon divers critères. Si deux concepts sont complètement similaires, et donc identiques, le degré de la similarité S_k est égal à 1. S_k est égal à 0 s'il n'y a aucune similarité selon le critère k . Dans le cas d'une similarité partielle, ce degré est évalué entre 0 et 1.

Notons S_k , le degré de similarité entre un concept i de MM1 et un concept j de MM2, selon le critère n : $S_k(i,j) = \text{Similarité}(\text{Concept } i, \text{Concept } j)$

Le degré de similarité global entre les concepts i et j , $S(i,j)$, est alors calculé à l'aide d'un poids qui représente l'importance de chaque critère :

$$S(i,j) = \sum_{k=1}^n (\text{Poids}(k) * S_k(i,j))$$

n est le nombre de critères à évaluer. $\text{Poids}(k)$ représente l'importance d'un critère k par rapport aux autres critères. C'est un nombre compris entre 0 et 10.

Ainsi, après calcul des similarités, nous supposons qu'un concept i est le correspondant d'un concept j si $C < S(i,j) < N$.

Les valeurs de C et N peuvent être modifiées en fonction du cas, mais par défaut :

$$C = 10 * n \text{ et } N = 10 * n / 2.$$

Ces corrélations proposées pour calculer C et N sont raisonnables car C avec cette formule définit le maximum de la similarité et N définit une similarité acceptable.

Après avoir effectué le calcul de S pour un concept *i*, nous pouvons identifier la similarité. Pour un concept *i* du méta-modèle source, quatre situations sont possibles :

- il existe dans le méta-modèle cible un concept correspondant et qui du même type que le concept *i* (attribut, classe, relation, etc.)
- il existe dans le méta-modèle cible un concept correspondant mais qui n'est pas du même type que le concept *i*.
- il n'existe pas dans le méta-modèle cible un concept correspondant à *i* mais il est possible de le construire à l'aide des informations disponibles.
- il n'existe pas dans le méta-modèle cible un concept correspondant à *i* et il est impossible de le construire à partir des informations disponibles.

Dans le dernier cas, et en fonction des possibilités de modification dans le méta-modèle cible du système PLM considéré, il y a soit une perte d'informations (lorsque le concept *i* ne peut être représenté dans le méta-modèle cible), soit une création de nouvelles entités dans le méta-modèle cible (pour représenter le concept *i*).

Le premier cas est un cas simple et la correspondance établie peut être utilisée pour la transformation.

Le deuxième cas correspond à un changement de granularité des entités considérées (par exemple : un concept représenté par une classe dans le méta-modèle source se trouve représenté par un attribut dans le méta-modèle cible). Ce cas ne pose pas de problème majeur, sauf dans le cas où ce changement de granularité entraîne un manque dans les informations disponibles.

Le troisième cas est le cas le plus difficile à traiter. D'abord, parce qu'il faut être capable de le distinguer du deuxième cas, et pour cela disposer d'un moyen pour s'assurer qu'on peut trouver la correspondance. Ensuite, parce qu'il n'existe pas de méthode structurée et universelle pour identifier la correspondance. Dans ce travail de thèse, une méthode générale est proposée pour les cas plus fréquents ; lorsque les informations nécessaires à la mise en correspondance proviennent des niveaux inférieurs (niveau modèle), la transformation est dite dans ce cas « inter-niveaux ».

2.4.1. Transformation

Le résultat du calcul de similarité est une liste de degrés de similarité pour chaque concept de méta-modèle cible et chaque concept de méta-modèle source. Avec cette liste, le concept cible correspondant pour chaque concept source peut être identifié ; c'est celui qui a le degré de similarité le plus élevé.

Une fois, les correspondances identifiées, il est alors possible de procéder à la transformation à l'aide de l'approche IDM pour assurer l'évolution des modèles sources considérés.

La figure 3 illustre l'ensemble de la démarche proposée.

Cette démarche commence par identifier et préciser la problématique d'évolution posée d'un point de vue métier. Ensuite, il s'agit de traduire le point de vue métier de la problématique

d'évolution en un point de vue informatique. Cette traduction permet de décider de la suite du processus d'évolution à suivre en choisissant un des quatre scénarios d'évolution possibles. Ensuite, en fonction du cas identifié, le processus se poursuit en définissant le cadre de similarité et enfin en exécutant la transformation.

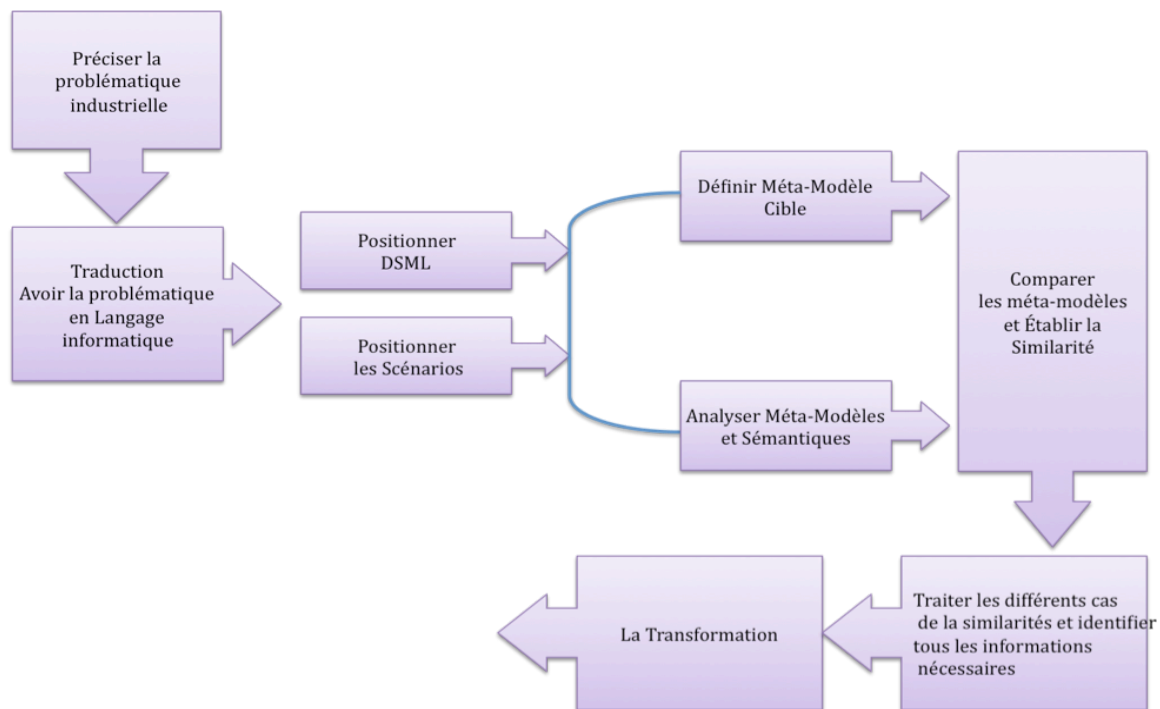


Figure 3 – Démarche générale d'évolution des modèles produit

3. Exemple d'évolution : du produit spécifique au produit générique

Pour illustrer notre approche, nous nous appuyons sur un cas industriel avec un contexte de migration d'un système P.L.M. dont le méta-modèle gère uniquement les produits spécifiques vers un système P.L.M. dont le méta-modèle permet de gérer les produits génériques.

Dans ce cas, le problème est de construire les modèles de produit générique à partir des modèles de produit spécifique tout en mettant en œuvre un mécanisme pour identifier les concepts dans le système cible.

Ce type d'évolution correspond au scénario 2 (cf. § 3.1), dans lequel le méta-modèle cible n'est pas défini à priori. L'approche adoptée pour résoudre le problème se doit d'inclure également la construction du méta-modèle cible.

Avant de présenter cette approche (§ 4.2), nous présentons d'abord la notion de généricité de produit (§ 4.1).

3.1. Généricité de produit

Afin de gérer efficacement les informations autour du produit, la nomenclature de produits peut être gérée selon trois niveaux d'abstraction [23] :

- Niveau exemplaire : il s'agit du niveau le plus concret de la nomenclature et représente la décomposition d'un produit physique donné, fabriqué et destiné à un client. L'évolution de

cette nomenclature permet de gérer les différentes phases de cycle de vie « as-designed », « as-built », « as-maintained ».

- Niveau spécifique : c'est un niveau plus abstrait que le niveau exemplaire qui représente la décomposition d'un ensemble de produits de même type, décrivant la structure commune à l'ensemble des exemplaires fabriqués selon un même modèle (une même configuration),
- Niveau générique : c'est le niveau le plus abstrait qui représente la décomposition de toute une ligne ou gamme de produits spécifiques, incluant des options et variantes.

La nomenclature générique est une structuration de données produit dont les principales caractéristiques sont énumérées ci dessous [24] [25] [26] [27] :

- Généricité et abstraction : du fait que la nomenclature générique est une représentation d'un ensemble de produits (une gamme de produits), elle contient des articles ayant plusieurs variantes possibles (la couleur par exemple) et des articles optionnels (un GPS sur un vélo). Les attributs attachés à ces articles ne sont alors valués que lors du passage à une nomenclature spécifique, voire à la nomenclature exemplaire. La généricité visée pour cette structure de données fait qu'elle est abstraite (aucun produit réel ne peut être l'instanciation de la nomenclature générique). Comme précisé précédemment, la généricité de cette nomenclature a pour but de factoriser les informations communes entre les composants de divers produits physiques. C'est cette factorisation qui différencie cette structure de données d'une simple gestion de la configuration de produit.
- Hiérarchisation et spécialisation : la nomenclature est une hiérarchie arborescente qui contient deux types de relation : la composition et la spécialisation (cf. figure 4). La composition matérialise la relation entre un composite et ses composants. Cette composition peut exister entre articles, mais aussi entre toutes entités métiers telles que les documents (dossiers composés de fiches). Dans cette hiérarchie, un composite ne peut être dérivé de ses composants (il n'y a pas de structure circulaire). La spécialisation met en évidence la relation entre un article et ses variantes. Chaque variante hérite des propriétés de l'article générique dont elle est variante, mais elle hérite également de tous les liens que cet article générique entretient avec d'autres articles et entités métiers (y compris donc sa composition).

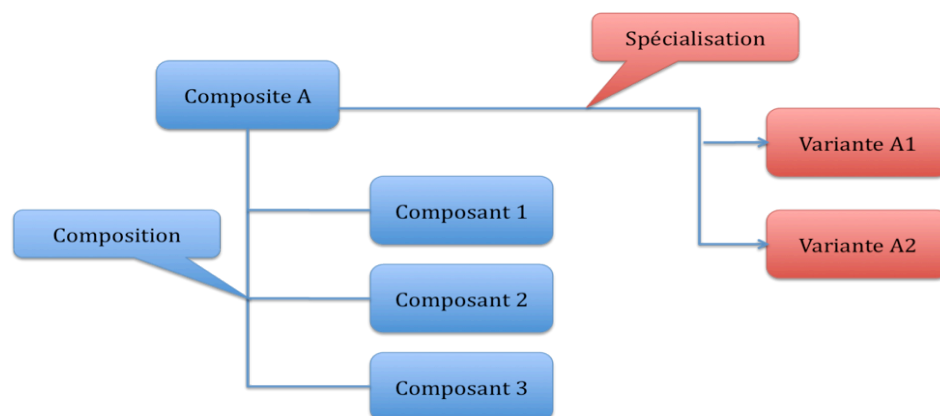


Figure 4 - Les relations dans la nomenclature générique

- Type de composants : la nomenclature générique peut contenir trois types de composants. Un composant peut être constant, ce qui signifie que ce composant existe dans tous les produits spécifiques. Il peut être également variant, c'est à dire ayant plusieurs variantes exclusives possibles et dans chaque produit spécifique, seule une variante peut être incluse. Finalement, un composant peut être optionnel ; il appartient à un produit spécifique ou exemplaire selon le choix de l'entreprise ou du client. Il est également possible de décrire des niveaux de compatibilité des composants optionnels et des composants à variantes entre eux (exemple toit ouvrant avec carrosserie 2 portes uniquement).

3.2. Approche d'évolution proposée

Comme nous l'avons déjà évoqué, le scénario correspondant au cas industriel traité est le deuxième. Le méta-modèle cible n'est préalablement pas identifié ; notre proposition inclut une démarche qui identifie et formalise les concepts essentiels pour construire les modèles génériques.

Figure 5 explique notre point de vue sur le contexte de modèle générique et spécifique de produit et leurs positions dans la structure IDM. Avec cette approche, le modèle générique conforme au méta-modèle générique et à partir de ce modèle et à l'aide de spécialisation, le modèle spécifique se produit. La spécialisation se réalise par la fixation des valeurs pour les variantes ou choisir entre les options présentées et proposées dans le modèle générique. Le modèle spécifique peut avoir plusieurs couches de spécificité, c'est à dire qu'on peut avoir plusieurs modèles spécifiques dans lesquelles le nombre des attributs ou options déterminées se varie. Mais tous ces modèles conforment au méta-modèle spécifique. L'objet exemplaire, c'est à dire l'instance de modèle, se trouve au niveau M0. L'instanciation est le processus de création des objets à partir des modèles spécifiques. Ces objets représentent les produits exemplaires et sont indépendantes l'un de l'autre. Dans le figure 5 il est montré qu'un objet peut se crée à partir d'un modèle générique. Cela veut impérativement dire que au moment donné, on peut créer une nomenclature exemplaire d'un produit directement de son modèle générique en traversant les modèles spécifiques.

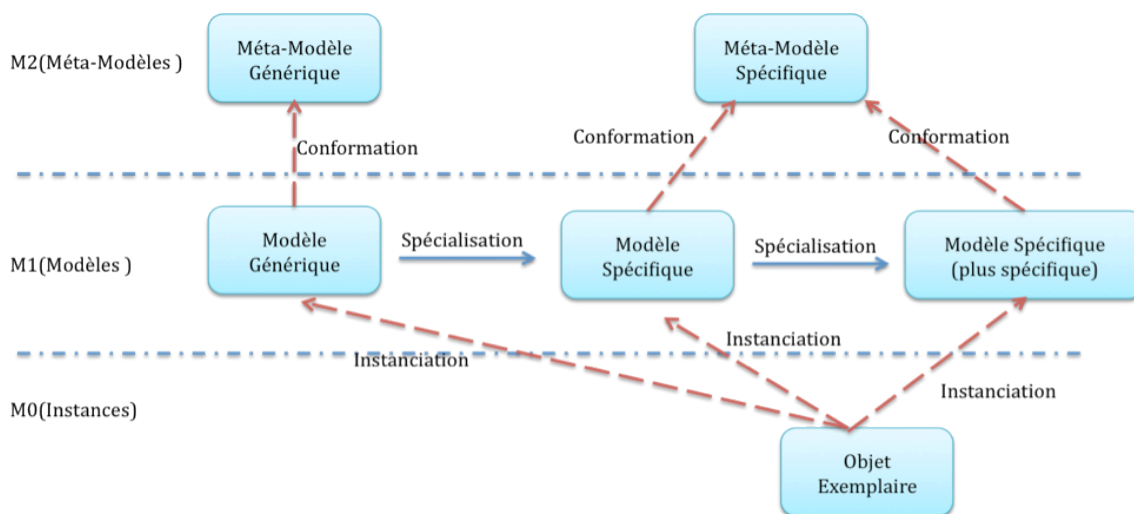


Figure 5- Contexte des modèles générique/spécifique avec l'approche IDM

L'approche proposée pour la structuration est illustrée dans la figure 6. Le principe général de cette approche est la structuration de la nomenclature générique à partir des données et des informations existantes dans le système source, c'est à dire les nomenclatures spécifiques.

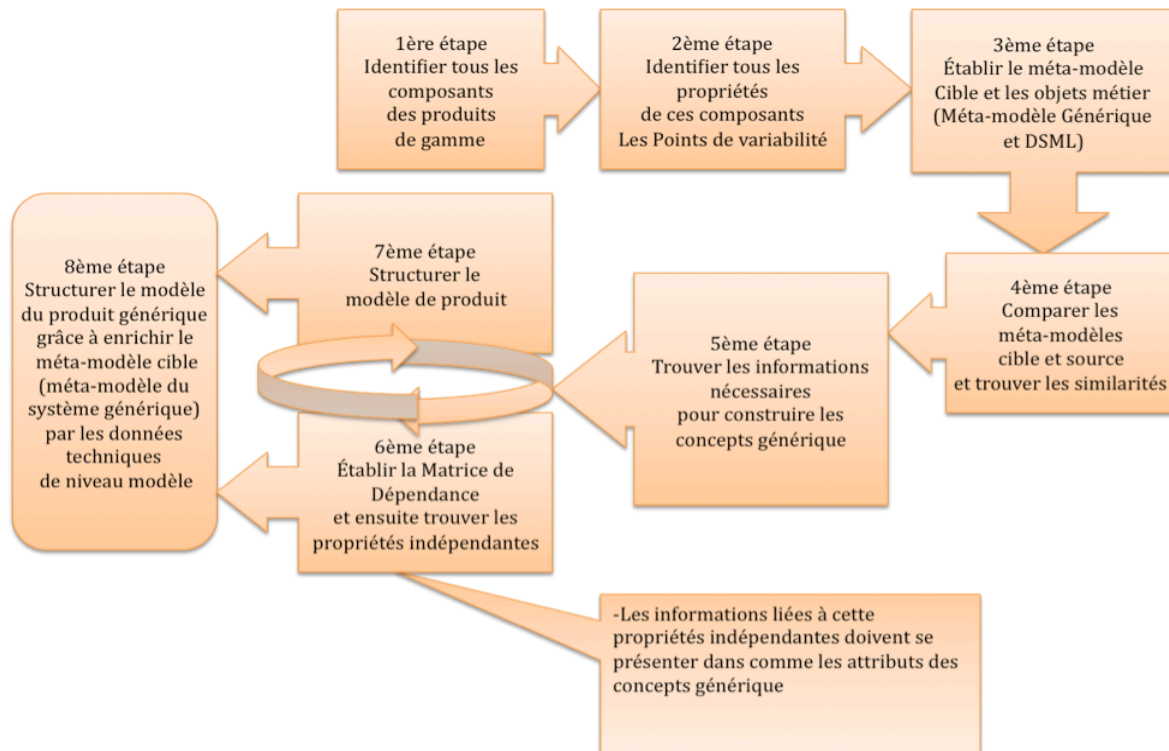


Figure 6- Démarche proposée pour la structuration de la nomenclature générique

Pour illustrer chacune des étapes de la démarche proposée, nous nous appuyons sur un exemple industriel relatif à la structuration de la nomenclature générique d'articles culinaires de type poêles.

La première étape commence par identifier tous les éléments entrant dans la composition des divers produits d'une gamme. La liste de ces composants doit couvrir l'ensemble des produits spécifiques.

Toute poêle, quel que soit son type, est composée des éléments suivants :

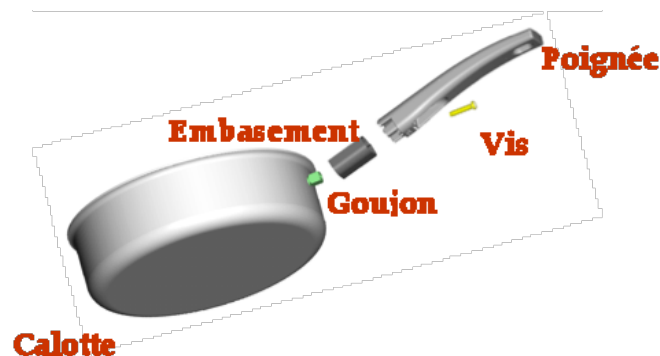


Figure 7 - Les composants d'une poêle

La deuxième étape consiste à identifier toutes les propriétés qui sont à l'origine de la diversité des produits spécifiques au sein d'une gamme. Elle conduit à la définition d'une liste de points de variabilité. Le concept principal de point de variabilité provient du fait que dans une gamme de

produits, certaines propriétés d'éléments de la nomenclature différencient un produit spécifique d'un autre. Ces propriétés sont les sources de diversité dans une gamme de produits [28]. Afin de construire la nomenclature générique pour une gamme de produits, il faut identifier et gérer ces sources de diversité.

Les éléments à l'origine de la diversité des poêles sont listés dans le tableau ci-dessous. Pour chaque élément, nous explicitons les propriétés qui sont sujettes à la variabilité.

Eléments	Propriétés
Poêle (le produit même)	Dimension de la poêle
	Famille de la poêle
Calotte	Dimension de la calotte
	Famille de la calotte
Outillage de la Calotte	Codification de l'outillage de la calotte
Disque Frappé	Dimension du disque frappé
Outillage de Disque Frappé	Codification de l'outillage de disque frappé
Les Equipements (Poignée, Goujon, Embasement et Vis)	Famille des équipements
	Les paramètres liés aux dimensions des équipements, ainsi que la dimension de la poêle (spécialement pour l'embasement)
Couvercle	Dimension du couvercle
	Famille du couvercle
Outillage de Couvercle	Codification de l'outillage du couvercle

Tableau 2 - Les éléments à l'origine de la diversité des poêles

Dans la troisième étape, qui peut être faite parallèlement à la deuxième, la nature des éléments de la nomenclature est définie. Il s'agit de distinguer les articles physiques, des documents, des plans, etc. La nature de ces objets métiers permet de définir les éléments du méta-modèle de la nomenclature. Ce méta-modèle permet de construire la nomenclature en mettant en évidence les objets métiers et leurs relations.

Pour notre exemple industriel, le méta-modèle de la nomenclature est défini sur la figure 8 :

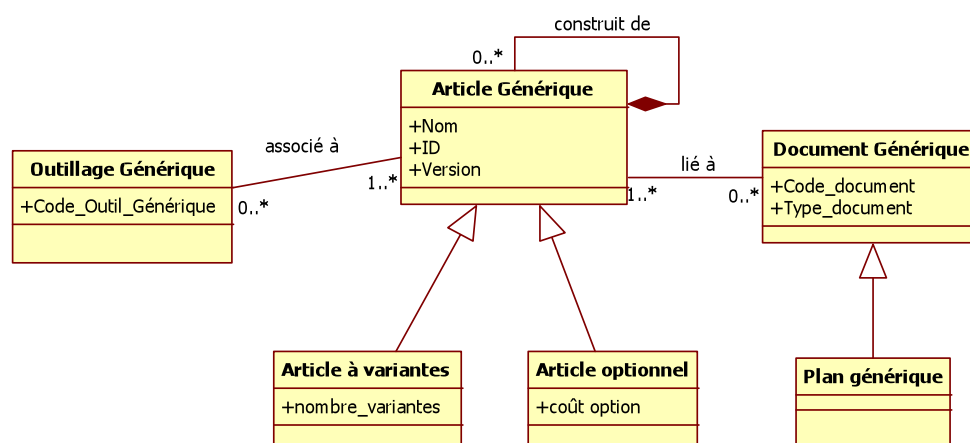


Figure 8- Le méta-modèle de la nomenclature générique

Ce méta-modèle doit être comparé au méta-modèle source, afin de trouver les correspondances. La quatrième étape consiste à comparer ce méta-modèle avec le méta-modèle source afin d'identifier les correspondances entre les concepts en prenant en compte les similarités entre eux. Ensuite, le type de similarité est identifié à partir des différents cas de similarité.

La figure 9 illustre la comparaison des méta-modèles, pour l'exemple industriel considéré.

Dans cette figure, les traits foncés représentent des similarités exactes entre les concepts, c'est à dire que les instances de ces concepts sont identiques dans les méta-modèles cible et source.

Les concepts dans les carrés sont les concepts de méta-modèle cible qui n'ont aucune correspondance dans le méta-modèle source. On ne peut pas trouver de concepts similaires pour eux dans le méta-modèle source. Comme ils n'ont pas d'équivalent dans le système source, ils ne sont pas instanciés (ou valués) et ils restent donc vides. Pour éviter cela, nous devons définir une méthode pour trouver les connaissances nécessaires à partir des modèles ou du méta modèle source et qui nous permettent d'instancier ou valuer ces concepts dans le système cible. A titre d'exemple, le document générique est un concept lié à l'article générique. Il est ainsi identifié après la construction du concept d'article générique. Cela veut dire qu'il reste sans correspondance. Par contre, le lien d'héritage revient à une factorisation des informations des articles spécifiques vers leur article générique associé. Cela implique l'élimination de ces informations dans l'article spécifique lorsque l'article générique est construit.

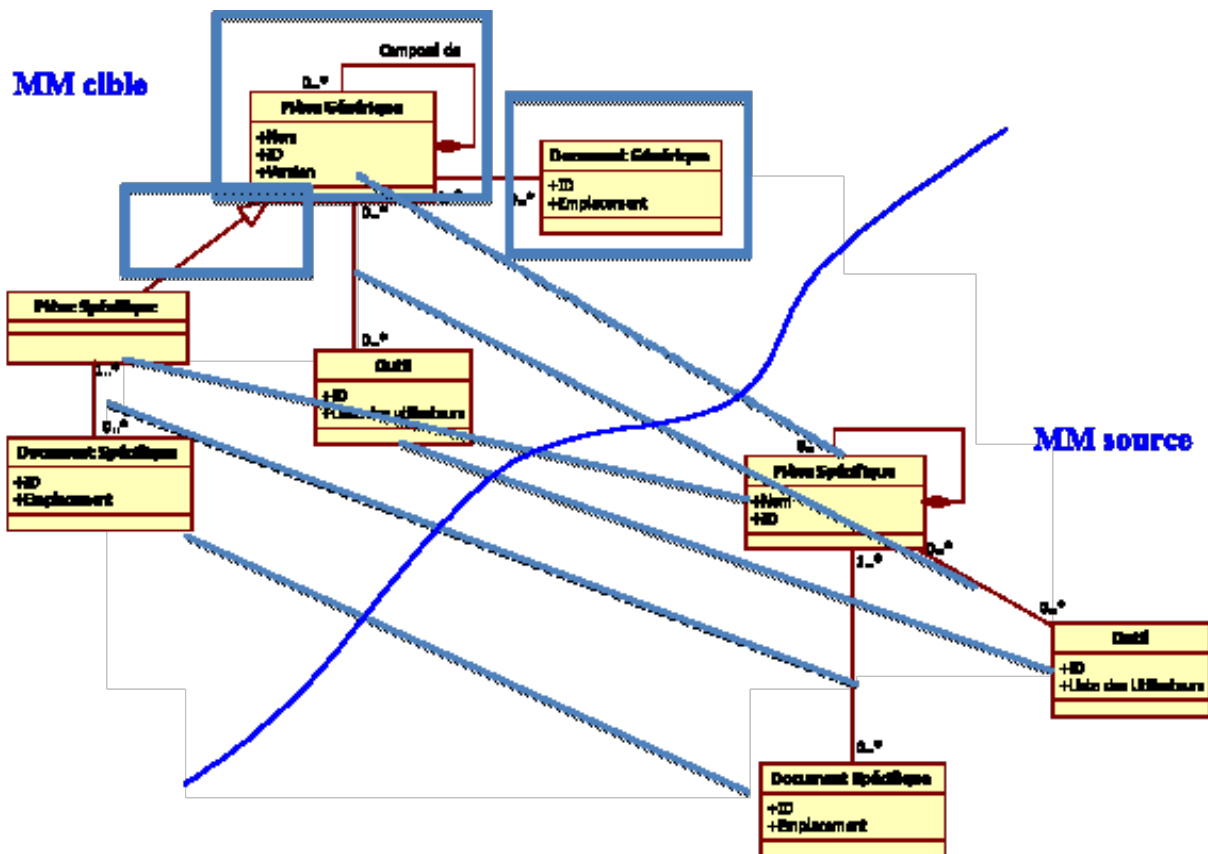


Figure 9 - Comparaison de méta-modèles source et cible

Les étapes suivantes ont pour objectif d'identifier les informations concernant la construction du concept d'article générique, notamment en s'appuyant sur les informations existantes dans les modèles cibles. Pour établir les correspondances au niveau méta-modèle, nous utilisons les informations qui sont présentes au niveau inférieur ; celui des modèles.

Pour illustrer cette démarche, nous présentons l'approche suivie pour construire le concept « article générique » à partir des instances d'article spécifique.

Les cinquième, sixième et septième étapes sont complémentaires. Elles commencent parallèlement et sont menées d'une manière itérative. Dans la sixième étape, nous concevons le modèle de la nomenclature. Ce modèle définit la structure de la nomenclature, en expliquant son organisation, l'emplacement des attributs, et les entités. Au sein de cette étape, des regroupements de composants ou des rajouts de nouveaux éléments abstraits sont également effectués. La structure obtenue est ensuite complétée à l'aide de la septième étape. Le but de cette septième étape est d'identifier les dépendances entre les points de variabilité listés à l'issue de la deuxième étape. Pour identifier ces dépendances, la matrice de dépendance (DSM : Dependency Structure Matrix) est utilisée [29] [30]. Dans cette matrice de dépendances, tous les points de variabilité sont comparés ; si la propriété en ligne dépend de la propriété en colonne, la case correspondante est cochée.

Deux types de dépendance sont distingués entre les points de variabilité : la dépendance forte et la dépendance faible. La dépendance forte signifie que si un point de variabilité se fixe, l'autre est obligatoirement fixé. En cas de dépendance faible, le point de variabilité peut avoir plusieurs valeurs. Dans ce cas il n'existe pas une correspondance mais une restriction [31].

Après la construction de la matrice, il faut intervertir les lignes et les colonnes de la matrice DSM afin d'avoir des sous-matrices indépendantes. La figure 10 illustre le principe de cette technique. Dans cet exemple théorique, il y a 4 propriétés à comparer ; la propriété A dépend de D et la propriété C dépend de B.

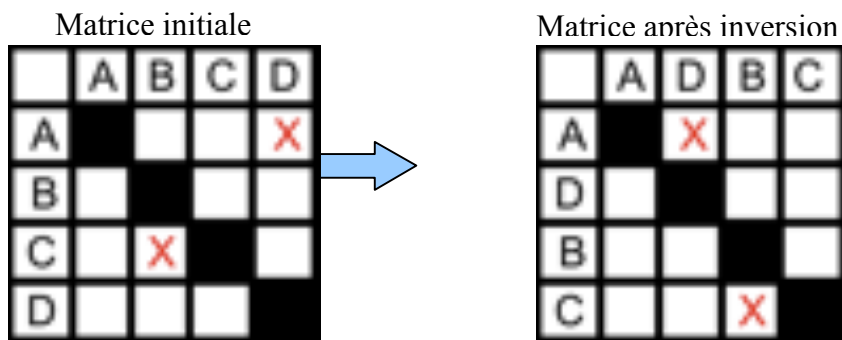


Figure 10 – Exemple de matrice de dépendance

Ces propriétés appartiennent aux composants Article 1, Article 2, Article 3 et Article 4 (cf. figure 11 à gauche). Le résultat de cette 7^{ème} étape sert à compléter le travail déjà accompli dans la 6^{ème} étape. Chaque sous-matrice indépendante révèle un regroupement possible de composants (les composants associés aux propriétés incluses dans la sous-matrice). La partie droite de la figure 11 illustre ce principe avec 2 regroupements de composants.

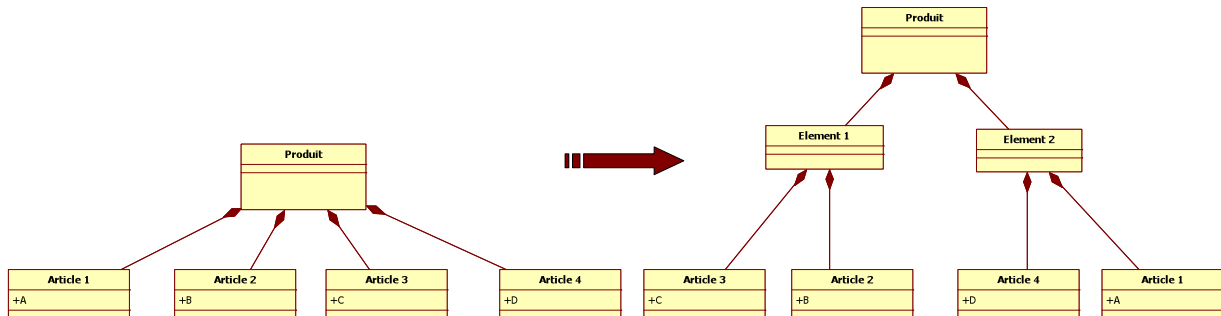


Figure 11 - Regroupement des composants

Ce regroupement introduit 2 nouveaux éléments intermédiaires dans la nomenclature. Ils sont abstraits du fait qu'aucun composant physique ne correspond à ces éléments NON ce sont des sous assemblages pas plus abstraits que le « produit ». Les dépendances entre propriétés qui nous ont amené à identifier ces composants peuvent aussi aider à factoriser ces propriétés et construire des attributs pour ces composants en éliminant les attributs de chacun de leurs composants. Cela est illustré sur la figure 8.

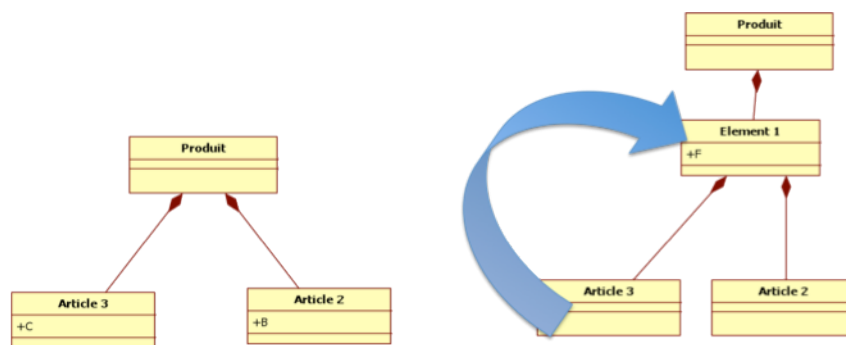


Figure 12- Factorisation

Dans la figure 12, les propriétés C et B sont similaires de telle façon que nous pouvons les éliminer et les monter au niveau abstrait, et donc les factoriser. Mais cela n'est pas réalisable pour tous les cas.

À ce stade, il y a 3 possibilités de modélisation :

1. Il existe une propriété commune (dépendance forte) entre les composants, que nous remontons alors aux éléments intermédiaires, comme dans la figure 12. C et B sont soit identiques soit en identifiant F, tous les deux seront reconnues.

2. Il n'existe pas de propriété commune mais il est possible d'introduire une nouvelle propriété abstraite pour les éléments intermédiaires qui factorise quelques propriétés des composants du groupe. Grace à ça, le nombre des propriétés indépendantes diminue.
3. Il n'y a aucune possibilité d'éliminer les propriétés des composants dans le groupe. Dans ce cas, l'élément intermédiaire de ce groupe n'aurait pas d'attribut et, au minimum, nous arrivons à regrouper les composants relatifs de produit et proposer une structure plus logique.

De toute façon, avec ce regroupement, les contraintes associées aux propriétés de composants se regroupent, ce qui facilite leur définition et leur modification.

Si nous ajoutons de nouvelles propriétés, il faut reconstruire la matrice et refaire le modèle de la nomenclature. Le résultat de ces étapes est une structure de nomenclature qui factorise les informations correspondant aux propriétés des composants de produit d'une manière plus optimale.

La nomenclature générique construite est fonction des propriétés qui restent indépendantes. Pour chaque ensemble de valeurs permises de ces propriétés, il y a une nomenclature générique correspondante. Les nomenclatures spécifiques, et par conséquent les nomenclatures exemplaires, sont construites en fixant les propriétés variantes et optionnelles.

En appliquant les étapes 6 et 7 au cas industriel considéré, la matrice de dépendances obtenue est la suivante :

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
P1	■											
P2		■										
P3	▨		■									
P4		▨		■								
P5			▨		■							
P6		▨				■						
P7	▨						■					
P8		▩						■				
P9	▨								■			
P10			▨	▨						■		
P11							▨				■	
P12									▨			■

- P1: Dimension de produit
- P2: Famille de produit
- P3: Dim de calotte
- P4: Famille de calotte
- P5: Dim des équipements
- P6: Famille des équipements
- P7: Dim de couvercle
- P8: Famille de couvercle
- P9: Dim disque frappé
- P10: Code des outillages calotte
- P11: Code des outillages couvercle
- P12: Code des outillages disque frappé

Figure 13, matrice de dépendance des pôles

La figure 14 illustre le modèle spécifique pour une poêle, qui est utilisé dans le système source :

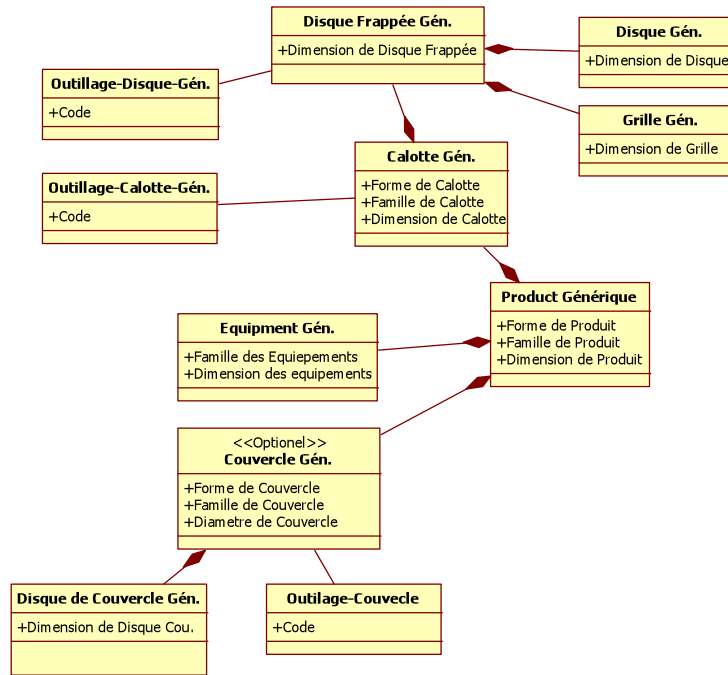


Figure 14 - Modèle de la nomenclature générique

Les dépendances fortes des propriétés permettent de réduire les dépendances. Celles-ci se résument dans le tableau 3.

Dépend de



Propriété A	Propriété B
Dimension de Calotte	Dimension de Produit
Famille de Calotte	Famille de Produit
Dimension Disc Frappée	Dimension de Produit
Les paramétrés liées à la dimension des équipements	Dimension de Produit (pour l'embase, dimension de calotte)
Famille des équipements	Famille de Produit
Dimension de Couvercle	Dimension de Calotte
Famille de Couvercle	Famille de Produit
Outillage de Couvercle	Dimension et Famille de Couvercle
Outillage de Calotte	Dimension et Famille de Calotte
Outillage de Disc Frappée	Dimension de Disc Frappée

Tableau 3 - Les dépendances entre les propriétés

La matrice de dépendances montre que la dimension et la famille de produit sont des propriétés indépendantes de poêle, avec lesquelles il est possible de distinguer les caractéristiques définissant la nomenclature générique. La nomenclature générique devient alors une fonction de la famille et de la dimension de produit. A titre d'exemple, une poêle générique serait la poêle 26 Cuisine (26 : dimension, Cuisine : famille)

La structure de la nomenclature de cette poêle générique est illustrée sur la figure 15.

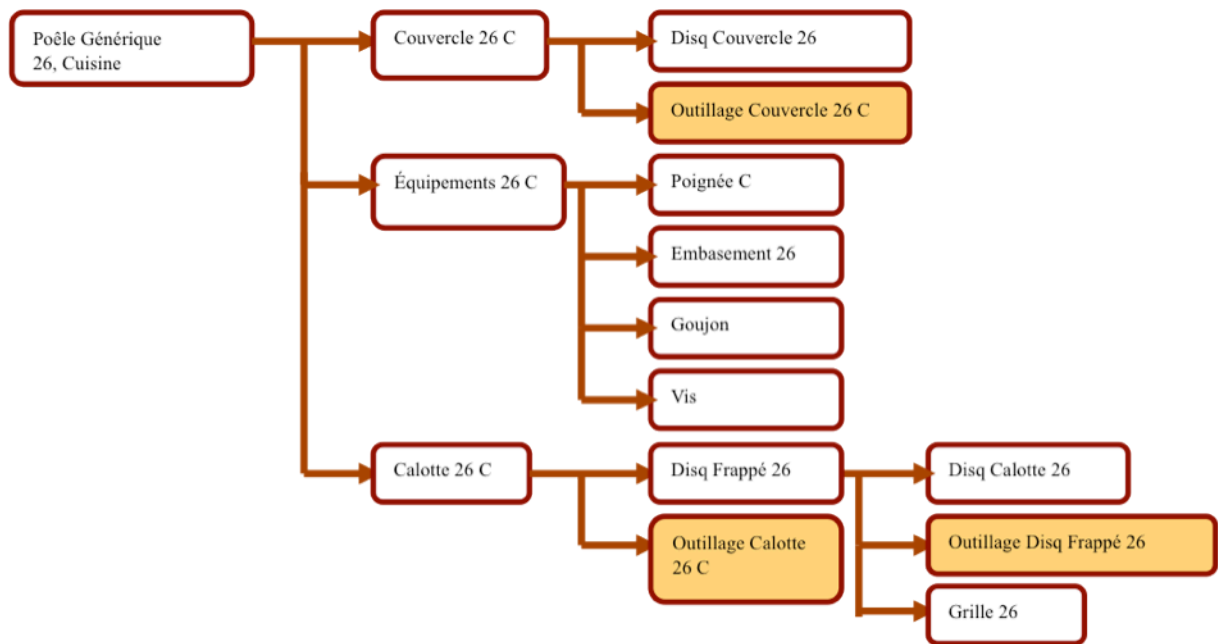


Figure 15 - Nomenclature de la poêle générique 26 Cuisine

Cette structuration permet de décrire la nomenclature générique de produit. Il faut noter que cette structure représente les éléments du modèle de poêle générique, et pas une poêle exemplaire. Bien évidemment la nomenclature d'une poêle exemplaire se crée par valoriser les attributs de ce modèle ainsi que identifier les options et les variantes.

En utilisant cette démarche, nous arrivons à identifier les informations nécessaires pour construire le concept générique et enrichir les instances de ce concept dans les modèles génériques associées. Ces informations aident à identifier les attributs présents dans l'article générique.

Avec cette démarche, la transformation, dernière étape de notre approche est effectuée en adoptant la démarche suivante pour chaque concept du modèle source :

- identifier le méta-concept du méta-modèle source.
- identifier le méta-concept correspondant dans le méta-modèle cible.
- le modèle cible contient l'instance de ce méta-concept cible.
- ajouter les informations éventuelles pour enrichir les concepts de modèles cibles.

4. Conclusion

Le système PLM est l'un des outils stratégiques de l'entreprise. Ce système améliore les performances de l'entreprise en garantissant l'organisation et le partage des données techniques et des informations vitales de produit pendant son cycle de vie et parmi des partenaires internes et externes de l'entreprise. Toutefois, ces systèmes sont sujets à des changements récurrents dans l'entreprise. Les évolutions organisationnelles, le changement de l'offre produit ou encore le remplacement de logiciels PLM peuvent déclencher l'évolution du système d'information PLM. Une des structures les plus importantes dans les systèmes PLM est le modèle du produit, autour duquel s'articule les informations et processus. C'est autour du modèle produit que se concentrent nos recherches. Les causes d'évolution des modèles produits sont des éléments

signifiants qui différencient les étapes de la démarche à suivre. Les méthodes d'IDM sont utilisées afin de formaliser la transformation des modèles. En plus, cette démarche bénéficie d'un cadre de similarité spécialement développé pour la configuration de produit. Ce cadre est utilisé pour comparer les méta-modèles des systèmes cible et source et pour identifier les correspondances entre leurs concepts. Ces correspondances sont ensuite calculées grâce à des formulaires développés dans le cadre de similarité.

Un exemple industriel est illustré et résolu en appliquant cette démarche. Il s'agit de l'évolution d'un système gérant les modèles spécifiques de produit vers un système qui est capable de construire et d'utiliser les modèles génériques de produit. Cet exemple est fréquent dans différents types d'industrie qui ont besoin d'avoir la possibilité de gérer la configuration générique. Dans cette recherche, un besoin concret industriel est identifié et les démarches générales sont customisées afin de résoudre cette problématique.

Un outil informatique support à nos travaux est en cours de développement. L'objectif est de modifier cet outil de telle façon qu'il arrive à se connecter à un outil PLM, récupérer les modèles et les instances, et les transformer en fonction des besoins attendus, et ensuite les réinsérer dans le système.

L'évolution des outils PLM ne se restreint pas à l'évolution des modèles produits ; les autres composants des systèmes PLM sont aussi l'objet d'évolutions. Le besoin de développer les outils, les méthodes et les démarches pour gouverner et exécuter l'évolution de ces composants est crucial.

5. Bibliographie

- [1] F. Danesi, N. Gardan, Y. Gardan, et M. Reimeringer, "P4LM: A methodology for product lifecycle management", *Computers in Industry*, vol. 59, Mar. 2008, pp. 304-317.
- [2] D. Kiritsis, A. Bufardi, et P. Xirouchakis, "Research issues on product lifecycle management and information tracking using smart embedded systems", *Advanced Engineering Informatics*, vol. 17, 2004, pp. 189-202.
- [3] D.T. Liu et X. William Xu, "A review of web-based product data management systems", *Computers in Industry*, vol. 44, Avr. 2001, pp. 251-262.
- [4] Z. Irani, M. Themistocleous, et P.E.D. Love, "The impact of enterprise application integration on information system lifecycles", *Information & Management*, vol. 41, Dec. 2003, pp. 177-187.
- [5] R.M. Rangan, S.M. Rohde, R. Peak, B. Chadha, et P. Bliznakov, "Streamlining Product Lifecycle Processes: A Survey of Product Lifecycle Management Implementations, Directions, and Challenges", *Journal of Computing and Information Science in Engineering*, vol. 5, 2005, pp. 227-237.
- [6] M. Bertoni, M. Bordegoni, U. Cugini, D. Regazzoni, et C. Rizzi, "PLM paradigm: How to lead BPR within the Product Development field", *Computers in Industry*, vol. 60, Sep. 2009, pp. 476-484.
- [7] C. Sung et S. Park, "A component-based product data management system", *The International Journal of Advanced Manufacturing Technology*, vol. 33, June 2007, pp. 614-626.
- [8] J.D. Golovatchev et O. Budde, "Next Generation PLM - an integrated approach for the Product Lifecycle Management", *Proceedings of ICCPR2007: International Conference on Comprehensive Product Realization 2007*, Beijing - China.

- [9] B. Eynard, T. Gallet, L. Roucoules, et G. Ducellier, "PDM system implementation based on UML", *Mathematics and Computers in Simulation*, vol. 70, Feb. 2006, pp. 330-342.
- [10] V. Srinivasan, "An integration framework for product lifecycle management", *Computer-Aided Design*, In Press, Corrected Proof available online.
- [11] S. Zina, M. Lombard, L. Lossent, et C. Henriot, "Generic modeling and configuration management in Product Lifecycle Management", *Computational Engineering in Systems Applications, IMACS Multiconference on*, Oct 2006, pp. 1252-1258.
- [12] B. Eynard, T. Gallet, P. Nowak, et L. Roucoules, "UML based specifications of PDM product structure and workflow", *Computers in Industry*, vol. 55, Dec. 2004, pp. 301-316.
- [13] J. M. Favre et J. Musset, "Rétro-ingénierie dirigée par les métamodèles : Concepts, Méthodes et Outils", Actes d'IDM06 : 2èmes journées sur l'ingénierie dirigée par les modèles, Juin 2005.
- [14] OMG, "MDA Guide V1.0.1" Jun. 2001.
- [15] M. Brambilla, P. Fraternali, et M. Tisi, "A Transformation Framework to Bridge Domain Specific Languages to MDA", *Models in Software Engineering, Lecture Notes in Computer Science, 2009, Volume 5421/2009*, pp. 167-180.
- [16] F. Krause et U. Kaufmann, "Meta-Modelling for Interoperability in Product Design," *CIRP Annals - Manufacturing Technology*, vol. 56, 2007, pp. 159-162.
- [17] A. Cicchetti, D. Di Ruscio, et A. Pierantonio, "A Metamodel Independent Approach to Difference Representation," *Journal of Object Technology*, vol. 6, 2007, pp. 165-185.
- [18] M. Kofman et E. Perjons, "MetaDiff- a Model Comparison Framework", Internal report - Department of Computer and System Sciences - Stockholm University and Royal Institute of Technology Stockholm, Sweden, 2004.
- [19] Y. Lin, "A model transformation approach to automated model evolution", PhD Thesis, University of Alabama at Birmingham, 2007.
- [20] A. Cicchetti, "Difference Representation and Conflict Management in Model-Driven Engineering", *PhD Thesis - Università degli Studi dell'Aquila*, April 2008.
- [21] G. Sunyé, A. Le Guennec, et J. Jézéquel, "Using UML Action Semantics for model execution and transformation", *Information Systems*, vol. 27, Sep. 2002, pp. 445-457.
- [22] J.E. Rivera et A. Vallecillo, "Representing and Operating with Model Differences", *Objects, Components, Models and Patterns In TOOLS EUROPE 2008*, Vol. 11, 2008, pp. 141-160.
- [23] L. Gzara, D. Rieu, et M. Tollenaere, "Product information systems engineering: an approach for building product models by reuse of patterns", *Robotics and Computer-Integrated Manufacturing*, vol. 19, June 2003, pp. 239-261.
- [24] T. Männistö, H. Peltonen, T. Soininen, et R. Sulonen, "Multiple abstraction levels in modelling product structures", *Data & Knowledge Engineering*, vol. 36, Jan. 2001, p. 55-78.
- [25] J. Jiao et M.M. Tseng, "An Information Modeling Framework for Product Families to Support Mass Customization Manufacturing", *CIRP Annals - Manufacturing Technology*, vol. 48, 1999, pp. 93-98.
- [26] P. Wu, K. Olsen, et P. Saetre, "Visualizing the Construction of Generic Bills of Material", *Recent Advances in Visual Information Systems*, 2002, pp. 302-310.
- [27] A. Jose Flores et M. Tollenaere, "Modular and platform methods for product family design: literature analysis", *Journal of Intelligent Manufacturing*, vol. 16, June 2005, pp. 371-390.
- [28] F. Zeng et Y. Jin, "Study on product configuration based on product model", *The International Journal of Advanced Manufacturing Technology*, vol. 33, July 2007, pp. 766-771.
- [29] A. A. Yassine et D. Braha, "Complex Concurrent Engineering and the Design Structure Matrix Method," *Concurrent Engineering*, vol. 11, Sep. 2003, pp. 165-176.
- [30] G. HARMEL, "Vers une conception conjointe des architectures du produit et de l'organisation du projet dans le cadre de l'Ingénierie Système", *Thèse de doctorat - Université de Franche-Comté*, 2007.
- [31] A. A. Yassine, T. Zambito, et D. E. Whitney, "Assessment of rework probabilities for simulating product development processes using the design structure matrix (DSM)", *Proceedings of DETC '01 ASME 2001 International Design Engineering Technical Conferences*

Computers and Information in Engineering Conference, Pittsburgh - Pennsylvania, 2001.