



Science Arts & Métiers (SAM)

is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <https://sam.ensam.eu>
Handle ID: [.http://hdl.handle.net/10985/10152](http://hdl.handle.net/10985/10152)

To cite this version :

Mehdi IRAQI-HOUSSAINI, Lionel ROUCOULES, Mathias KLEINER - Vers une ingénierie produit collaborative et interopérable basée sur les modèles - Revue Ingénierie des Systèmes d'Information - Vol. 17, n°14, p.79-94 - 2012

Any correspondence concerning this service should be sent to the repository

Administrator : scienceouverte@ensam.eu



Vers une ingénierie produit collaborative et interopérable basée sur les modèles

Un cadre général pour l'acquisition des données métier

Iraqi-Houssaini Mehdi*— **Kleiner Mathias***— **Roucoules Lionel***

**Arts et Métiers ParisTech ; CNRS, LSIS, 2 cours des Arts et Métiers, 13697 Aix-en-Provence, France*

mehdi.iraqi-houssaini@ensam.eu

mathias.kleiner@ensam.eu

lionel.roucoules@ensam.eu

RÉSUMÉ. Depuis plusieurs années les processus de conception de produit ont évolué, la maîtrise de l'information est devenue un verrou majeur à contrôler pour une collaboration efficace. Afin de favoriser cette collaboration, il est nécessaire de formaliser les processus et les données métier dans un environnement théorique et logiciel ouvert et dynamique. Nous présentons dans cet article les travaux de recherche que nous avons menés en utilisant l'Ingénierie Dirigée par les Modèles (IDM) pour assurer l'interopérabilité des outils métier. La principale difficulté est d'être capable de projeter les données manipulées par un outil métier depuis / vers cet espace de collaboration de données métier. Cet article propose une méthodologie générale pour effectuer ces opérations, l'approche est consolidée par son expérimentation sur un cas d'étude.

ABSTRACT. For several years the processes of product design have evolved, data management became a major issue that need to be controlled for an effective collaboration. To foster this collaboration, it is necessary to formalize the processes and data in a theoretical / software open source and dynamic environment. The paper present here the research work conducted using Model Driven Engineering (MDE) to ensure the interoperability of expert software tools. The main difficulty is being able to project data handled by an expert tool to / from the space of business data collaboration. This article proposes a general methodology to perform these operations; the approach is consolidated by its experimentation on a case study.

MOTS-CLÉS : Ingénierie dirigée par les modèles, Conception de produit, Interopérabilité, projection de données.

KEYWORDS: Model Driven Engineering, Product Design, Interoperability, data projection.

1. Introduction

La Conception collaborative, ou plus généralement l'ingénierie collaborative, est aujourd'hui un moyen essentiel pour augmenter la valeur dans les procédés industriels. Les outils classiques dans ce domaine portent principalement d'une part sur une forte intégration des relations produit / processus, et d'autre part sur la fédération des connaissances liées aux différentes expertises. Cette approche d'intégration n'a pas été totalement efficace en ingénierie multi-acteurs qui impliquent un large éventail d'expertises. De nouvelles approches de collaboration tentent d'aller plus loin en considérant les aspects sémantiques du système d'ingénierie comme une nécessité pour les aspects techniques.

Dans nos travaux de recherche, nous considérons que la fédération des données est à la base d'une interopérabilité organisationnelle flexible et efficace. Dans ce contexte, nous pensons que l'ingénierie dirigée par les modèles (Bezivin, 2006 ; Favre et al., 2006) est une approche efficace pour assurer l'interopérabilité dans les deux aspects sémantique et technique. Cette approche traite un double niveau de modélisation processus (chainage de données) / produit (multi vues des données du produit conçu) en s'appuyant sur les principes de l'ingénierie dirigée par les modèles.

Dans le premier chapitre nous donnons une brève introduction à l'ingénierie dirigée par les modèles, à la conception produit et à l'application de l'ingénierie dirigée par les modèles à la conception produit. Dans le deuxième chapitre nous proposons un cadre général pour l'acquisition des données métier en se basant sur les principes de l'ingénierie dirigée par les modèles. Ce cadre est expérimenté dans le chapitre 3 par une étude de cas, suivi par une conclusion et nos perspectives.

1.1 Introduction à la conception produit

L'activité de modélisation système est à la base de tout processus de conception (Noel et al., 2004). Le rôle de celle-ci c'est essentiellement de développer une description abstraite d'un système physique. Le modèle système dépend du point de vue selon lequel on observe le système, mais aussi de l'utilisation que l'on souhaite faire de ce modèle système au sein du processus de conception. Un modèle système offre donc la possibilité de valider les caractéristiques d'une partie du système ou de l'ensemble de ce dernier.

Ces modèles systèmes sont définis et manipulés par des outils experts dans leurs propres espaces techniques. Afin de communiquer entre eux, ces outils utilisent parfois des standards (par exemple STEP (ISO-10303, 1994; ISO-10303-21, 1994)). Cependant il subsiste de nombreux problèmes d'interopérabilité, à la fois syntaxiques et sémantiques (Noel et al., 2004).

Dans le contexte industriel, l'activité de conception est constituée de tâches collaboratif et à distance qui ont besoin de relier l'ensemble des connaissances provenant de différents experts métiers (analyse fonctionnelle, les composants et le choix des matériaux, analyse de la structure, processus de fabrication ...). Aujourd'hui la plupart de ces tâches sont assistées par ordinateur. Deux grandes catégories d'outils informatiques peuvent être répertoriées:

- Le PLM (Product Lifecycle Management) (Debaeckerd, 2004) s'inscrit dans le cadre d'une démarche qui facilite l'interopérabilité des outils métier tout au long du cycle de vie du produit. Le principe du PLM consiste principalement à gérer une base de données des fichiers manipulés par différents outils métiers (CAO ou CAx), avec une gestion de WorkFlow supplémentaires (tel que : les référentiels des fichiers, des restrictions d'accès et versioning). Donc, le PLM ne fournit pas la gestion des connaissances détaillées et s'appuie sur les normes d'échange de fichiers pour réaliser l'interopérabilité.
- Les outils CAO (Conception Assistée par Ordinateur) qui gèrent les caractéristiques forme et agit comme l'un des espace de collaboration pour les concepteurs car le processus de conception est CAD-centrique. Certains outils de CAO ont été étendus au fil des ans pour embrasser l'aspect collaboratif croissant de l'ingénierie. Un parfait exemple de cette approche est le logiciel de CAO CATIA (Catia, 2011). Basé sur de bonnes pratiques techniques, CATIA propose d'intégrer des modules supplémentaires qui s'ajoutent aux fonctionnalités de bases de l'outil. Les limites d'une telle approche sont bien connues par les développeurs logiciels: le manque de modularité (intégration ad hoc), le manque de fonctionnalités (les modules sont moins puissants que des outils spécialisés), le manque d'efficacité (les ingénieurs doivent adapter leurs pratiques à l'outil).

1.2 Introduction à l'ingénierie dirigée par les modèles

L'IDM (Ingénierie dirigée par les modèles) (Bezivin, 2006 ; Favre et al., 2006) est un domaine de l'informatique mettant à disposition des outils, concepts et langages afin de créer et transformer des modèles, un modèle étant une abstraction ou un point de vue sur un système étudié. L'idée principale est de définir une architecture déclarative permettant de se concentrer sur les concepts étudiés et les liens entre ces derniers indépendamment de toute implémentation logicielle.

La figure 1 présente l'architecture standardisée par l'OMG (Object Management Group) (OMG, 2012). Le niveau M0 est le système que l'on souhaite modéliser. Le niveau M1 est un modèle représentant un point de vue sur un système. Ce modèle s'exprime à travers un langage de modélisation situé au niveau M2. La syntaxe abstraite de ce langage est définie par un méta-modèle qui structure les concepts

manipulés. Ce dernier est à son tour exprimé par un méta-méta-modèle unique et auto-descriptif situé au niveau M3 (par exemple MOF).

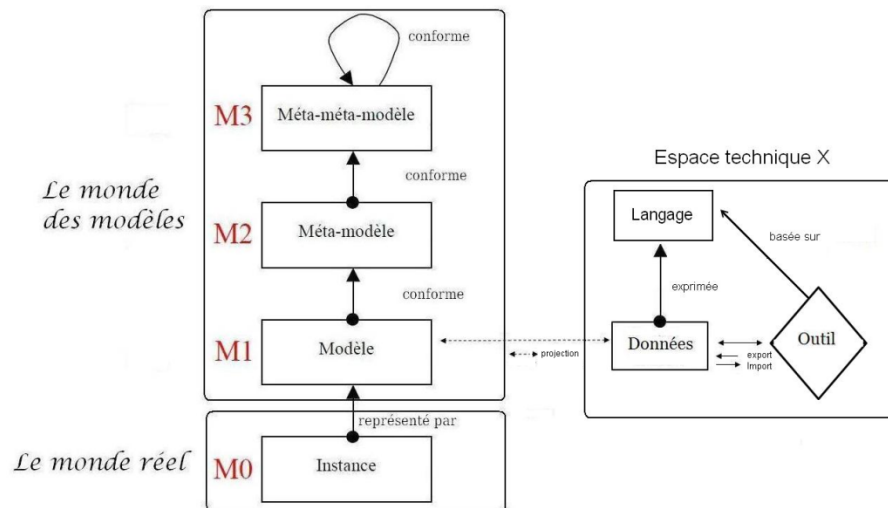


Figure 1. Représentation de l'architecture quatre niveaux et principe de la projection

Un espace technique (Bezivin, 2006) est un ensemble de techniques, de principes syntaxiques et d'outils associés à un format particulier. L'IDM propose une approche intégrant de manière homogène ces différents espaces techniques à travers des opérations (ici appelés projection) permettant d'obtenir les modèles correspondant à ces données.

Dans la démarche IDM, une transformation de modèles génère un ou plusieurs modèles cibles à partir d'un ensemble de modèles sources conformément à des règles de transformation de modèles. Ces modèles (cibles et sources) sont décrits dans un ou plusieurs méta-modèles. Atlas Transformation Language (ATL) (Brambilla et al., 2008 ; Jouault et al., 2005) est un langage de transformation de modèles semi-déclaratif qui permet de définir une transformation de modèles en se basant sur des règles déclaratives et éventuellement des fonctions auxiliaires. C'est une solution qui répond bien à la problématique de transformation de modèles en se basant sur les connaissances de collaborations et qui permet d'exprimer et d'exécuter des règles de transformation de modèles. La figure 2 représente le principe d'utilisation du langage de transformation de modèles ATL.

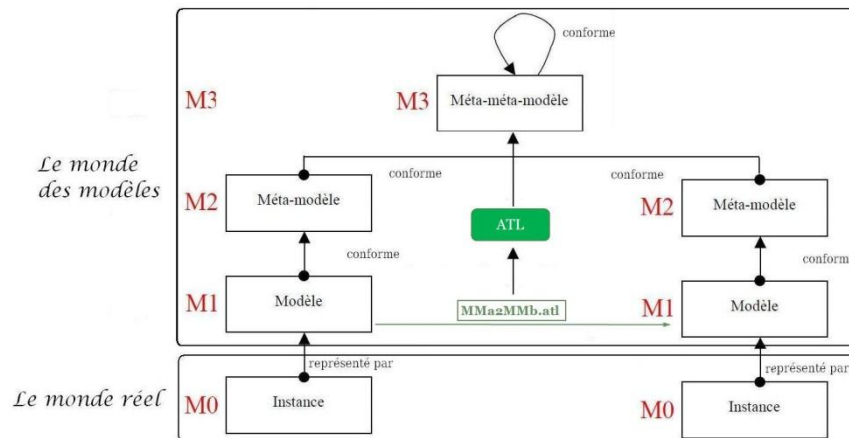


Figure 2. Principe d'utilisation du langage de transformation de modèles ATL

Une implémentation de cette architecture standard de modélisation peut être trouvée dans EMF (Eclipse Modeling Framework) (EMF, 2012 ; Budinsky et al., 2008) notamment une implémentation du MOF appelé ECORE. EMF est un cadre de modélisation et une infrastructure de génération de code pour la construction d'outils et d'autres applications basés sur des modèles.

1.3. Application de l'IDM à la conception produit

Afin de soutenir l'activité de conception de produits, le système d'information est maintenant reconnu comme une composante essentielle des pratiques d'ingénierie collaborative (Kadiri et al., 2009). Le paragraphe 1.1 présente les principales catégories d'outils informatiques courants actuellement utilisés dans l'industrie pour soutenir la modélisation de produits. Bien que ces outils aient atteint un haut niveau de fonctionnalités plusieurs problèmes restent à résoudre. Le paradigme de l'activité de conception a évolué à partir d'un processus séquentiel vers un processus simultanés. Ce nouveau paradigme a augmenté la participation de plusieurs experts métiers dans le cadre d'un même projet. Le processus de conception doit alors être centré sur les connaissances partagées des différents experts métiers. De nouveaux enjeux sont donc liés à la complexité de la gestion de connaissance.

L'interopérabilité est l'aptitude de plusieurs systèmes à communiquer, coopérer et échanger des données et services, malgré les différences dans les langages et les implémentations ou les modèles d'abstraction (Wegner, 1996). L'interopérabilité entre plusieurs outils métier peut être définie selon 3 points de vue distincts (Paviot, 2010):

- L'approche d'intégration vise à proposer un métamodèle unique qui intègre (ou rassemble) tous les concepts métiers. Un consensus doit être trouvé entre chaque expert métier, et doit être changé quand un nouveau concept est ajouté au métamodèle.

- L'approche d'unification vise à proposer un métamodèle utilisé pour lier les différents concepts métier via des associations sémantiques. Ce métamodèle doit évoluer à chaque fois qu'un nouveau concept est mis à jour (ajout, suppression, ...).

- La fédération consiste à associer plusieurs métamodèles distincts dynamiquement selon une (ou des) carte(s) de correspondance en se basant sur plusieurs concepts reliés au niveau sémantique (similarité ou équivalence). Cette approche distribuée semble être la plus souple puisque seuls des changements locaux (au différents métamodèles) doivent être traités lors de l'ajout de nouveaux concepts. L'idée principale est de se concentrer sur les concepts étudiés et les liens entre ces derniers indépendamment de toute implémentation logicielle.

L'IDM apporte une manière de représenter uniformément les connaissances portées par différents experts métiers sous forme de modèles. L'IDM favorise une approche fédérative vue la nature distribuée de ces concepts. Nous pouvons donc considérer que les données produites élaborées par un expert métier et utiles à d'autres experts peuvent être modélisées et collaborer dans un environnement IDM.

Une approche IDM peut en effet offrir de nombreux avantages, notamment:

- Un meilleur découplage métiers / technologie (une gestion des connaissances indépendante des outils métiers utilisés)

- Une maintenance simplifiée et une meilleure évolutivité (langages de modélisation déclaratifs et extensibles)

- Une meilleure gestion de la cohérence des données métiers (grâce à l'utilisation uniforme des méta-modèles et des liens qu'ils entretiennent)

Un enjeu majeur de cette approche est d'être capable de projeter les données manipulées par un outil de son espace technique vers le monde des modèles, et inversement. Cet article se propose de montrer la faisabilité de ces opérations dans le cadre de la conception de produits.

2. Projection des données métiers

Selon les outils métier, il est possible d'exporter ou d'importer les données sous plusieurs formats. Par la suite, nous considérerons indépendamment deux cas :

- XML (eXtensible Markup Language) (Michard, 2001 ; XML, 2001) un format standard d'échange de données.

- Autre format textuel, spécifique ou éventuellement conforme à un standard (ex. : STEP).

Pour les autres cas (format binaire), il est possible de se ramener à un de ces cas en implémentant manuellement l'export ou l'import des données dans un format textuel. Nous proposons donc un cadre général pour traiter ces deux cas de figures.

2.1. Format de fichier XML

XML est un langage informatique de balisage générique et extensible, il permet de décrire, structurer, stocker et échanger des données.

Les principales caractéristiques de XML sont reprises dans sa dénomination :

- eXtensible : XML est extensible et permet de créer ses propres balises en fonction des données traitées
- Markup : XML est un langage de balises (Markup Language)
- Language : XML est un métalangage (un langage pour écrire d'autres langages)

La figure 3 schématise une méthode permettant de réaliser les opérations de projection d'un fichier XML. Un fichier XML bien formé est dit valide quand il est conforme à une déclaration de type de document XSD. Une XSD (XML Schema Document) ou (W3C XML Schema) est une norme utilisée pour décrire de façon structurée le type de contenu, la syntaxe et la sémantique d'un document XML. On peut également souligner le fait que les XSD sont eux-mêmes des documents XML.

Les opérations de projection sont basées sur une carte de correspondance entre le schéma d'un fichier XML et un méta-modèle correspondant automatiquement généré grâce au Framework EMF.

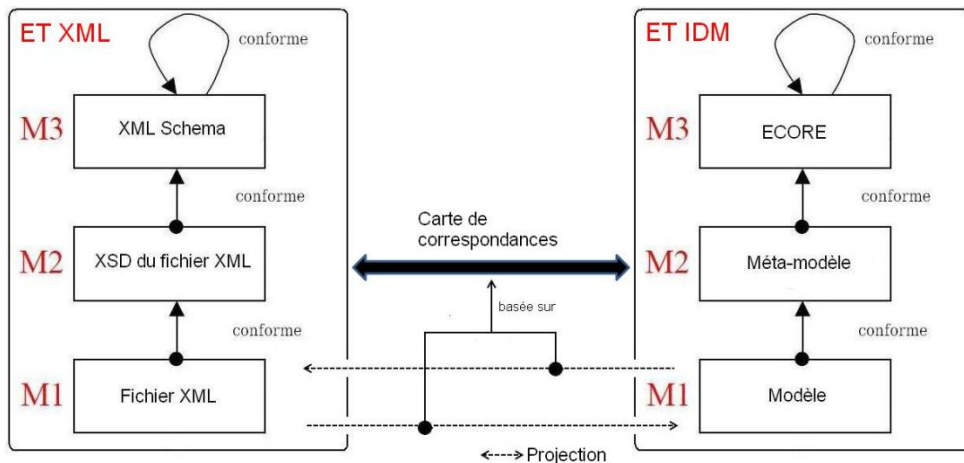


Figure 3. Mécanisme général pour la projection d'un fichier XML

2.2. Autres formats textuels (spécifiques ou standards)

Dans ce cas les données que l'on souhaite utiliser sont exprimées dans une syntaxe textuelle spécifique ou standard. Il est généralement possible de capturer cette syntaxe dans une grammaire. Une grammaire est un ensemble de règles syntaxiques exprimées dans un formalisme particulier. Notre approche est basée sur un outil et un langage de description de grammaire appelé XText (Efftinge, 2006), mais généralisable à d'autres formalismes.

La figure 4 schématise une méthode permettant de réaliser les opérations de projection en utilisant XText. XText permet non seulement de décrire la grammaire d'un langage spécifique, mais génère également à partir de cette grammaire un méta-modèle, un parseur et un éditeur textuel. De plus, on peut définir de manière déclarative des contraintes sur la grammaire qui seront alors vérifiées dans l'analyseur et dans l'éditeur.

Dans la littérature, il existe d'autres solutions pour définir des DSL et des grammaires tel que JavaCC (Java Compiler Compiler) (Copeland, 2007) ou encore ANTLR (ANOther Tool for Language Recognition) (Parr, 2007). Elles permettent à travers de la grammaire que l'on a définie pour le langage, de générer automatiquement plusieurs objets (un parseur qui s'occupe de la construction de l'arbre syntaxique abstraite, un contrôleur de type...) nécessaires pour sa mise en œuvre. Le développeur devait, dans un premier temps, manipuler ces objets pour développer son compilateur. Puis, dans un deuxième temps, vient la phase de création de l'IDE du langage où le développeur devait faire lui-même une grande partie du code à la main.

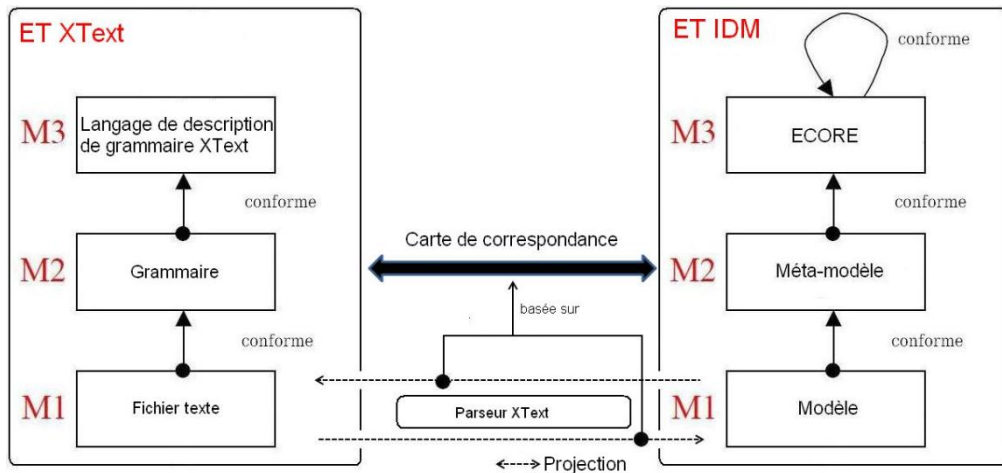


Figure 4. Utilisation de XText pour réaliser la projection

Au final, XText est un Framework intégré dans Eclipse et qui offre une simplicité dans le développement de DSL. Il surcouche ANTLR et offre un éditeur Eclipse, qui peut être enrichi par le développeur, du nouveau langage avec déjà de nombreux opérateurs par défaut tels que la complétion de code, la coloration syntaxique, ou encore l'analyse syntaxique.

3. Etude de cas

Dans cette section nous présentons deux études de cas permettant de mettre en pratique les deux techniques montrées précédemment sur des outils experts de l'industrie. Les modèles, méta-modèles et transformations de modèles réalisées sont téléchargeable (Delvion, 2011).

3.1. XML - Suite TDC

TDC système (TDC system, 2011) offre un large catalogue de logiciels et de plateformes collaboratives d'ingénierie système et de gestion de projet. Les outils TDC peuvent exporter leurs données sous format XML. Nous nous intéressons plus précisément au logiciel d'analyse fonctionnelle interne TDC Structure qui permet d'affecter les fonctions de services aux organes/composants des différentes solutions techniques imaginées par le concepteur. L'objectif de TDC Structure est d'optimiser la conception d'un produit en récupérant éventuellement en point de départ les fonctions de services définies avec le logiciel d'analyse fonctionnelle externe TDC Need.

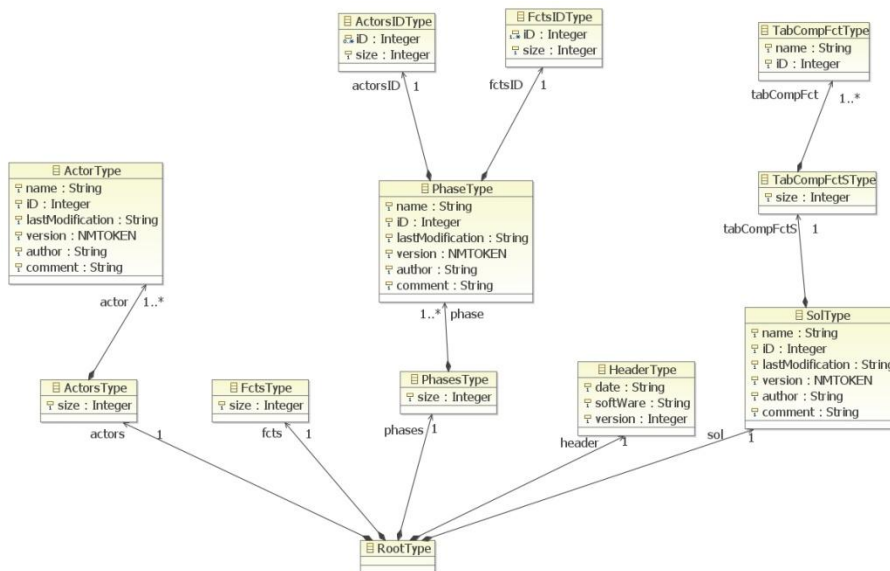


Figure 5. Extrait de la syntaxe abstraite du méta-modèle TDC sous forme de diagramme de classe

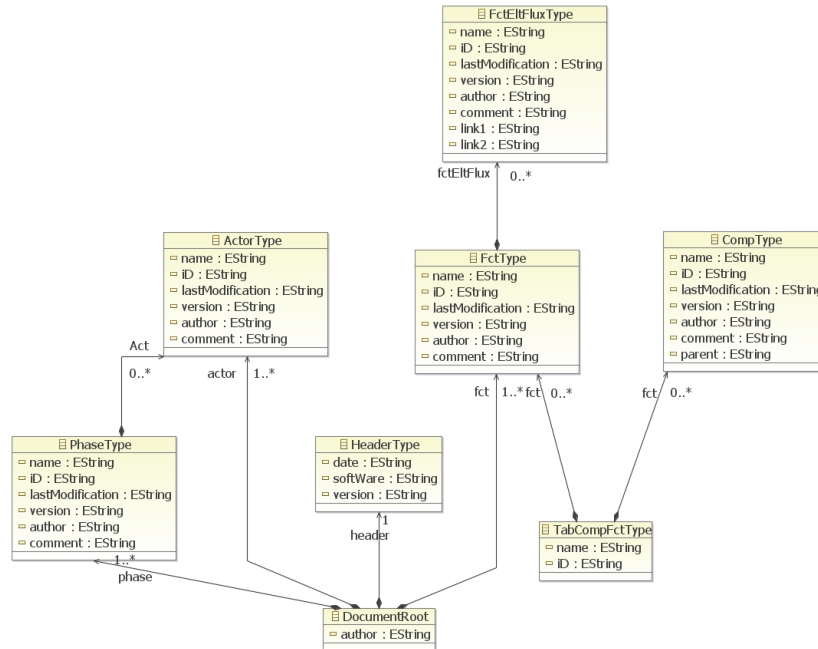


Figure 6. Extrait de la syntaxe abstraite du méta-modèle TDC optimisé sous forme de diagramme de classe

Nous avons tout d'abord défini le schéma XML des fichiers exportés par TDC Structure, car celui-ci n'était pas directement disponible. La figure 5 représente un extrait de la syntaxe abstraite du méta-modèle TDC correspondant à ce schéma et obtenu grâce à EMF.

Du point de vue de la modélisation XML schéma n'est pas aussi expressif qu'ECORE parce qu'il ne peut ni préciser le type d'une référence cible, ni définir des références bidirectionnelles (EMF_DOC, 2011). On peut alors noter sur la figure 5 que le méta-modèle contient des concepts intermédiaires superflus (exemple : PhasesType). Ces concepts sont superflus en raison de la nature arborescente de XML, les références associatives du schéma sont simulées grâce à l'équivalence des propriétés textuelles (ID). Afin d'obtenir un méta-modèle optimisé représenté dans la figure 6 (certains concepts tel que -PhasesType- dans la figure 5 sont issus de la traduction automatique du schéma XML et n'ajoutent pas de valeur sémantique supplémentaire), nous avons défini une transformation ATL (Bezivin et al., 2003 ; Jouault et al., 2005) qui s'exécute à la suite de la projection comme le montre la figure 7.

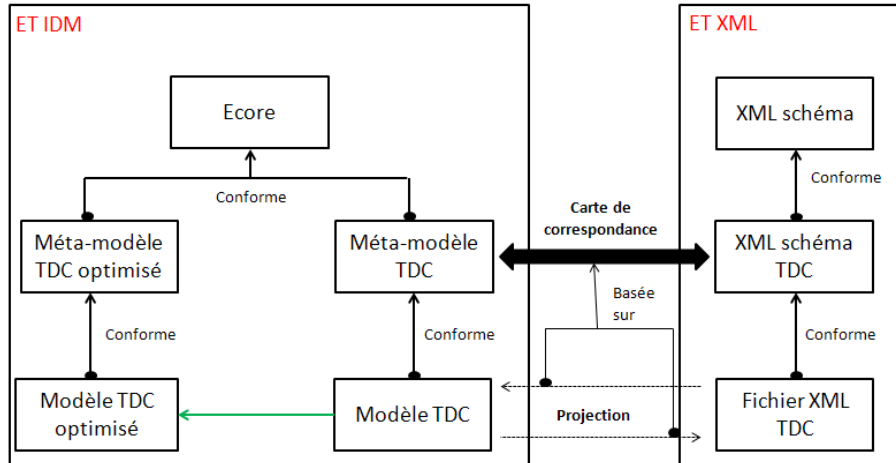


Figure 7. Mécanisme général pour la projection d'un fichier TDC

3.2. STEP - CATIA

STEP fournit un moyen neutre pour décrire les informations d'un produit dans toutes les étapes de son cycle de vie (ISO-10303, 1994). STEP a pour objectif de permettre la communication entre plusieurs outils. Il couvre de nombreux domaines et est organisé en une suite de parties (parts). Certaines de ces parties sont connues comme des Protocoles d'Application (APs) qui servent à décrire les données du produit pour une application ou un ensemble d'applications. L'OMG a déjà considéré la réalisation de l'interopérabilité entre l'espace technique de STEP et l'espace technique des modèles. Deux alternatives ont été envisagées, la première est la réalisation d'une carte de correspondance entre EXPRESS et le MOF : Nous soulignons qu'il existe des activités de recherche qui vont dans ce sens (Steel et al., 2011) : L'approche qu'ils proposent est conforme à la première alternative envisagée par l'OMG, et bien qu'il existe des travaux de recherche qui s'occupent de cette alternative, aucune n'est encore suffisamment mature. Nous avons choisi la deuxième alternative qui consiste à la réalisation d'une carte de correspondance entre un schéma STEP spécifique (donc une AP particulière) et son méta-modèle correspondant.

Nous nous intéressons dans cet article à une AP particulière : l'AP203 (Conception et contrôle de configuration) utilisée en modélisation géométrique. Le format STEP se compose de 2 parties :

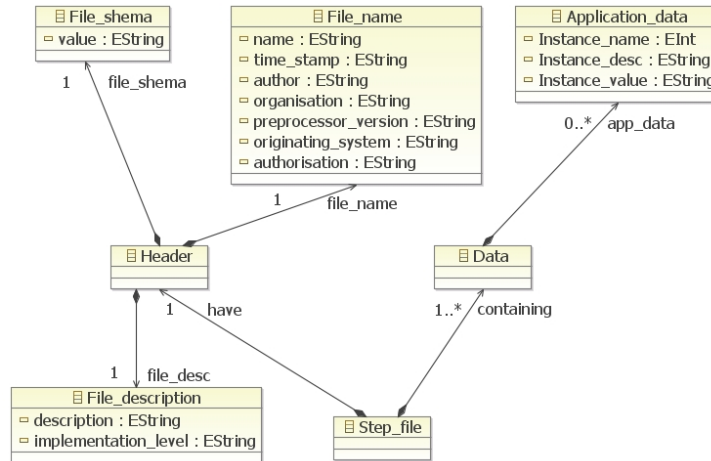


Figure 8. Syntaxe abstraite du méta-modèle du format neutre STEP sous forme de diagramme de classe

- L'entête (Header) contient entre autres la référence au schéma de données contenant la description du modèle en langage EXPRESS

- La section des données (Data) se présente sous la forme d'une succession de déclarations d'instances des classes décrites dans le schéma référencé dans la partie entête

Nous avons donc établi une grammaire de STEP grâce au langage XText qui reprend l'ensemble des règles de définition d'un fichier STEP. Depuis cette grammaire nous avons généré automatiquement le méta-modèle STEP présenté sur la figure 8. Grâce à XText il est possible de projeter les données qui se trouvent dans le modèle vers STEP. Cependant, la partie « DATA » étant regroupé dans une unique classe « Application_data » il est difficile de manipuler ces modèles STEP. Nous avons donc créé un autre méta-modèle permettant de manipuler plus finement ces données qui décrivent la description d'un produit, son arborescence et autres informations contenues dans l'AP203. La figure 9 représente un extrait de la syntaxe abstraite du méta-modèle de l'AP203.

Nous avons également opté pour cette solution qui consiste à utiliser deux méta-modèles différents pour pouvoir éventuellement manipuler les données conformes à d'autres protocoles d'application que l'AP203 vu qu'ils respectent tous la même structure du format neutre STEP représentée dans la figure 8.

La figure 10 montre l'enchaînement des transformations utilisées pour l'acquisition de données STEP-AP203 suite à une transformation ATL puis l'opération de projection vers un fichier STEP, sachant que le méta-modèle Header contient des informations relatives à la partie header d'un fichier STEP indépendamment de l'AP utilisé.

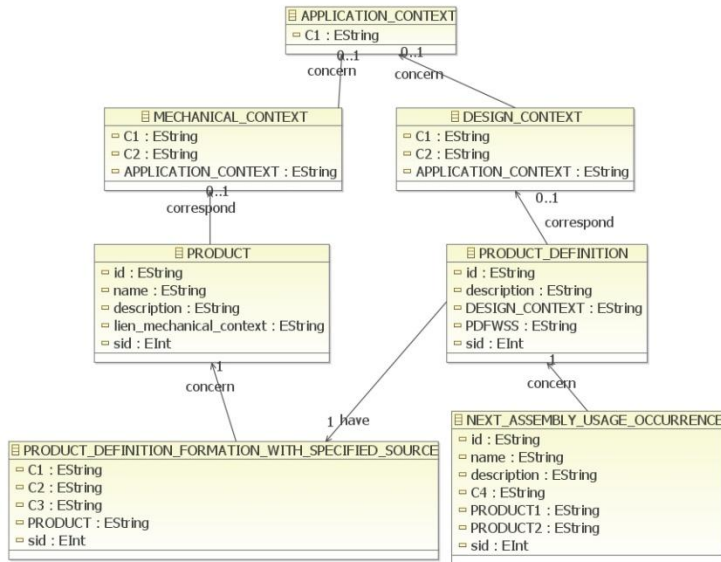


Figure 9. Extrait de la syntaxe abstraite du méta-modèle de l'AP203 sous forme de diagramme de classe

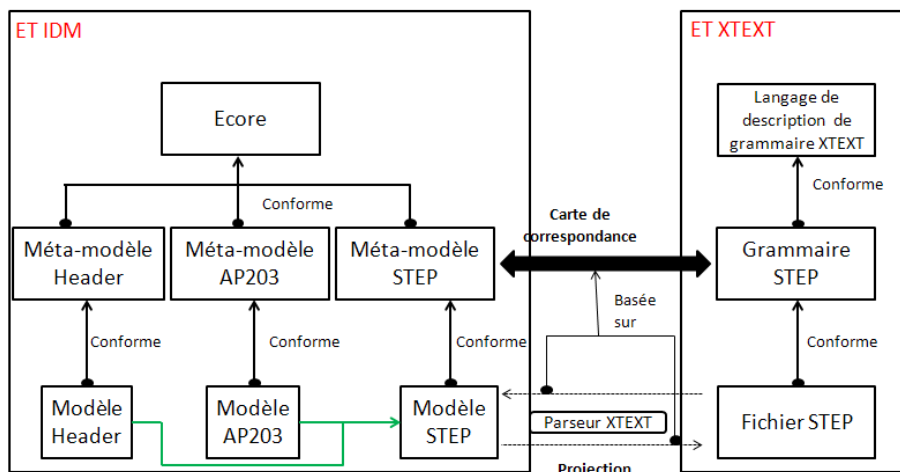


Figure 10. Mécanisme général pour la projection d'un fichier STEP

4. Travaux liés

L'approche que nous présentons dans cet article a été abordée dans le cadre de travaux de recherches connus de la littérature scientifique, notamment les travaux présentés dans (Brambilla et al., 2008) qui présentent une procédure générale d'intégration de DSL piloté par les modèles ou encore les travaux présentés dans (Bezivin et al., 2005) qui présentent leur résultats dans le cadre des projections de données et transformations de modèles en se basant sur une étude de cas basée sur deux plateformes MDE : MS/DSL et EMF.

Le travail présenté rentre dans le cadre d'une démarche fédératrice du problème d'interopérabilité des outils métiers basée sur les modèles (Iraqi-Houssaini et al., 2011): plusieurs outils métiers, basée sur un processus de conception, devraient être reliés d'une manière dynamique et flexible par la modélisation des différentes relations sémantiques et syntaxiques dans un environnement basé sur les modèles.

Cet article propose une approche pour acquérir et modéliser des données métiers pour permettre une interopérabilité fédératrice basée sur des modèles métiers. Il existe d'autres approches pour réaliser l'interopérabilité entre différents outils tel que des approches d'intégration ou d'unification (par exemple la norme STEP), et les solutions PLM tel que mentionné dans la section 1.1 et 1.3. Nous avons également souligné tout au long de l'article des travaux de recherche qui traitent les mêmes problématiques.

5. Conclusion et perspectives

Les processus de conception de produit ont dû évoluer, la maîtrise de l'information est devenue un enjeu majeur pour une collaboration efficace. Les différentes activités industrielles génèrent et manipulent de grandes quantités de données aux formats variés qui doivent être échangées et stockées de manière cohérente. A partir de ces préoccupations les experts métier ont orienté leurs travaux de recherches vers le développement de concepts, méthodes et outils informatiques qui permettent d'augmenter la qualité des produits, réduire le temps de conception et faciliter les possibilités d'innovation.

Plusieurs travaux de recherche ont été menés mettant principalement l'accent sur la mise au point de normes pour assurer la standardisation de la représentation de données (approche unifiée). Aujourd'hui, dû à la complexité des produits et des processus de conception les variations dans les échanges de données sont de plus en plus nombreuses, il faut donc être plus flexible. C'est dans ce sens que l'approche fédératrice peut apporter une bonne solution qui permet de faire évoluer le système d'information. Nous supposons que l'approche fédératrice pourrait être mixée avec l'approche unifiée afin de couvrir au mieux les zones processus-produit à changements fréquents d'une part (approche fédérée) et à changement plus lents (approche unifiée).

Dans cet article, nous avons proposé un cadre général pour acquérir et modéliser des données métier manipulées par différents outils d'ingénierie produit. Notre approche est expérimentée par deux cas d'études différents issus du monde industriel. Ces travaux s'inscrivent dans une démarche d'interopérabilité fédérative basée sur l'ingénierie des modèles et dont l'objectif est de faciliter la conception collaborative entre différents experts métier.

Ce travail suit naturellement les travaux précédents sur l'adaptation des techniques de modélisation au contexte de la conception des produits (Iraqi-Houssaini et al., 2011), en fournissant des techniques pour réaliser l'interopérabilité des outils métiers entre les différents espaces techniques. Les futures investigations porteront sur l'un des aspects les plus prometteurs de cette approche fédératrice: la possibilité de créer dynamiquement et automatiser le système d'information (logiciels) selon le procédé de conception de produits.

References

- Bezivin J. (2006). *Model Driven Engineering: An Emerging Technical Space*, Lecture Notes in Computer Science, Volume 4143/2006.
- Bézivin J., Hillairet G., Jouault F., Kurtev I., Piers W., (2005). *Bridging the MS/DSL Tools and the Eclipse Modeling Framework*, In Proceedings of the International Workshop on Software Factories at OOPSLA.
- Bezivin J., Dupe G., Jouault F., Pitette G., Rougui J., (2003). *First experiments with the ATL model transformation language: Transforming XSLT into XQuery*, 2nd OOPSLA Workshop on Generative Techniques in the context of Model Driven Architecture.
- Brambilla M., Fraternali P., Tisi M., (2008). *A Transformation Framework to Bridge Domain Specific Languages to MDA*, Models in Software engineering - selection from MoDELS Workshops 2008, Springer LNCS, vol. 5421, 2008, pp. 167-180.
- Budinsky F., Steinberg D., Merks E., Paternostro M., (2008). *Eclipse modeling framework: a developer's guide*, Series Editors: Erich Gamma - Lee Nackman – John Wiegand.
- CATIA (Dassault systems) (2011). <http://www.3ds.com/products/catia/welcome/>.
- Copeland T. (2007). *Generating Parsers with JavaCC: An Easy to Use Guide for Programmers*, Centennial Books.
- Debaeckerd D. (2004). *PLM, la gestion collaborative du cycle de vie des produits*, Product Life-Cycle Management, Hermès – Lavoisier.

Delvion usecase (2011). http://www.lsis.org/kleiner/MPD/Delvion_usecase.html.

Efftinge S. (2006). *oAW xText: A framework for textual DSLs*, Eclipse Summit 2006.

EMF (2012). Eclipse Modeling Framework, <http://www.eclipse.org/emf/>

EMF_DOC (2011). Generating an EMF 1.1 Model using XML Schema, http://www.eclipse.org/modeling/emf/docs/1.x/tutorials/xlibmod/xlibmod_emf1.1.html.

Favre J.M., Estublier J., Blay M., (2006). *L'ingénierie dirigée par les modèles au-delà du MDA*, Lavoisier.

Iraqi-Houssaini M., Kleiner M., Roucoules L., (2011). *Model-based (Mechanical) Product Design*, Wellington, New Zealand, MODELS 2011.

ISO 10303:1994, Industrial automation systems and integration – Product data representation and exchange is the formal name for the international standard familiarly known as STEP.

ISO 10303-21:1994, Industrial automation systems and integration – Product data representation and exchange – Part 21: Implementation methods: Clear text encoding of the exchange structure.

Jouault F., Kurtev I., (2005). *Transforming Models with ATL*, Proceedings of the Model Transformations in Practice Workshop at MoDELS 2005, Montego Bay, Jamaica.

Kadiri S.E., Pernelle P., Delattre M., Bouras A., (2009). *Current situation of plm systems in sme/smi: Survey's results and analysis*, International Conference on Product Lifecycle Management.

Michard A. (2001). « *XML, langage et applications* », Eyrolles, Paris, France.

Noel F., Roucoules L., Teissandier D., (2004). *Specification of product modelling concepts dedicated to information sharing in a collaborative design context*, 5th International Conference on Integrated Design and Manufacturing in Mechanical Engineering, IDMME 2004, University of Bath, Bath, United Kingdom.

OMG (2012). Meta Object Facility (MOF) Specification, <http://www.omg.org/mof/>.

Parr T. (2007). *The Definitive ANTLR Reference: Building Domain-Specific Languages*, Pragmatic.

Paviot T. (2010). *Méthodologie de resolution des problèmes d'interopérabilité dans le domaine du Product Lifecycle Management*, Thèse de Doctorat de l'École Centrale Paris.

Steel J., Duddy K., Drogemiller R., (2011). *A Transformation Workbench for Building Information Models*, Zurich, Switzerland, ICTM 2011.

TDC system (2011). <http://www.tdc.fr/>.

Wegner P. (1996). *Interoperability*, ACM Computing Survey, pages 258_287.

XML (2001), eXtensible Markup Language,
<http://www.omg.org/spec/XML/1.1/PDF>.