Apport de l'approche MDA pour une interopérabilité sémantique : Interopérabilité des systèmes d'information d'entreprise

Salah Baïna, Hervé Panetto, Khalid Benali

▶ **To cite this version:**

Salah Baïna, Hervé Panetto, Khalid Benali. Apport de l'approche MDA pour une interopérabilité sémantique : Interopérabilité des systèmes d'information d'entreprise. Revue des Sciences et Technologies de l'Information - Série ISI : Ingénierie des Systèmes d'Information, 2006, 11/13, pp.11-29. hal-00086522

HAL Id: hal-00086522
https://hal.science/hal-00086522

Submitted on 22 Dec 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Apport de l'approche MDA pour l'interopérabilité des systèmes d'entreprise

Salah Baïna[1], Hervé Panetto[1], Khalid Benali[2]

[1] CRAN (UMR 7039), Université Henri Poincaré Nancy I, F 54506 Vandoeuvre-les-Nancy, France.
{salah.baina, herve.panetto} @cran.uhp-nancy.fr

[2] LORIA (UMR 7503), Campus scientifique,F 54506 Vandoeuvre-les-Nancy, France,
benali@loria.fr

RÉSUMÉ. *Depuis quelques années, l'intérêt autour de l'interopérabilité n'a cessé d'augmenter. Ceci montre le besoin de plus en plus omniprésent de développer des applications et des systèmes complètement intégrés. En effet l'interopérabilité des applications permet d'assurer l'échange de données et de modèles entre applications de manière complètement transparente à l'utilisateur pour qui le système devient une seule entité fortement intégrée. Cet article présente une approche pour l'interopérabilité basée sur une architecture orientée modèle (MDA). L'approche MDA (Model-driven architecture) fournit les bases pour l'utilisation de modèles afin de guider la compréhension, la conception, la construction, le déploiement, la maintenance et la modification des systèmes, au sens général. L'association de l'approche MDA avec le domaine de l'interopérabilité permet de proposer une méthode structurée pour l'établissement de l'interopérabilité entre modèles applications.*

ABSTRACT. *In the last few years, lot of work has been done in order to ensure interoperability between applications. Enterprise applications integration and the opening of information systems towards integrated access have been the main motivation for the interest around systems interoperability. In this Paper, to handle the interoperability problem at modelling phase, an MDA based approach for interoperability is proposed. The MDA is an approach for systems development, which increases the power of models in that work. It is model-driven because it provides a means for using models for better understanding, designing, constructing, deploying, operating, maintaining and modifying systems. The three primary goals of MDA are portability, interoperability and reusability. In this paper we apply the MDA as a formal base that supports our approach for interoperability in the enterprise context.*

MOTS-CLÉS : *Interopérabilité, Approche MDA, Meta-modélisation, Ontologies, Mapping de modèles, Synchronisation de modèles, Modèle de référence.*

KEYWORDS: Interoperability, MDA approach, Meta-modelling, Ontologies, Models mapping, Models Synchronisation, Reference model.

1. Introduction

Depuis quelques années, l'interopérabilité des applications est devenue le leitmotiv des développeurs et concepteurs de systèmes. Cette importance a donné lieu à d'innombrables travaux de recherche avec pour chacun une définition particulière plus ou moins formelle de l'interopérabilité des applications. Les définitions les moins formelles donnent lieu à des classes d'interopérabilité difficilement prouvables et encore moins quantifiables. L'objectif principal de notre approche est de proposer une définition formelle de l'interopérabilité basée sur l'architecture MDA. Cette approche offre un mécanisme efficace pour l'établissement de l'interopérabilité entre applications. Dans la suite de l'article, par « interopérabilité » nous entendons « la capacité à communiquer, coopérer et échanger des modèles ou des données entre deux ou plusieurs applications malgré les différences dans les langages d'implémentation, les environnements d'exécution ou les modèles d'abstraction » (Kalfoglou et Schorlemmer, 2004).

La plupart des approches pour l'interopérabilité existant dans l'environnement de l'entreprise ont pour objectif principal l'ajustement et l'adaptation des types et structures des paramètres d'appel lorsqu'une procédure doit en invoquer une autre. Ces approches se basent sur le paradigme d'appel de procédures ; elles traitent l'interopérabilité au moment de l'exécution ou l'appel de procédure. Ce type d'interopérabilité est appelé l'interopérabilité orientée procédure. Dans ce cadre, il est admis que la procédure requise par le client et celle offerte par le serveur soient fonctionnellement identiques, autrement dit les fonctions réalisées par la procédure requise et celles réalisées par la procédure offerte sont les mêmes ou tout au moins compatibles. Bien que l'interopérabilité orientée procédure soit une bonne base pour l'interopérabilité entre composants ou applications hétérogènes en se basant sur l'unification des interfaces, les solutions offertes résolvent uniquement le côté syntaxique de l'interopérabilité (types des entrées → types des sorties). Le protocole d'interconnexion des interfaces ne traite ni la sémantique des fonctions réalisées par les procédures ni la sémantique des données en entrée ou en sortie. Les travaux relatifs aux services web apportent un peu plus de sémantique par la définition des services disponibles de manière distribuée mais restent, tout de même, très technologiques. Pour traiter l'interopérabilité sémantique des données, plusieurs travaux de recherche ont vu le jour, on peut les classifier en trois grandes approches : l'interopérabilité par mapping, l'interopérabilité via intermédiaires et l'interopérabilité par échange de requêtes (Park et Ram, 2004).

L'approche par mapping (mapping-based approach) a pour but de construire des mappings entre données sémantiquement reliées. Cette approche se base, en général, sur un schéma global (ou fédérateur) et un ensemble de mappings reliant le schéma fédéré à chacun des schémas locaux (Rahm E, Bernstein, 2001 ; Madhavan et al, 2002 ; Halevy et Madhavan, 2003). Cependant, dans cette approche, le schéma fédéré dépend fortement des schémas locaux et spécifiques à chaque système. Une représentation explicite de la sémantique des données source pourrait aider à réaliser les mappings. Mais cette représentation pose le problème de la définition d'une

sémantique, problème scientifique encore ouvert et pour lequel il n'existe pas encore de solutions viables à une échelle réelle.

L'approche via intermédiaires se base sur l'utilisation de mécanismes intermédiaires (ex : médiateurs, agents, ontologies, etc.) dans le but de mettre en place l'interopérabilité (Adali et al, 1996 ; Garcia-Molina et al, 1997 ; Papastavrou et al, 1999 ; Maedche et Staab, 2000 ; Halevy et al, 2005). Ces intermédiaires doivent avoir une connaissance spécifique du domaine, ou des heuristiques précises pour coordonner différentes sources de données ou peuvent contenir eux-mêmes des règles de construction de mapping. Dans la majorité des cas, ces intermédiaires utilisent des ontologies pour partager un vocabulaire ou un protocole de communication commun. Le recours aux ontologies permet d'avoir une représentation relative au domaine étudié mais indépendante des schémas individuels utilisés par les systèmes et les applications. L'approche par les ontologies semble prometteuse mais se heurte à la problématique de définition d'ontologies communes nécessitant des consensus difficile à atteindre.

La troisième et dernière approche est l'approche par échange de requêtes basée sur des langages interopérables ; des langages de logique déclarative ou bien des langages dérivés de SQL (Miller et al, 2000; Chamberlin, 2000; Fauvet et Baïna, 2003). Ces langages permettent de formuler des requêtes interrogeant plusieurs bases de données. Pour résoudre des conflits sémantiques, ces langages doivent prendre en compte les deux aspects des bases de données: Données et Méta Données. L'un des principaux inconvénients de cette approche est le fait de déléguer à l'utilisateur la résolution des concordances entre les différentes bases de données. Ce qui implique la responsabilité de l'utilisateur lui même dans le processus de résolution des conflits sémantiques. Cette approche est donc difficilement automatisable.

Dans cet article nous présentons une approche se basant sur une architecture orientée modèles pour définir des mappings entre modèles d'applications afin de faciliter l'échange d'information entre applications. Dans la section 2, nous commençons par présenter brièvement l'architecture orientée modèles (Model-Driven Architecture : MDA). Ensuite, nous montrons comment les mappings entres meta-modèles d'applications peuvent être utilisés pour établir une définition de l'interopérabilité sémantique. La section 3 présente une application de notre approche dans le cas spécifique de l'interopérabilité entre applications d'entreprises manufacturières. Dans la section 4, une implémentation du cas d'application présenté dans la section 3 est proposée. Finalement, la section 5 conclut en récapitulant les points importants de l'article et les perspectives de travaux de recherche futurs.

2. Approche pour l'interopérabilité dans le cadre d'une architecture orientée modèles (MDA)

Dans cette section, nous introduisons une approche pour l'interopérabilité autour de l'architecture orientée modèles « Model-Driven Architecture » (MDA) (Bezivin, 2001; Mellor, et al., 2004) définie par l'OMG (Object Management Group[1]). L'objectif principal de cette section est d'expliquer l'apport de l'approche MDA dans le contexte de l'interopérabilité. L'approche MDA est utilisée pour le développement de systèmes. Elle est orientée modèles parce qu'elle fournit les bases pour l'utilisation des modèles afin de guider la compréhension, la conception, la construction, le déploiement, la maintenance et la modification des systèmes développés. Les trois principaux objectifs du MDA sont la portabilité, l'interopérabilité et la réutilisabilité à travers la séparation des aspects dépendants de la plateforme ou de l'application étudiée et des aspects plus abstraits indépendants de l'application. L'approche MDA est basée sur quatre niveaux ontologiques (Naumenko et Wegmann, 2003) (*cf.* figure 1).

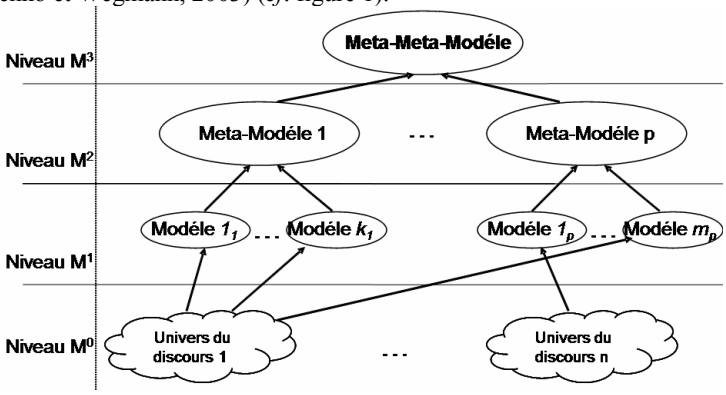

Fig. 1 *L'approche MDA et les quatres niveaux ontologiques.*

Le niveau le plus bas M^0 représente les différents objets de modélisation aussi appelés « univers du discours ». Le niveau M^1 spécifie les différents modèles de chaque univers de discours. Le modèle d'un système est sa description ou sa spécification. Un modèle est souvent présenté comme une combinaison de dessins et de textes. Le texte peut être dans un langage de modélisation spécifique ou bien en langage naturel. Les modèles du niveau M^1 appartiennent à différents domaines d'intérêt relatifs aux univers du discours représentés par les modèles. Il est possible qu'un domaine d'intérêt recouvre plusieurs univers du discours. De même, des modèles provenant de différents univers de discours peuvent relever du même domaine d'intérêt. Le niveau M^2 représente les meta-modèles spécifiques à chaque

1. OMG , http://www.omg.org

domaine : un meta-modèle pour chaque domaine d'intérêt pertinent pour les modèles de niveau M^1. Finalement, le niveau M^3 présente le meta-meta-modèle et doit être conçu pour permettre la définition de tous les concepts nécessaires pour la modélisation de meta-modèles et pour leur unification dans un cadre de référence commun. Un meta-meta-modèle est indépendant du domaine, il contient des meta-caractéristiques pour des meta-modèles spécifiques.

L'approche MDA offre un support formel pour traiter le problème de l'interopérabilité des applications. Pour faire face au problème d'interopérabilité dans le contexte MDA, nous utilisons le fait que chaque application (ou système) possède un modèle représentant un univers du discours (ou une partie de cet univers). Chaque modèle étant lui-même construit sur les bases d'un meta-modèle. Etant donné deux applications A et B; nous dirons que A et B sont sémantiquement interopérables si et seulement si nous pouvons construire une instance du modèle de B à partir d'une instanciation du modèle de A et construire une instance du modèle de A à partir d'une instanciation du modèle de B. Etablir l'interopérabilité entre applications revient donc à définir les transformations possibles entre les modèles se trouvant à la base de ces applications. Lemesle (1998) introduit la transformation de modèles en établissant des règles de transformations entre meta-modèles (*cf*. figure 2).

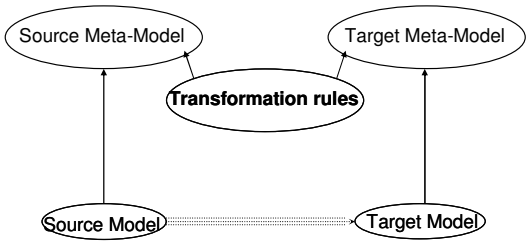

Fig. 2 : *Transformation de meta-modèle (Lemesle, 1998)*

On appelle donc mapping, l'ensemble des règles de transformation de modèles permettant de traduire les instances d'un meta-modèle source en des instances d'un meta-modèle cible (Levendovszky *et al,* 2002).

Pour les besoins de formalisation, nous définissons formellement les notions de meta-modèles et de mappings. Pour cela, nous adoptons une approche algébrique basée sur la théorie algébrique pour les mappings d'ontologies définit par (Kalfoglou and Schorlemmer, 2003). Une ontologie est formellement définie comme un couple $O = (S, A)$, où S est la signature *(ontologique)* – décrivant un vocabulaire – et A est un ensemble d'*axiomes (ontologiques)* – spécifiant l'interprétation du vocabulaire dans l'univers du discours. Typiquement, une signature est munie d'une structure mathématique ; cette structure est généralement exprimée à l'aide d'une hiérarchie de concepts ou de symboles définissant un ensemble partiellement ordonné (Kalfoglou et Schorlemmer, 2003). Par analogie, nous définissons de la même

manière un meta-modèle ou un modèle comme étant un ensemble de concepts issus d'une ontologie, et un ensemble de relations ou associations entre ces concepts. Les relations du modèle (meta-modèle) peuvent être basées sur les axiomes définis dans l'ontologie. Nous définissons, donc, un meta-modèle (ou modèle) comme un couple *M = (C, R)* où *C* est l'ensemble des concepts utilisés dans M, et *R* l'ensemble des relations entre ces concepts. Étant donné que les concepts utilisés par le meta-modèle *M* sont issus d'une Ontologie *O* = (*A,S*), la structure sémantique définie sur *A* se retrouve donc projetée sur *C*. On en déduit que *C* est muni d'une relation d'ordre partiel. Un mapping entre meta-modèles est ainsi défini comme étant une fonction qui relie entre eux les concepts utilisés des meta-modèles, cette fonction doit permettre aussi de préserver la structure sémantique entre les concepts dans chacun des vocabulaires. Plus formellement, soit M = (C, R) et M' = (C', R') deux meta-modèles ; on appelle mapping total de M vers M,' une application *f* de *C* vers *C'*, préservant l'ordre partiel, qu'on notera '≤', définie par la structure sémantique de *C* et *C'* (*voir axiomes 1 et 2*).

$$\forall\, a \in C,\ \exists\, y \in C' \qquad \textit{tel que } f(a) = y \qquad [1]$$

$$\forall\, a \in C,\ b \in C\ ;\ a \leq b \quad \Rightarrow \quad f(a) \leq f(b) \qquad [2]$$

Un mapping est donc un morphisme entre des vocabulaires dont le but est de conserver la sémantique des termes transcrits. Généralement, il est très difficile de faire en sorte que tous les concepts définis par un meta-modèle soient transcrits en concepts appartenant à un autre meta-modèle. Pour s'adapter à des mappings plus faibles et moins couvrants, nous définissons la notion de mapping partiel entre meta-modèles. Un mapping entre deux meta-modèles M1 = (C1, R1) et M2 = (C2, R2) est dit partiel s'il se réduit à un mapping total entre une sous partie du meta-modèle de départ M1 vers une sous partie meta-modèle d'arrivée M2. Maintenant, nous établissons la relation entre cette définition formelle des mappings et l'interopérabilité dite sémantique entre applications.

***Définition**:* Considérons *A* et *B* deux applications; *A* et *B* sont dites sémantiquement interopérables si et seulement si il existe un mapping bijectif *de* M_A vers M_B, que nous noterons *f*. La bijection de *f* assure que l'on peut construire une instance du modèle de *B* à partir d'une instanciation du modèle de *A* (en utilisant *f)* et construire une instance du modèle de *A* à partir d'une instanciation du modèle de *B* (en utilisant le mapping inverse f^{-1}*)*.

Nous pouvons donc définir trois niveaux d'interopérabilité en se basant sur les mappings identifiés :

Niveau 2 : Il existe un isomorphisme total entre M_A et M_B*:*

> Dans ce cas de figure, tout concept apparaissant dans M_A a son équivalent dans M_B et vice versa, ce qui signifie que M_A et M_B sont équivalents.

Niveau 1 : il existe un isomorphisme partiel entre M_A et M_B.

> Il existe une sous partie de M_A qu'on notera M'_A *et* une sous partie de M_B (M'_B), telles que l'interopérabilité entre M'_A et M'_B est de niveau 2 (M'_A et M'_B sont équivalentes)

Niveau 0 : Il n'existe pas d'isomorphisme partiel entre M_A et M_B

> Cependant, il se peut que des mappings non bijectifs existent entre M_A et M_B, mais dans ce cas, on ne peut pas parler d'interopérabilité sémantique entre A et B.

L'interopérabilité sémantique de niveau 2 est la plus difficile à établir étant donné qu'il est très rare que deux meta-modèles M_A et M_B soient totalement équivalents. En général, on cherche à établir une interopérabilité partielle entre les deux meta-modèles, qui peut être quantifiée en calculant la proportion de M'_A et M'_B par rapport à M_A et M_B.

Pour illustrer l'utilisation de l'approche MDA pour l'interopérabilité, nous introduisons, dans la suite, un exemple où nous nous basons sur l'approche MDA pour la définition de mappings dans le but d'établir l'interopérabilité entre modèles d'entreprise spécifiques.

3. Cas d'application

L'interopérabilité des applications et des systèmes d'entreprise est au centre des préoccupations des responsables des systèmes d'information au sein des entreprises. Dans cette étude nous nous intéressons au cas de l'interopérabilité des systèmes d'entreprise dans le contexte des entreprises manufacturières et plus précisément l'interopérabilité B2M (Business to Manufacturing). Dans ce cadre, le problème d'interopérabilité entre entreprises est un résultat direct qui découle du besoin d'échanges et de coopérations entre entreprises et au sein même d'une entreprise (*cf* figure 3).

3.1. l'interopérabilité entre modèles d'entreprise dans le contexte manufacturier.

Comme le montre la figure 3, les flux échangés entre systèmes d'entreprise, que ce soit au sein d'une même entreprise (intra-entreprise), ou entre différentes entreprises (inter-entreprises), peuvent être de nature différente, physique ou informationnelle. Cependant, les échanges informationnels ne sont pas complètement indépendants des échanges de matières et vice versa ; en général, la liaison information/matière est implicite mais n'est pas maintenue concrètement. Cette liaison permet entre autre de maintenir la cohérence entre la réalité physique des flux de matière (produits, matières premières ou ressources) et le contenu informationnel véhiculé dans les flux d'informations échangés par les différents systèmes d'informations.

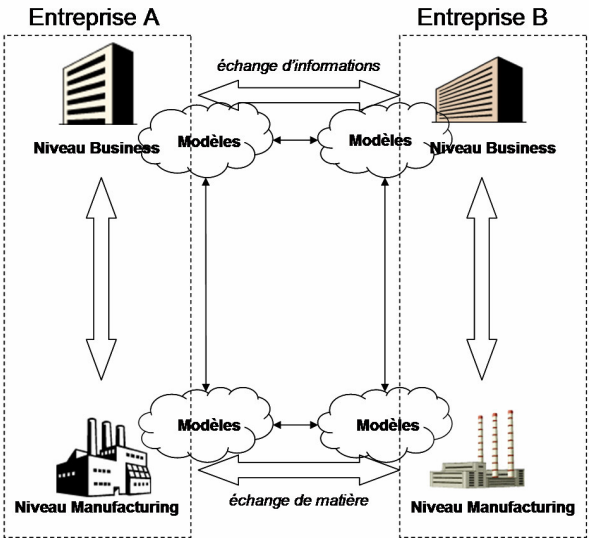

Fig. 3 *Les différents échanges dans le cadre des entreprises manufacturières.*

Pour expliciter le lien entre information et matière lors des différents échanges dans une entreprise, nous introduisons la notion d'*holon* (Koestler, 1967), définie comme étant : un élément identifiable autonome, coopérant avec son environnement faisant partie d'un système de transformation, d'échange et de validation d'informations ou d'objet physique (Seidel et Mey, 1994 ; Wyns, 1999).

Un *holon* est donc vu comme étant l'agrégation :

- d'une partie physique représentant la partie matérielle du produit, matière première ou ressource.
- et d'une partie informationnelle contenant toutes les données et informations relatives à la partie physique, sa structure et son interaction avec son environnement.

Dans certains cas de figure, il se peut qu'un holon ne possède pas de partie informationnelle ou de partie physique, cependant, il doit posséder au moins l'une des deux.

Les holons, parties physiques et parties informationnelles, peuvent être véhiculés à travers des flux. Le type d'un flux étant contraint par la nature de ce qu'il contient, nous identifions donc trois types de flux:

- des *flux d'holons* contenant des holons.
- des *flux physiques* contenant des parties physiques

- des *flux informationnels* transportant des informations.

L'holon représente donc l'unité identifiable permettant de faire le lien entre éléments informationnels véhiculés dans les flux informationnels et les éléments matériels véhiculés dans des flux de matière. Le concept d'holon permet aussi de faire abstraction du type des objets véhiculés (produits, ressources ou matière première) car le type dépend de l'usage et non de la structure ou la définition des objets ; un produit provenant d'une entreprise A peut être vu comme matière première pour l'entreprise B et comme ressource pour l'entreprise C, alors que l'objet lui-même ne change pas entre une utilisation et une autre. L'objectif est donc de structurer les échanges entres les applications ou entre modèles dans le contexte manufacturier en éléments holoniques. (*cf*. figure 4).

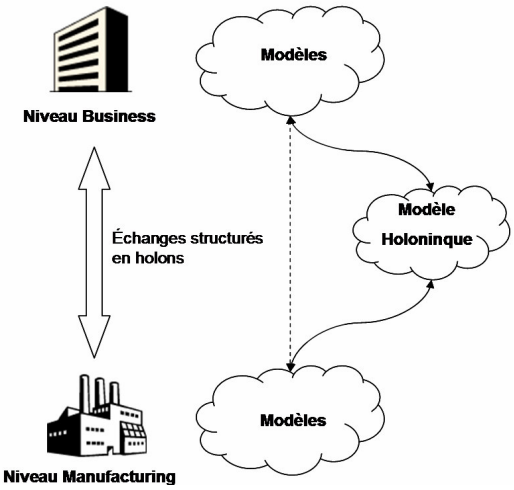

Fig. 4: *Holon comme passerelle entre modèles d'applications.*

Pour mettre en œuvre cette approche, il faut donc que les modèles des applications de niveau gestion (business) (resp. production (manufacturing)) soient interopérables avec nos modèles holoniques, au sens de la définition présentée dans la section 2.

Pour illustrer l'utilisation de la modélisation basée sur le concept d'holon dans l'interopérabilité entre systèmes d'entreprises, nous proposons un modèle du produit basé sur l'approche holonique et par la suite nous montrons comment ce modèle holonique peut être utilisé pour les échanges autour du produit entre modèles d'entreprise. L'objectif étant d'utiliser l'approche MDA pour établir des mappings entre le modèle holonique du produit et les différents modèles rencontrés dans l'entreprise.

3.2. l'holon pour la modélisation du produit.

Dans le contexte manufacturier, l'importance du produit en tant qu'entité identifiable, interagissant avec son environnement, jouant un rôle actif dans la prise de décision, ne cesse d'augmenter. Selon leurs besoins et leurs interactions avec le produit, la plupart des applications de l'entreprise possèdent une vue même très réduite du produit. Le besoin de maintenir la coordination et la synchronisation entre ses vues avec la réalité physique du produit est devenu une nécessité évidente. Il est donc primordial de fournir une méthode permettant l'échange d'informations entre les différentes applications afin de maintenir la cohérence entre les vues du produit manipulées par les différents composants du système d'information de l'entreprise, et de garantir leur correspondance avec l'état réel du produit tout au long de son cycle de vie (conception, fabrication, commercialisation, utilisation…).

Nous utilisons donc le concept d'holon définit précédemment pour la représentation unifiée du produit lors des échanges entre applications. La partie physique du produit (ou vue physique) est représentée par la partie physique de l'holon et les informations relatives au produit (vue informationnelle) sont représentées par la partie informationnelle de l'holon. Les informations relatives au produit peuvent êtres classées en deux catégories:

- Des informations intrinsèques de l'objet physique lui-même, qu'on appellera des *Attributs,* ces attributs décrivent des informations substantielles relatives à la partie physique même de l'holon (ex : poids, couleur, forme, matériaux…)
- Des informations attachées au produit durant les différentes étapes de modélisation ; chacune de ces informations étant relative à un certain domaine de modélisation. Ces informations seront appelées *Propriétés* (ex : identifiant, date de fabrication, date de livraison…)

L'holon représentant le produit peut être classifié en deux types d'holons : *(i)* les holons élémentaires et *(ii)* les holons complexes :

- Les holons élémentaires sont la combinaison d'une partie physique unique et de la partie informationnelle lui correspondant ;
- Les holons complexes quant à eux sont le résultat d'une composition ou une décomposition d'un ou plusieurs holons (élémentaires ou complexes).

La figure 5 représente la spécification en UML du meta-modèle holonique qui sert de base la modélisation de produits.

La Classe *Holon* définit la structure de base d'un holon, qu'il soit élémentaire ou complexe. Un *Holon Complexe* est, contrairement à un *holon Elémentaire*, un holon ayant subit au moins un processus de composition ou de décomposition lors de sa fabrication. La classe *Etat* représente l'ensemble des états par lesquels passe un *holon*. L'historique d'un *holon* est composé de l'ensemble de ces états permettant d'assurer la traçabilité des produits au long de leur cycle de vie. L'état d'un holon est définit par un ensemble de couples (attribut, valeur) et (propriété, valeur). Une *Partie physique* est une référence à la partie matérielle associée à un *holon*. Une

Instance de Processus fait référence à l'exécution d'un Processus sur une instance d'*holon*. Un *Processus* décrit un traitement (physique ou informationel) effectué sur un ensemble de produits dans l'entreprise.

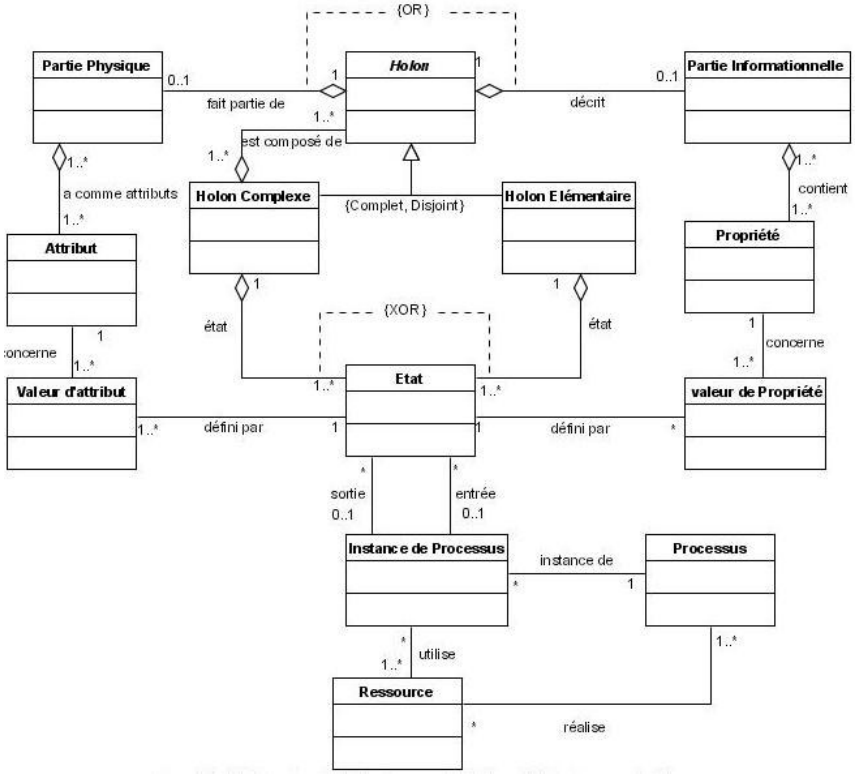

Fig. 5: *Meta-modèle holonique pour la modélisation du produit*

Dans la section qui suit, nous tenterons de combiner l'utilisation de l'approche holonique et l'approche MDA dans le but de montrer comment le meta-modèle holonique pour le produit peut être utilisé pour assurer l'interopérabilité orientée produit des applications d'entreprise qui manipulent des modèles de données relatifs au produit.

3.3. MDA et Holon pour l'interopérabilité orientée produit

Pour illustrer l'application de l'interopérabilité orientée produit dans un environnement manufacturier, nous nous intéressons aux différents types d'applications coexistants dans un tel environnement. La figure 6 présente notre vision de l'entreprise; deux univers séparés l'univers Business et l'univers Manufacturing. Le premier regroupe l'ensemble des applications qui traitent plutôt les dimensions stratégiques, économiques de l'entreprise, le deuxième regroupe l'ensemble des infrastructures nécessaires au bon déroulement de la production au niveau de l'atelier. Cette séparation se justifie par la différence des objectifs et des préoccupations dans ces deux univers. A ces deux aspects s'ajoute souvent la séparation géographique (ateliers et bureaux).

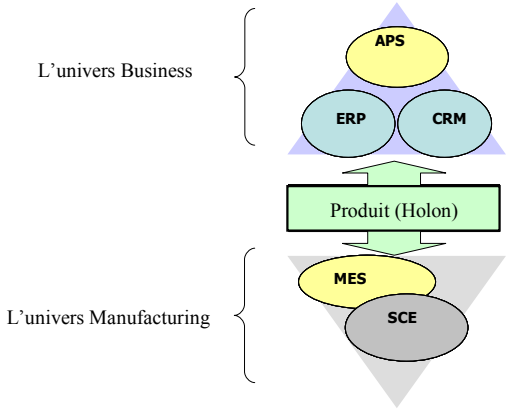

Fig. 6 : *Interopérabilité orientée produit*

L'objectif de cette section est de combiner l'approche *MDA pour l'interopérabilité* et l'approche de modélisation holonique pour établir l'interopérabilité entre modèles et applications de l'entreprise. Cette interopérabilité porte sur les informations et données concernant les produits, échangées entre l'ensemble des applications de l'entreprise. La première étape de notre démarche consiste tout d'abord à identifier l'univers du discours dans lequel nous nous situons, puis, positionner l'approche de modélisation holonique par rapport aux quatre niveaux ontologiques de l'approche MDA (*cf.* fig. 1). Ensuite, nous définirons les différents mappings permettant de passer d'un modèle utilisant un langage de modélisation d'entreprise spécifique, à un modèle holonique et vise versa ; l'objectif étant de faire de l'approche holonique pour le produit un moyen pour la communication entre modèles d'entreprise, dans le but de mettre en oeuvre l'interopérabilité orientée produit entre ces différents modèles.

Pour positionner l'approche holonique pour la modélisation du produit dans le cadre du MDA, nous avons identifié l'*univers du produit* comme étant l'univers de discours principal pour notre approche (M^0) ; cet univers est composé d'un ensemble de produits évoluant dans un environnement manufacturier. Ces produits sont modélisés en utilisant le concept d'holon. Ces holons et les relations entre eux ainsi que les concepts qui en découlent forment ce que l'on appelle des modèles holoniques (M^1). Nous situons, ainsi, le meta-modèle holonique qui décrit l'ensemble des concepts utilisés dans les modèles holoniques dans le niveau M^2.

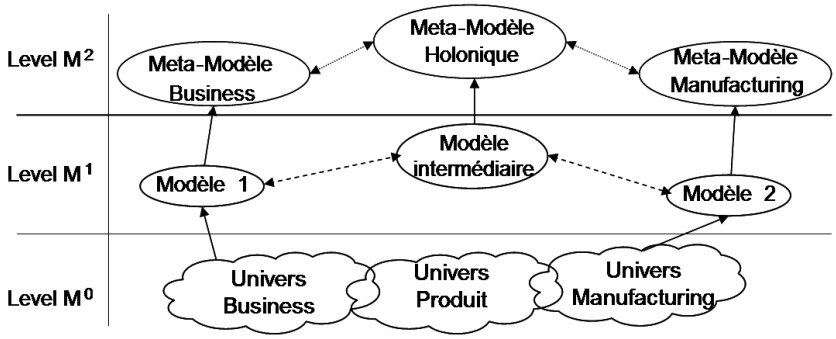

Fig. 7 : *Interopérabilité à travers l'holon produit dans le contexte MDA*

Comme le montre la figure 7, l'intérêt du meta-modèle passerelle (le meta-modèle holonique dans notre cas) s'explique par la diminution du nombre de mapping nécessaires pour mapper *n* meta-modèles ; en effet en cas de l'utilisation d'un meta-modèle passerelle *n* mappings sont nécessaires (un pour chaque meta-modèle), par contre en cas de non existence de meta-modèle passerelle pour connecter les *n* meta-modèles il faut définir *n* x *(n-1)/2* mappings pour relier l'ensemble des meta-modèles entre eux (nombres d'arcs bidirectionnels dans un graphe complet).

Pour illustrer l'utilisation du meta-modèle holonique, comme passerelle pour l'interopérabilité entre meta-modèles de différentes applications d'entreprise relatives aux niveaux de la figure 6, nous étudions le mapping entre le meta-modèle holonique et deux meta-modèles pour la modélisation d'entreprise : UEML (Berio, *et al*., 2003 ; Panetto *et al*, 2004) et the IEC 62264 standard (2002). Le but principal est de montrer que le meta-modèle définissant la représentation holonique du produit est interopérable avec d'un côté un meta-modèle issu du niveau Business (UEML), et un meta-modèle plutôt orienté Manufacturing (IEC 62264) ; et donc de ce fait, que le meta-modèle holonique peut effectivement jouer le rôle d'une passerelle entre les différents modèles coexistant dans l'entreprise (Business/Manufacturing).

- Premier exemple: mapping entre meta-modèle holonique et le meta-modèle de UEML (Unified Enterprise Modelling Language). UEML étant un langage pour la modélisation orientée Business. L'objectif de cette étude est de

montrer qu'à partir d'un modèle orienté Business formalisé en UEML nous pouvons extraire une description, formalisée en modèle d'holons, pour décrire chacun des éléments pouvant être échangés entre applications, systèmes et activités au niveau business, et vice versa.

- Deuxième exemple: mapping entre meta-modèle holonique et le meta-modèle du standard ISO/IEC 62264 utilisé pour la modélisation orientée Manufacturing. L'objectif est ici de montrer qu'il est possible de passer d'un modèle IEC 62264 concernant les produits manipulés au niveau Manufacturing à un modèle d'holons et vice versa.

3.3.1. UEML

UEML est le résultat du réseau thématique IST UEML (UEML, 2003) dont l'objectif principal était de résoudre le problème de l'existence d'une multitude de langages de modélisation d'entreprise (Vernadat, 1996). Ce projet a montré la faisabilité de définition d'un standard d'échange de modèles d'entreprise décrivant l'ensemble des notions et concepts pertinents pour la modélisation de processus d'entreprise, à savoir :

- *Activité :* une *activité* représente une description générique d'une partie du comportement de l'entreprise transformant un certain nombre d'entrée en sorties. Une *activité* regroupe plusieurs exécutions d'activité ayant des propriétés communes;
- *Objet :* Un *objet* représente toute sorte d'entité qui peut circuler via un flux; autrement dit, un objet est une entité qui peut être utilisée ou produite lors de l'exécution d'une activité, que ce soit une entité physique (Ressource) ou une entité purement informationnelle (Objet informationnel) ;
- *Objet Informationnel :* Un *Objet informationnel* est un *objet* constitué exclusivement d'information ;
- *Ressource*: Une *Ressource* est un sous-type d'objet support à l'exécution d'une activité. Il existe deux types de ressources: les *ressources physiques* et les *ressource humaine ;*
- *Flux :* Un *Flux* représente l'écoulement d'un certain nombre d'objets, dans le temps, depuis une source vers une destination. Un *flux* peut être de plusieurs types :
  - *Flux d'entrées/sorties :* représente les objets d'entrée/sorties des activités,
  - *Flux de contrôle:* véhicule l'information événementielle ou transactionnelle permettant de contraindre les activités,
  - *Flux de ressources:* véhicule les ressources nécessaires pour l'exécution d'une activité

Pour que le meta-modèle holonique soit dit interopérable avec le meta-modèle d'UEML, il nous faut trouver un mapping isomorphe (partiel ou total) entre le meta-modèle holonique et UEML. Le processus de mapping étant basé sur les relations sémantiques entre les concepts et leurs interprétations, il n'existe pas d'heuristique définie pour la découverte de mapping. Nous proposons le mapping résumé dans le tableau suivant comme premier mapping entre les deux meta-modèles.

Tableau 1: Mapping entre Holons et UEML

Meta-modèle Holonique	Meta-modèle UEML
Holon	Objet
Partie Informationnelle	Objet Informationnel
Partie Physique	Ressource Physique
Instance de Processus	Activité
Ressource	Ressource
Flux physique	Flux de ressource
Flux informationnel	Flux d'entrées/sorties (sous-type de)
Flux d'holons	Flux d'entrées/sorties (sous-type de)

3.3.2. ISO/ IEC 62264

Le standard ISO/IEC 62264 (2002) spécifie l'interface d'échange de données entre le niveau gestion d'entreprise (Business) et le niveau atelier (Manufacturing). Il définit les modèles et interfaces entre les activités de l'entreprise et les activités du niveau contrôle. Chaque niveau traite une vue particulière de l'intégration du niveau contrôle avec le reste de l'entreprise. Ces modèles peuvent être classes en deux catégories: modèles opérationnels et modèles de ressources.

Pour illustrer le mapping des concepts de ce standard avec notre modèle holonique, nous nous intéressons plus particulièrement à deux vues de ce standard :

- Le *material model* et le *product definition model.* Le *« material model »* est un modèle de ressource qui concerne les matières premières, les objets, les produits et les groupements de produits circulant dans le niveau atelier. Ce modèle spécifie la définition des classes de matière (material class) et les informations concernant ces classes. L'information traitée par ce modèle inclut l'inventaire des éléments bruts, finis, et intermédiaires.
- Le *« product definition model »* est le modèle de l'IEC 62264 qui définit l'information relative aux règles de production, les ressources utilisées et les besoins spécifiques pour la réalisation d'une opération de transformation.

Par analogie au mapping entre le meta-modèle holonique et UEML, nous proposons dans le Tableau 2, un mapping partiel, mais non formel, entre holon et les deux modèles du standard IEC 62264.

Tableau 2: Mapping entre Holons et ISO/IEC 62264

	Meta-modèle Holonique	ISO/IEC 62264
Material Model	Holon	Material sublot
	Flux d'Holon	Material lot
	Partie Informationnelle	Material definition
	Propriétés et Attributs	Material lot property definition
Product Definition Model	Instance de Processus	Product segment
	Ressource	Equipment specification

En résumé, le mapping défini entre le meta-modèle holonique et UEML d'un côté, et celui défini entre le meta-modèle holonique et l'IEC 62264 de l'autre, sont des mappings isomorphes et partiels, on peut donc en déduire que l'interopérabilité entre Holons et UEML (resp. IEC 62264) est une interopérabilité de niveau 1. Cet exemple a montré la faisabilité d'utilisation du meta-modèle holonique comme un meta-modèle de fédérateur pour l'échange des données concernant le produit et son cycle de vie entre les différents modèles d'application d'entreprises (Business ou Manufacturing).

Dans la suite de l'article, nous proposons une implémentation de l'approche de modélisation holonique sur un outil de modélisation commercial nommé MEGA Suite[2].

4. Implémentation

Pour expérimenter l'approche de modélisation holonique dans un cas réel, nous proposons de commencer par implémenter cette approche dans un outil de modélisation. Pour cet effet, nous avons utilisé l'outil de modélisation MEGA Suite, un outil commercial de modélisation de processus et du système d'information d'entreprise proposant des outils de modélisation, d'analyse et de conception d'environnement d'entreprise. MEGA Suite permet de définir et concevoir différents types de diagrammes et modèles: diagrammes de processus et de workflows, diagrammes UML, diagrammes d'architecture applicative, modèle de traitement et de flux de données.

Pour réaliser notre objectif, nous nous intéressons plus spécialement aux diagrammes de processus MEGA, qui reposent sur un meta-modèle proche de la notation BPMN (BPMN, 2004). En effet, MEGA est une des premières sociétés à

2. MEGA International, http://www.mega.com

avoir rejoint l'organisation BPMI où elle a participé activement aux travaux sur BPMN.

L'outil de modélisation MEGA est construit autour de son propre référentiel décrivant tous les concepts utilisés ainsi que les relations entre ces concepts. Ce référentiel peut être personnalisé et spécialisé pour une utilisation spécifique aux besoins de l'utilisateur. MEGA offre des outils pour permettre de spécialiser et personnaliser le référentiel interne; nous utilisons ses fonctionnalités pour intégrer les spécificités du meta-modèle holonique du produit dans MEGA, et ceci dans le but de tester l'utilisation de la modélisation holonique dans un cas pratique.

Le meta-modèle des diagrammes de processus sous MEGA définit la notion de « message »; comme étant un échange entre processus, activités, prodécures, acteurs ou autres. Pour intégrer le meta-modèle holonique MEGA, nous avons étendu le meta-modèle initial de MEGA par l'ajout des concepts nécessaires à notre modélisation. Nous avons commencé par créer la notion de "Flux" dans le meta-modèle ; de manière similaire à la notion de "message", puis, nous avons défini les éléments pouvant être contenus dans un "Flux", ce qui a donné lieu à l'intégration de la notion de "Partie Physique", de "Partie informationnelle", ainsi qu'au concept "Holon". Pour chacune de ces notions nous définissons, ainsi, une meta-classe ainsi que l'ensemble des relations entre ces meta-classes. Nous avons ainsi défini les notions de "Propriété holonique" et d' "Attribut holonique" permettant de décrire les caractéristiques d'un holon.

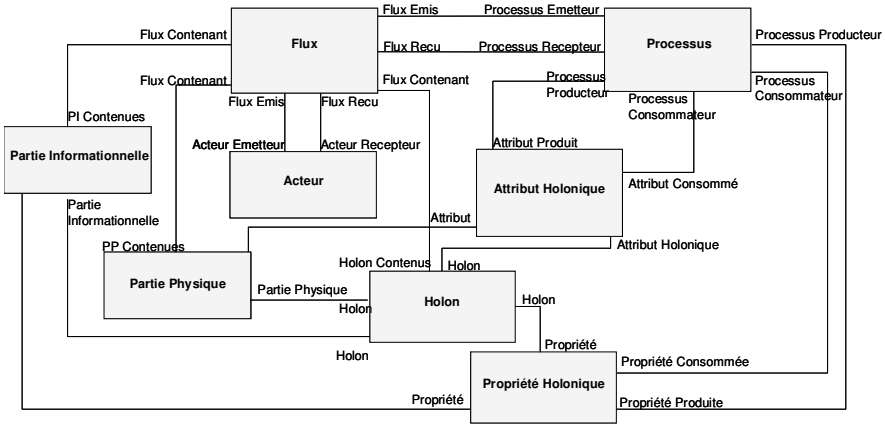

Fig. 8: *Prise en compte partielle du meta-modèle holonique dans MEGA.*

La figure 8 représente le meta-modèle étendu de MEGA après l'intégration des nouveaux concepts issus du meta-modèle holonique. Nous avons introduit deux nouvelles associations ; la première reliant directement un « holon » à une « Propriété Holonique » et permettant ainsi de faire abstraction des relations (Holon,

Partie informationnelle) et (Partie Informationnelle, Propriété Holonique), par analogie, la seconde relie un « holon » à un « Attribut Holonique ». L'objectif de ces deux associations étant de simplifier les diagrammes en diminuant le nombre d'éléments représentés et le nombre des liens entre eux.

L'environnement MEGA Suite permet aussi de définir des « descripteurs » spécifiant des formats d'import et export. Nous avons ainsi développé de tels descripteurs afin de générer automatiquement, à partir d'une instance de modèle holonique, des fichiers XML compatibles avec UEML, d'une part, et avec B2MML[3], l'implémentation XML du standard IES 62264 (B2MML, 2002).

5. Conclusion

Dans cet article, nous avons présenté notre approche pour l'interopérabilité basée sur une architecture MDA. Cette approche met en oeuvre l'architecture MDA comme support pour la définition de l'interopérabilité ; offrant ainsi le moyen de construire et vérifier l'interopérabilité existante entre deux applications. Cette approche utilise le paradigme de mapping entre meta-modèles pour assurer l'interopérabilité des modèles d'application. Nous définissons donc l'interopérabilité entre deux applications en étudiant les caractéristiques et qualité des différents mappings identifiés entre leurs meta-modèles respectifs. Cette approche permet de résoudre le problème de l'interopérabilité entre application construite autour d'un modèle possédant un meta-modèle bien défini. Cependant, il est nécessaire de formaliser ces mapping afin de s'assurer le maintien de la cohérence de la sémantique des concepts modélisés. Cette formalisation fait l'objet de travaux en cours non encore finalisés.

Dans la deuxième partie de cet article nous avons proposé un cas d'application à notre approche dans le contexte spécifique de l'interopérabilité entre applications d'entreprise manufacturière. L'objectif de cette application est de réaliser une interopérabilité orienté produit entre applications de l'entreprise et ce pour maintenir la cohérence des informations décrivant les produits. A cet effet, nous construisons un meta-modèle de référence pour la description des produits de l'entreprise ; ce meta-modèle se base sur le concept d'holon utilisé pour modéliser les produits, et leurs informations. Nous avons montré comment ce meta-modèle se situe dans la structure de l'entreprise pour jouer le rôle d'une passerelle entre les meta-modèles (pour la partie concernant le produit). A l'aide de deux exemples nous avons illustré la possible utilisation de mappings pour l'établissement de l'interopérabilité entre applications et leur implémentation dans un environnement de modélisation. Nos travaux portent maintenant sur la formalisation des mappings sur la base d'heuristiques du domaine.

3. B2MML : Business to Manufacturing Markup Language, http://www.wbf.org

6. Références

Adali S, Candan K, Papakonstantinou Y, Subrahmanian VS « Query caching and optimization in distributed mediator systems ». *In: Proceedings of SIGMOD*, Montreal, 4–6 June 1996, pp 137–148

B2MML (2003), The World Batch Forum. Business To Manufacturing Markup Language (B2MML), version 02, 2003.

Berio G, et al. « D3.2: Core constructs, architecture and development strategy », UEML TN IST – 2001 – 34229, March 2003

Bezivin, J. « From Object Composition to Model Transformation with the MDA ». *Proceedings of TOOLS* USA, Santa Barbara, August 2001.

BPMN (2004), OMG/BPMI Business Process Management Notation Specification, Version 1.0 http://www.bpmn.org/.

Chamberlin D., Robie J., Florescu D., « Quilt : an XML query language for heterogenous data sources », *Proc. of the Workshop on the Web and Databases (WedDB)*. In conj. with SIGMOD'00, Dallas - Texas, May 2000, Addison Wesley.

Fauvet M-C, Baina S, « Evaluation coopérative de requêtes sur des données semi-structurées distribuées », RSTI - ISI – 8/2003. Bases de données semi-structurées, pages 27 à 47

Garcia-Molina H., Papakonstantinou Y., Quass D., Rajaraman A., Sagiv Y., Ullman J., Vassalos V., Widom J., « The TSIMMIS Approach to Mediation: Data Models and Languages ». *Journal of Intelligent Information Systems*, Volume 8 Issue 2, March 1997.

Halevy G. Ives, Dan Suciu D. and Tatarinov I., « Schema mediation for large-scale semantic data sharing », The VLDB Journal — The International Journal on Very Large Data Bases, Volume 14 , Issue 1 (March 2005)

Halevy A, Madhavan J « Composing mappings among data sources ». *In: Proceedings of the conference on very large databases*, Berlin, Germany, 9–12 September 2003, pp 572–583

ISO/IEC 62264 (2002). ISO/IEC 62264-1:2002. *Enterprise-control system integration, Part 1. Models and terminology*, ISO

Kalfoglou, Y. and Schorlemmer, M. (2003). « Ontology Mapping: The State of The Art. » *The Knowledge Engineering Review*, 18: 1-31. 2003.Cambridge University Press

Kalfoglou, Y. and Schorlemmer, M. (2004). « Formal Support for Representing and Automating Semantic Interoperability ». In *Proceedings of 1st European Semantic Web Symposium (ESWS'04)*, 2004. pages pp. 45-61, Heraklion, Crete, Greece.

Koestler A. *The Ghost in the Machine* Arkana, London. (1967)

Lemesle, R. « Transformation Rules Based on Meta-Modelling ». *EDOC'98,* La Jolla, California, 3-5 November 1998, p. 113-122.

Levendovszky T., Karsai1 G., Maroti1 M., Ledeczi1 A. and Charaf H. « Model Reuse with Metamodel-Based Transformations ». In *Proceedings of the 7th International Conference on Software Reuse: Methods, Techniques, and Tools table of contents. 2002, April 15 - 19, 2002*. p. 166 - 178

Madhavan, P.A. Bernstein, P. Domingos, and A. Halevy. « Representing and reasoning about mappings between domain models". *In Proceedings of the 18th National Conference on Artificial Intelligence (AAAI'02)*, Edmonton, Alberta, Canada, August 2002.

Maedche A. and Staab S. « Semi-automatic engineering of ontologies from texts ». *In Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering (SEKE 2000)*, Chicago, IL, USA, pages 231–239, July 2000.

Miller R., Haas L. M. and Hernandez. M. « Schema mapping as query discovery ». *In Proceedings of the 26th International Conference on Very Large Data Bases, September 2000* Cairo Egypt..

Mellor S.J., Kendall S., Uhl A. and Weise D. *Model Driven Architecture,* Addison-Wesley Pub Co, March, ISBN: 0201788918. 2004

Naumenko, A. Wegmann, A. « Two Approaches in System Modelling and Their Illustrations with MDA and RM-ODP ». In *ICEIS 2003, the 5th International Conference on Enterprise Information Systems* (2003), p. 398-402.

Panetto H., Berio G., Benali K., Boudjlida N. and Petit M.. «A Unified Enterprise Modelling Language for enhanced interoperability of Enterprise Models».*In Proceedings of IFAC INCOM Symposium*, April 7th -9th 2004, Salvador de Bahia, Brazil

Papastavrou S., Pitoura E. and Samaras G. « Mobile Agents for WWW Distribued Database Access ». In the 15th *International Conf. on Data Engineering*., Sydney, Australia, March 1999.

Park J. and Ram S. « Information System Interoperability: What Lies Beneath? » *In ACM Transactions on Information Systems*, Vol. 22, No. 4, October 2004.

Rahm E, Bernstein PA « A survey of approaches to automatic schema matching » The VLDB Journal — The International Journal on Very Large Data Bases, Volume 10 issue 4:334–350 (2001)

Seidel, D. and Mey, M. (1994), «IMS - Holonic Manufacturing Systems: Glossary of Terms », In Seidel D. and Mey M. (eds), *IMS - Holonic Manufacturing Systems: Strategies* Vol. 1, March, IFW, University of Hannover, Germany.

UEML. (2003). Unified Enterprise Modelling Language (UEML) Thematic Network. IST-2001-34229, www.ueml.org.

Vernadat F.B.. *Enterprise modelling and integration: principles and applications.* Chapman & Hall (1996).

Wyns J, « Reference architecture for holonic manufacturing system », Ph.D. Thesis, Katholieke Universiteit Leuven 1999