
SISRO : Conception de bases de données à partir d'ontologies de domaine

Chimène Fankam* — **Ladjel Bellatreche*** — **Hondjack Dehainsala**** — **Yamine Ait Ameer*** — **Guy Pierra***

* LISI/ ENSMA Université de Poitiers - 86961 Futuroscope Chasseneuil Cedex
{fankamc, bellatreche, yamine, pierra}@ensma.fr

** Mondeca (hondjack.dehainsala@mondeca.com)

RÉSUMÉ. Les méthodologies de conception des bases de données exigent du concepteur à la fois, une connaissance du domaine modélisé et une maîtrise des techniques de modélisation. Le premier aspect est source d'importantes difficultés pour un concepteur appelé à passer de domaine à domaine. Par ailleurs, les bases de données résultantes sont très hétérogènes et difficiles à intégrer. Dans cet article, nous proposons une méthodologie de conception de base de données nommée SISRO. Elle se base sur la définition d'une ontologie locale à la base de données à construire définie par spécialisation d'une ontologie partagée de domaine, supposée préexistante et par importation sélective de propriétés et extension éventuelle de celle-ci pour les concepts et/ou les propriétés nécessaires qui n'y figuraient pas. Le modèle conceptuel est ensuite défini comme un fragment de l'ontologie locale. La représentation, au sein de la base de données cible, des ontologies locales et partagées et de leur articulation facilite l'accès aux données et permet l'échange et l'intégration.

ABSTRACT. Database design methodologies require both good modelling capabilities and knowledge of the field to be modelled. The first aspect is a one of the major difficulties for designers, since they have to move from one field to another one. Moreover, resulting databases are heterogeneous and difficult to integrate. In this paper, we propose a database design methodology, called SISRO. It is based on a prior definition of a local ontology for the database to be designed. This ontology is defined by specialization of shared domain ontologies, selective import of properties, and eventually extension with necessary concepts and/ or properties. The conceptual model is defined as a fragment of local ontology. Representing both ontologies and their articulations in the resulting database provides easy access, exchange and integration of data.

MOTS-CLÉS : Ontologie, Base de données, Conception de bases de données, Intégration.

KEYWORDS: Ontology, Database, Database design, Integration.

1. Introduction

Les approches de conception de bases de données actuelles passent principalement par trois étapes de modélisation : la modélisation conceptuelle, la modélisation logique et la modélisation physique. A partir d'un cahier des charges spécifique à une application donnée, le concepteur génère le dictionnaire de données qui énumère les différentes propriétés utilisées par l'application à modéliser. A partir de ce dictionnaire, le modèle conceptuel est construit en utilisant un formalisme de modélisation : modèle de Chen (Chen 1976) ou UML en général. Une validation de ce modèle conceptuel est effectuée par la définition de flux d'informations ou de processus au travers d'un modèle de traitement. Le modèle conceptuel de données (MCD) est ensuite traduit en un modèle logique de données dépendant du système de gestion de bases de données cible utilisé. Pour assurer une meilleure performance, l'administrateur de la base de données doit enfin sélectionner des structures d'optimisation (les index, le partitionnement, etc.) qui font partie du modèle physique. Ces étapes ont été adoptées par la plupart des méthodes (comme Merise) et par les outils de conception de système de gestion de bases de données comme PowerAMC de Sybase.

La conception des bases de données selon l'approche ci-dessus présente néanmoins trois difficultés :

- *La difficulté de conception.* Le concepteur doit tout d'abord acquérir une maîtrise du domaine de l'application à mettre en place. Une telle compréhension est en effet nécessaire pour permettre l'identification et le choix des concepts pertinents pour l'élaboration de la base de données et son intégration future. Le temps et l'effort demandés pour aboutir à cette maîtrise sont assez considérables, surtout lorsque le concepteur change fréquemment de domaine d'application. Par exemple, la conception d'une base de données d'ingénierie nécessite de modéliser l'ensemble des composants (mécaniques, électroniques, logiciels, etc.) utilisés dans une entreprise pour concevoir ses produits. Les familles de composants sont modélisées à différents niveaux d'abstraction (une vis, une vis hexagonale, une vis hexagonale partiellement fileté) et elles possèdent des caractéristiques et des structures spécifiques. La modélisation d'un tel univers nécessite d'intégrer une expertise métier approfondie souvent difficile à appréhender pour un concepteur n'ayant aucune formation technologique.

- *L'hétérogénéité du résultat.* Les modèles conçus par deux équipes de conceptions différentes pour deux bases de données ayant des modèles de traitements voisins dans des champs d'application similaires, seront en général suffisamment différents pour qu'il devienne très difficile, à posteriori, de les intégrer (Wache *et al.* 2001). Cette hétérogénéité est due à deux causes :

- Les contextes applicatifs et les traitements à effectuer sont toujours légèrement différents. L'identification des entités et des relations pertinentes pour l'application à informatiser se fait généralement à partir des informations préexistantes (cahier des charges) et de l'objectif applicatif du système. Ces deux éléments sont toujours légèrement différents. Ainsi, chaque cahier des charges débouche sur un modèle conceptuel spécifique. Chaque modèle est d'autant plus différent (Kim *et al.* 1991) des autres

modèles relevant du même domaine que ces derniers ne peuvent pas être réutilisés. En effet, les méthodologies usuelles ne permettent pas de se fonder sur l'existant. Ainsi, même si elles concernent essentiellement le même domaine, les bases de données existantes présenteront toujours de nombreux conflits rendant difficile toute tentative d'intégration (Goh *et al.* 1999, Bellatreche *et al.* 2006).

- Chaque concepteur possède son propre style de modélisation. Le choix des entités et relations dans le modèle conceptuel dépend de la vision du domaine par le concepteur et de ses habitudes de modélisation, par exemple, du point de vue nommage (nom vs. identifiant), typage (numéro de sécurité sociale : chaîne de caractères vs. entier), ou granularité (adresse : chaîne de caractères vs. entité constituée de nombreuses propriétés). Les méthodes usuelles laissent une large marge de manœuvre au concepteur de ce point de vue et, pour un même cahier des charges, deux concepteurs différents aboutiront en général à deux modèles conceptuels assez largement différents.

- *La difficulté d'utilisation ultérieure de la base de données résultante.* Malgré les problèmes d'hétérogénéité soulignés précédemment, le modèle conceptuel permet d'exprimer à la fois les besoins applicatifs et la connaissance du domaine sous une forme intelligible pour un utilisateur ultérieur. Malheureusement, c'est le modèle logique qui est exploité ; et celui-ci, résultant de la normalisation et de l'adaptation au système support, est en général très différent du modèle conceptuel. Ainsi, l'absence de représentation du modèle conceptuel dans la base de données cible rend l'interrogation de celle-ci très difficile même pour les utilisateurs ayant une bonne connaissance du domaine d'application. En conséquence, l'équipe de développement sera fréquemment appelée à recoder le modèle conceptuel dans l'applicatif d'accès. Par exemple, les tables d'association générées lors du passage du modèle conceptuel vers le modèle logique et la migration des clés à l'issue de la normalisation de ce dernier conduisent à un modèle difficilement compréhensible par l'utilisateur. D'où la nécessité de coder le MCD afin d'offrir à l'utilisateur un accès via un modèle compréhensible par ce dernier.

Nous présentons dans cet article une méthodologie de conception de base de données à partir d'ontologies qui vise à la fois à aider le concepteur dans sa tâche, et à rendre le contenu résultant localement intégrable et exploitable, en particulier dans le contexte du Web sémantique. Cette approche, est particulièrement pertinente dans les domaines tels que la médecine (Haendel *et al.* 2007) ou l'ingénierie (Pierra 2008) dans lesquels des ontologies standard sont en voie d'émergence. Elle se base sur la définition préalable d'une ontologie locale à la base de données à construire, et sur son articulation avec la ou les ontologies partagées du domaine. La représentation de cette ontologie locale, et de ses correspondances avec les ontologies partagées dans la base de données, permet de faciliter l'intégration automatique et l'échange des données contenues dans la base de données au sein de la communauté dans laquelle la ou les ontologies partagées font l'objet d'un consensus.

Cet article est organisé comme suit. Dans la section 2, nous discutons de la notion d'ontologie de domaine et des relations entre ontologies et modèles conceptuels de

données. Dans la section 3, nous présentons les différentes approches visant à rapprocher bases de données et ontologies, et nous discutons de leurs limitations. La section 4 présente les différentes étapes de notre méthodologie de conception dont nous présentons dans la section 5, la mise en œuvre. La section 6 discute les apports de SISRO pour réduire les difficultés identifiées ci-dessus. La section 7 précise le contexte dans lequel a été élaborée la méthode SISRO et les perspectives de ce travail. Enfin, une conclusion est donnée dans la section 8.

2. Ontologie et base de données

Les ontologies sont utilisées dans de nombreux domaines pour faciliter l'intégration et l'échange d'informations. Ces dernières années ont vu l'apparition de plusieurs travaux de rapprochement entre bases de données et ontologies visant soit à faciliter la conception des bases de données, soit à faciliter leur intégration dans des environnements ouverts tels que le Web Sémantique. Nous discutons dans cette section des caractéristiques des ontologies de domaine qui les rendent attractives dans une perspective base de données.

2.1. Ontologie de domaine

Définie initialement par T. Gruber (Gruber 1993) comme *une spécification explicite d'une conceptualisation*, définition ensuite étendue par N. Guarino (Guarino *et al.* 1995) qui en souligne le caractère formel, une ontologie est essentiellement une représentation explicite de la sémantique d'un domaine, telle qu'elle est perçue par une communauté donnée. Nous retiendrons, quant à nous, la définition proposée dans notre laboratoire (Jean *et al.* 2007) qui caractérise une ontologie comme *une représentation formelle, référençable et consensuelle de l'ensemble des concepts partagés d'un domaine sous forme de classes, de propriétés et de relations qui les lient*. Cette définition met en effet bien en évidence les trois caractéristiques qui distinguent une ontologie de domaine des autres modèles de domaine : modèle conceptuel et modèle de connaissance. Une ontologie est une représentation :

- *formelle*, permettant ainsi des raisonnements automatiques ayant pour objet soit d'inférer de nouveaux faits, soit d'effectuer des vérifications de consistance ;
- *consensuelle*, c'est-à-dire admise par l'ensemble des membres (et des systèmes) d'une communauté et,
- *référençable*, c'est-à-dire que toute entité ou relation décrite dans l'ontologie peut être directement référencée par un symbole, à partir de n'importe quel contexte, afin d'explicitier la sémantique de l'élément référençant.

Les ontologies peuvent être utilisées dans beaucoup de contextes et pour des raisons différentes (Guarino *et al.* 1995). Ainsi, plusieurs formalismes de représentation des ontologies ont été proposés. On peut citer en particulier : RDF Schéma (Brickley *et al.* 2002), DAML+OIL (Connolly *et al.* 2001), OWL (Bechhofer *et al.* 2004), PLIB

(Pierra 2008), etc. Malgré cette diversité, il apparaît que tous ces formalismes possèdent un noyau commun que nous proposons de formaliser de la façon suivante : Une ontologie de domaine O est définie par le quadruplet : $O : \langle C, P, Sub, Applic \rangle$, avec :

- C : ensemble des classes utilisées pour décrire les concepts d'un domaine donné. Chaque classe est associée à un identifiant globalement unique. L'ensemble des classes peut être partitionné en deux sous-ensembles $C = C_c \cup C_{nc}$; où C_c représente l'ensemble des classes *primitives* ou *canoniques*, c'est à dire celles "pour lesquelles nous ne sommes pas capable de donner une définition axiomatique complète et dont la compréhension repose sur une connaissance préalable du lecteur" (Gruber 1993). Les classes de C_c définissent donc un langage canonique pour caractériser les objets du domaine. Les classes de C_{nc} , dites *définies* ou *non-canoniques*, sont celles qui peuvent se définir complètement à partir des classes canoniques. Par exemple, si *Homme* et *Femme* sont des classes canoniques, *Personne* est une classe non canonique puisque, dans tout contexte, $Personne = Homme \cup Femme$. Inversement, si *Personne* est une classe canonique et *genre* est défini comme une propriété applicable, *Homme* et *Femme* sont des classes non canoniques.

- P : ensemble des propriétés utilisées pour décrire les instances de l'ensemble des classes C par des valeurs appartenant soit à des types simples, soit à d'autres classes ¹. Chaque propriété est associée à un identifiant globalement unique. L'ensemble des propriétés peut également être partitionné en propriétés canoniques et non canoniques : $P = P_c \cup P_{nc}$. Par exemple, le *nom complet* d'une personne peut être défini comme la concaténation de son *prénom* et de son *nom*. Si *nom* et *prénom* ont été choisies comme propriétés canoniques, le *nom complet* est non canonique.

- $Sub : C \rightarrow 2^C$; la relation de subsomption ² qui, à chaque classe c_i de l'ontologie, associe ses classes subsumées directes ³.

- $Applic : C \rightarrow 2^P$, la fonction qui associe à chaque classe c_i de l'ontologie les propriétés qui sont applicables pour cette classe, c'est-à-dire, qui peuvent être utilisées pour en décrire les instances. Seules les propriétés ayant pour domaine une classe, ou l'une de ses super-classes, sont applicables à cette classe. Si le domaine d'une propriété n'est pas spécifié, la racine de l'ontologie est implicitement considérée.

Notons que la subsomption, relation ontologique, ne doit pas être confondue avec l'héritage, relation logique introduite dans les langages à objets. Une ontologie ne prescrivant en aucun cas les propriétés devant être évaluées pour chaque instance, le fait qu'une propriété soit applicable pour une classe n'impose en aucun cas que toutes les instances de cette classe (ou d'une de ses sous-classes) possèdent une valeur pour cette propriété.

1. Le terme propriété regroupe les deux notions parfois distinguées sous les noms d'attribut et de relation (ou rôle)

2. c_1 subsume c_2 ssi : $\forall x \in c_2, x \in c_1$

3. c_1 est une subsumante directe de c_2 dans une ontologie O ssi la relation de subsomption entre c_1 et c_2 ne peut se déduire par transitivité, des autres relations de subsomption définies entre les différentes classes de l'ontologie O

2.2. Comparaison entre ontologie de domaine et MCD

Les ontologies, comme les modèles conceptuels, conceptualisent l'univers du discours au moyen de classes hiérarchisées par subsomption et associées à des propriétés. Les ontologies présentent cependant quatre différences par rapport aux modèles conceptuels. Ces différences sont à la base de la contribution que peuvent apporter les ontologies dans la problématique des bases de données.

– *Objectif de modélisation.* Une ontologie *décrit* les concepts d'un domaine d'un point de vue assez général, indépendamment de chaque application précise et de tout système dans lequel elle est susceptible d'être utilisée. A l'opposé, un MCD *prescrit* l'information qui doit être représentée dans un système informatique particulier pour faire face à un besoin applicatif donné. Les éléments de la connaissance du domaine représentés par une ontologie et pertinents pour un système particulier pourront donc être extraits de l'ontologie par le concepteur du système sans que celui-ci ait besoin de les redécouvrir. Ainsi par exemple, la norme IEC 61360-4 (IEC 61360-4 1998) constitue un dictionnaire formel et consensuel de la plupart des catégories de composants électroniques existants, ainsi que leurs propriétés et des relations qui les relient. Ce domaine constitue donc une description de référence du domaine des composants électroniques dont un concepteur de bases de données peut aisément extraire, en interaction avec les utilisateurs, les entités, propriétés et relations devant être représentés au sein d'une base de données d'ingénierie d'une entreprise de conception électronique. Une démarche tout à fait analogue pourrait exister dans le domaine bancaire pour autant qu'une ontologie des produits bancaires usuels soit formalisée.

– *Atomicité des concepts.* A la différence d'un MCD, où chaque concept ne prend son sens que dans le contexte du modèle dans lequel il est défini, dans une ontologie, chaque concept est identifié de façon individuelle et constitue une unité élémentaire de connaissance. Un MCD peut donc extraire d'une ontologie seulement les concepts (classes et propriétés) pertinents pour son objectif applicatif. Il peut également, sous réserve de respecter leur sémantique (par exemple subsomption), les organiser de façon assez différente de leur organisation dans l'ontologie, la référence au concept ontologique permettant de définir avec précision la signification de l'entité référençante.

– *Consensualité.* Une ontologie de domaine représentant les concepts sous une forme consensuelle pour une communauté, l'exploitation de l'ontologie tant dans une phase de conception, par les concepteurs, qu'ultérieurement dans une phase d'exploitation, par un utilisateur, permet un accès naturel et ergonomique aux informations du domaine dès lors que concepteur et utilisateur relèvent de la communauté visée par l'ontologie. De même, l'intégration sémantique de tous les systèmes basés sur une même ontologie pourra facilement être réalisée si les références à l'ontologie sont explicitées.

– *Non canonicité des informations représentées.* A la différence des modèles conceptuels qui utilisent un langage minimal et non redondant pour décrire les informations d'un domaine, les ontologies utilisent, en plus des concepts primitifs, des concepts définis qui fournissent donc des accès alternatifs à la même information. La représentation de ces concepts non canoniques sous forme de vue sur la base de

données permet d'étendre encore l'ergonomie d'accès à l'information.

Ainsi, dans les domaines où existent des ontologies, leur prise en compte dans le processus de conception des bases de données pourrait permettre de réduire de façon significative les trois difficultés identifiées dans l'introduction.

3. Approches existantes de rapprochement entre ontologies et MCD

Trois principales approches de rapprochement des bases de données et des ontologies ont été proposées. Les approches *a posteriori*, dans lesquelles la conception du MCD précède la recherche ou la conception de l'ontologie partagée, visent à faciliter l'accès *a posteriori* d'une base de données préexistante à partir d'une ontologie. Les approches *a priori*, dans lesquelles l'ontologie est supposée existante lorsque la base de données est conçue, visent à aider à concevoir son MCD. Enfin, les approches directes, dans lesquelles la structure de l'ontologie définit directement la structure de la base de données.

Nous présentons ci-dessous ces différentes approches, en discutant de leurs contributions possibles pour résoudre les trois difficultés énoncées dans l'introduction. Ces contributions sont analysées à travers trois critères. Le premier correspond à la résolution de la première difficulté : difficulté de modélisation d'un domaine inconnu. Le deuxième évalue la facilité d'accès à la base de données résultante par les acteurs du domaine, ainsi que les capacités d'intégration. Il s'agit donc des deuxième et troisième difficultés identifiées dans l'introduction. Enfin, le troisième critère vérifie que l'approche proposée ne va pas à l'encontre de l'objectif fondamental du processus de conception d'une base de données : à savoir que celle-ci satisfait, in fine, les besoins applicatifs pour lesquels elle est définie.

Ces critères d'évaluation sont donc les suivants :

- C1) l'aide apportée lors de la conception, pour faciliter la compréhension du domaine cible et sa modélisation ;
- C2) l'aide apportée à l'intégration, à l'échange des données, et à l'interrogation ultérieure de la base de données par un utilisateur du domaine ;
- C3) l'autonomie dont dispose le concepteur de la base de données pour assurer la bonne adaptation de la base de données au contexte applicatif particulier auquel elle est destinée.

3.1. Approches *a posteriori*

L'objectif des approches *a posteriori*, est d'explicitier la sémantique d'une base de données existante en permettant son interrogation au travers d'ontologies. L'approche proposée par Roldan-Garcia et ses collègues (Roldan-Garcia *et al.* 2005) distingue trois étapes principales :

– *la conception ou la rétro-conception du MCD*. Dans cette première étape, si le MCD n'est pas disponible, il est reconstruit par des techniques de rétro-conception à partir de la base de données.

– *la recherche d'une ontologie de domaine*. Dans cette étape, on identifie une ontologie partagée dont le domaine recouvre au mieux le domaine d'application de la base de données implémentée.

– *la définition des correspondances entre MCD et ontologie partagée*. Cette dernière étape consiste à identifier les correspondances pouvant être établies entre les concepts (classes et propriétés) du MCD et ceux de l'ontologie partagée. Ces auteurs suggèrent alors d'exploiter ces correspondances dans un adaptateur susceptible d'interpréter les requêtes formulées en termes de l'ontologie partagée. Notons que, dans ce travail, aucun détail n'est fourni ni sur les ontologies visées, ni sur les correspondances établies, ni sur leur exploitation au sein d'outils.

De façon beaucoup plus précise, Barassa et ses collègues (Barassa *et al.* 2004) ont proposé un langage R_2O permettant de spécifier directement des correspondances entre une ontologie existante, décrite en RDFS ou en OWL et une base de données préexistante. L'approche suivie est une approche de type Global As View (GaV) (Chawathe *et al.* 1994) dans laquelle les concepts de l'ontologie partagée (considérée comme un schéma global) sont décrits comme des vues sur les relations internes dans la base de données. L'originalité de l'approche est de définir les vues non seulement par les opérateurs algébriques SQL, mais également par un langage d'appariement permettant d'exprimer les correspondances entre concepts de l'ontologie et vues (SQL) sur la base de données. R_2O permet d'abord de déclarer la structure de l'ontologie cible et celle de la base de données source. Puis, les appariements seront définis classe par classe, en trois étapes :

- d'abord, si besoin est, la vue SQL telle qu'une instance de cette vue puisse être en relation bijective avec un individu de la classe cible de l'ontologie est identifiée (seuls les opérateurs de sélection, projection et jointures sont utilisables, sans possibilité d'imbrication),

- ensuite, un appariement (*conceptmap-def*) est spécifié pour chaque instance de classe de l'ontologie. Cet appariement définit : (1) à quelle condition un tuple de la vue correspond à une instance de classe ; (2) quelle est la clé de ce tuple ; (3) quel est l'identifiant (*uri-as*) de cet individu (calculé en général, d'après la clé du tuple),

- enfin, un appariement (*attributemap-def*) est spécifié pour chaque attribut ou relation de l'instance identifiée. Cet appariement définit : (1) à quelle condition une valeur d'attribut du tuple correspond à une valeur d'attribut de l'instance (*applies-if*) ; (2) quelle transformation doit subir sa valeur (*aftertransform*).

Le langage ainsi défini a une puissance d'expression significative puisqu'il combine une partie de la puissance de SQL avec celle des primitives de filtrage et de transformation introduites. On peut cependant noter (1) que le langage ne comporte pas de partie impérative permettant par exemple d'exprimer des itérations à la différence de la plupart de langages modernes de correspondance entre modèles et (2) que la séparation entre, d'une part, l'expression des correspondances et leur interpréteur qui joue

le rôle d'adaptateur ("wrapper" (Wiederhold 1992)) et d'autre part, l'interpréteur de requête SQL, ne permet pas d'optimiser la traitement global.

Si les méthodes a posteriori permettent de rendre accessibles ou d'échanger les données d'une base de données en termes d'une ontologie correspondant au même domaine (critère C2), elles ne permettent pas, par contre, d'aider à la conception d'une base de données (critère C1). Ce sera l'objectif des méthodes a priori.

3.2. *Approches a priori*

Dans les approches a priori, une ontologie (ou éventuellement plusieurs) partagée est tout d'abord identifiée. Elle est ensuite prise en compte tout au long du processus de conception de la base de données. Cette méthode est préconisée lorsqu'il existe au préalable des ontologies partagées décrivant la sémantique du domaine cible, ou lorsque les applications à mettre en œuvre sont appelées à subir des évolutions. Roldan-Garcia et ses collègues (Roldan-Garcia *et al.* 2005) ébauchent, pour les approches a priori également, une méthodologie. Celle-ci passe par deux étapes :

- l'extension, au besoin, de l'ontologie de domaine avec de nouveaux concepts (classes et propriétés) manquants et nécessaires pour le système à mettre en place.
- l'extraction du sous-ensemble de classes de l'ontologie ainsi étendue couvrant les besoins du système en cours. Ce sous-ensemble constitue le MCD et sera utilisé dans la suite du processus de conception de la base de données.

Cette méthodologie vise à la fois à faciliter la conception de la base de données en réutilisant la sémantique formalisée du domaine telle qu'elle est définie dans les ontologies, et à permettre, ensuite, son accès à partir de l'ontologie du domaine utilisée. Cette approche correspond à la même philosophie que celle que nous proposons ici, néanmoins elle reste très préliminaire, rien n'étant défini concernant les correspondances existant entre les différentes ontologies et les MCDs générés, ni sur la façon dont ces correspondances pourraient être utilisées dans une perspective d'intégration (critère c2).

Au contraire, Suguraman (Sugumaran *et al.* 2006) propose un véritable outil d'aide à la conception de MCD à partir d'ontologies de domaine. L'approche proposée permet d'ailleurs également la validation de MCDs existants par rapport à une ontologie du même domaine. L'approche suivie est une approche linguistique dans laquelle, à partir d'une expression en langage naturel du cahier des charges de l'application et du choix d'ontologie(s) de domaine, le système analyse les termes du cahier des charges et, par référence à l'ontologie, suggère différents éléments à introduire dans le modèle conceptuel. Afin de rendre la méthode plus efficace, les auteurs proposent d'ajouter à l'ontologie des contraintes d'intégrité existentielles permettant de raisonner sur le modèle conceptuel. Ainsi, la contrainte ontologique *prerequisite* spécifie qu'un *acheteur* ne peut exister que s'il existe un *item* à vendre. Si le MCD en cours possède la notion d'*acheteur* mais pas celle d'*item* (ou un terme dérivé, tel que *produit* trouvé dans l'ontologie), le système pourra soit suggérer d'introduire ce terme (en aide à la

conception) soit définir le modèle comme invalide (en validation de MCD). Un intérêt de cette approche est que les auteurs montrent, à travers un protocole expérimental rigoureux, que l'exploitation de la connaissance spécifique à un domaine représentée dans une ontologie de domaine aide effectivement et de façon significative le concepteur dans sa tâche (critère C1). Par contre, pour la validation de leur approche, les auteurs utilisent une ontologie définie à partir d'un corpus par traitement automatique de la langue. Il s'agit donc d'une ontologie peu formelle, qui ne possède en particulier, pas d'attribut. De plus, en aide à la conception, la méthode peut suggérer au concepteur des concepts et relations supplémentaires, extraits de l'ontologie de domaine, mais elle ne permet pas de garder le lien entre les concepts et relations de l'ontologie et le MCD défini. Ainsi, aucun lien formel n'est établi in fine entre les concepts et relations de l'ontologie, et les concepts et relations qui constituent le MCD final. Cette approche ne permet donc pas d'accéder à la base de données finale via l'ontologie de domaine et ne facilite ni l'échange de données, ni l'accès aux données (critère C2). Enfin, aucun mécanisme n'est fourni pour étendre ou spécialiser l'ontologie pour l'adapter aux besoins applicatifs particuliers auxquels doit répondre le MCD (critère C3).

3.3. Utilisation directe d'une ontologie pour structurer une base de données

Dans le domaine du Web Sémantique, avec l'émergence des ontologies sont également apparues des données à base ontologiques. Utilisées en particulier pour annoter des documents, ces données, constituées d'instances de classes de l'ontologie ont rapidement atteint des tailles qui devenaient incompatibles avec le traitement en mémoire centrale. De nombreux chercheurs ont alors proposé des structures de base de données pour gérer ces données.

Différentes structures persistantes de stockage des données à base ontologique ont ainsi été proposées (Broekstra *et al.* 2002, Alexaki *et al.* 2001, Pan *et al.* 2003, B.McBride 2001, Bozsak *et al.* 2002, Chong *et al.* 2005, Park *et al.* 2007, Dehainsala *et al.* 2007, Ma *et al.* 2008). Le schéma proposé le plus simple consiste à stocker à la fois les ontologies et les données à base ontologique dans une unique table à trois colonnes (sujet, prédicat, objet) (B.McBride 2001, Alexaki *et al.* 2001). Dans cette représentation, dite "par triplet", l'ensemble des informations, ontologies et données, est décomposé sous la forme de triplets, ce qui correspond précisément à la structure de RDF. Cette représentation est également utilisée par les systèmes, dits RDF natifs, directement construits sur des systèmes de fichiers pour gérer les données RDF indépendamment de toute base de données, tels que OWLIM (Kiryakov *et al.* 2005), HSTAR (Chen *et al.* 2006) ou YARS (Harth *et al.* 2005). Dans le contexte des bases de données, la principale faiblesse de cette approche est la non distinction ontologie/données qui entraîne la nécessité d'un très grand nombre d'auto jointures de la table de triplets, souvent de très grande taille (Alexaki *et al.* 2001). Pour éviter ce problème, les systèmes plus récents ont proposé soit la création de vues matérialisées soit la séparation entre ontologie et données.

La structure par vues matérialisées est utilisée, par exemple, par Oracle (Chong *et al.* 2005) pour le premier système de gestion de données RDF au dessus d'une base de données commercialisée. Dans ce système, Oracle propose de stocker ontologie RDFS et données RDF sous forme de triplets dans deux tables. La première, *IdTriples*, stocke les triplets sous une forme normalisée en utilisant des identifiants. La seconde, *UriMap*, permet d'associer à chaque identifiant de la table *IdTriples* l'URI correspondante. Le stockage de littéraux est également réalisé dans la table *UriMap*. La première forme rencontrée d'un littéral est considérée comme sa forme canonique. Les formes équivalentes rencontrées ensuite réfèrent la forme canonique pour faciliter les comparaisons. Afin de réduire le coût de jointure liée à l'utilisation de la table principale de triplet *IdTriple*, les prédicats tels que *subProperty* sont sauvegardés dans une structure redondante de type *materialized joined views*. La description de chaque instance par un ensemble de valeurs de propriétés peut également être matérialisé par une vue, appelée *subject_property matrice materialized join views*. De plus, le système offre la possibilité de définir implicitement de nouveaux triplets par des règles, ces nouveaux triplets pouvant eux-mêmes être matérialisés dans des vues. Cette extension d'Oracle constitue donc un système autonome, permettant de gérer efficacement les données RDF/RDFS du Web sémantique. Une caractéristique importante de l'approche proposée ici est de permettre l'interrogation de ces données dans une syntaxe SQL par l'introduction d'une fonction *RDF_MATCH* qui accède aux données RDF/RDFS. La même requête SQL permet donc d'accéder à la fois aux données RDF/RDFS, et aux autres données représentées dans la base de données. Notons que cette caractéristique pourrait permettre d'offrir simultanément un accès traditionnel aux données, à partir des tables SQL et un accès à partir d'ontologies.

La seconde structure de représentation des données à base ontologique consiste à séparer la représentation des ontologies et des données. Les ontologies sont représentées au sein d'un schéma dépendant du modèle d'ontologie utilisé (RDFS, OWL, etc.). Les données sont stockées soit par une représentation par triplet, soit sous une forme éclatée dans laquelle chaque classe est représentée par une table unaire et chaque propriété par une table binaire (Pan *et al.* 2003, Broekstra *et al.* 2002). C'est cette deuxième approche qui est suivie par le système SOR (Lu *et al.* 2007) proposé par IBM sur le SGBD DB2. Ce système permet, outre RDF/RDFS, de gérer et de raisonner sur des ontologies OWL. A la différence d'Oracle, le modèle d'ontologie est explicitement représenté dans la base de données en associant une table aux différents prédicats de l'ontologie (ex. *TypeOf*, *someValuesFrom*, *IntersectionOf*, *Domain*, etc.). Afin d'optimiser les traitements sur les données, SOR comporte à la fois un raisonneur sur l'ontologie et un moteur d'inférence sur les données afin de précalculer et de matérialiser tous les axiomes (ex. *subClass*, *subProperty*, *inverse*, etc.) et tous les nouveaux faits ou triplets inférés dans la base de données. Enfin, SOR propose un traducteur *SPARQL-to-SQL* doté d'une gestion efficace des patrons utilisées par SPARQL pour l'interrogation de données.

Quelque soit la structure utilisée, l'approche directe qui range dans une base de données à la fois une ontologie et les données conformes à cette ontologie, permet évidemment de rendre cette base de données facilement intégrable avec toutes les autres

bases de données construites sur la même ontologie. Par contre, il ne s'agit pas d'une base de données au sens usuel. En effet, la même information pouvant être représentée de différentes façons (par exemple : *Femme*, ou *Personne* : *genre = F*), l'hypothèse du monde fermé ne pourra pas être posée, et des capacités de raisonnement seront nécessaires à l'intérieur du système que ce soit par saturation a priori, ou par adjonction d'un moteur au sein du système. De plus, dans un tel système, le concepteur n'a aucune autonomie ni pour permettre des références simultanées à plusieurs ontologies si celles-ci se recouvrent partiellement, ni pour adapter le MCD à ses besoins applicatifs particuliers (critère *c3*).

3.4. Limites des approches existantes

Le tableau 1 compare les approches existantes présentées ci-dessus. Ainsi, il apparaît clairement que l'approche a priori et l'approche directe constituent une aide à la conception car le MCD est dérivé de la (ou des) ontologie(s) partagée(s) préalablement existante(s). Le concepteur dispose donc d'un point de référence où sont définis les concepts du domaine ainsi que les relations qui peuvent exister entre eux. Inversement, l'approche a posteriori ne se basant pas sur une conceptualisation préexistante n'offre donc aucune aide à la conception. L'approche a posteriori et l'approche directe facilitent l'échange, l'intégration des données et l'accès aux données par les utilisateurs du domaine (+). Par contre cela est au prix, pour la méthode a posteriori, d'une rétro-conception souvent complexe de la base de données (-). Les approches a priori actuelles, quant à elles, ne facilitent pas ou peu l'intégration car le lien entre données et ontologie est ou absent (Sugumaran *et al.* 2006) ou indéfini (Roldan-Garcia *et al.* 2005). Enfin, concernant l'autonomie de l'administrateur qui est un aspect très important dans l'univers des bases de données puisque c'est ce dernier qui permet d'assurer la bonne adaptation d'une base de données au contexte applicatif particulier auquel elle est destinée, dans l'approche a posteriori, le MCD est défini en toute liberté par le concepteur. Dans l'approche a priori, le concepteur peut extraire ce qui lui apparaît pertinent de l'ontologie de domaine puis enrichir le MCD obtenu par introduction de nouveaux concepts spécifiques. Enfin, dans l'approche directe aucune autonomie n'est offerte dans la mesure où le MCD est strictement inclus dans l'ontologie utilisée.

	Aide à la conception	Intégration Echange	Autonomie
A priori existantes	+	-	+
A posteriori	-	+/-	++
Approches directes	+	++	-

Tableau 1. Comparaison des approches existantes

Nous présentons dans la section suivante une nouvelle approche de conception des bases de données à partir d'ontologies qui vise à remédier aux limitations des approches existantes. Cette approche, de type a priori, permet, en effet, à la fois d'aider

le concepteur, de respecter son autonomie et de faciliter l'intégration automatique de la base de données obtenue en explicitant la sémantique des données qu'elle contient.

4. Notre approche de conception

Notre approche de conception des bases de données que nous appelons méthode SISRO (pour Spécialisation des classes, Importation Sélective des propriétés et Représentation des Ontologies), est une approche a priori dans laquelle le concepteur (1) extrait par spécialisation des ontologies de domaine existantes les concepts pertinents pour le système à mettre en œuvre, (2) importe de façon sélective les propriétés⁴ nécessaires, (3) étend si besoin est l'ontologie ainsi définie, représente effectivement l'ontologie dans la base de données.

Cette méthode vise à (1) permettre au concepteur de réutiliser la connaissance du domaine formalisée dans l'ontologie, (2) offrir au concepteur une liberté de structuration des concepts importés de l'ontologie partagée, (3) faciliter l'échange de données, (4) permettre l'intégration automatique et (5) offrir un accès aux données au niveau connaissance, c'est à dire au niveau ontologique. Pour atteindre cet objectif, SISRO présente les caractéristiques originales suivantes :

- le modèle conceptuel n'est pas fondé sur l'ontologie de domaine, mais sur une ontologie locale autonome formellement articulée avec une ou plusieurs ontologies de domaine par des relations de subsomption ;
- au niveau logique, chaque classe de l'ontologie locale est associée à une vue qui permet l'accès au niveau ontologique, indépendamment de la représentation physique des données ;
- il est également possible (Dehainsala 2007), et c'est la démarche recommandée par la méthode SISRO, de représenter explicitement dans la base de données :
 - l'ontologie locale,
 - son articulation par subsomption avec l'ontologie partagée.

Il est alors possible, automatiquement, à la fois de fournir un accès aux données au niveau ontologique, et d'intégrer le contenu de la base de données avec d'autres sources fondées sur la même ontologie partagée.

Notre méthodologie comporte quatre étapes : (1) définition de l'ontologie locale ; (2) conception du modèle conceptuel ; (3) définition des modèles logique et physique ; (4) fourniture d'accès au niveau ontologique. La figure 1 ci-dessous, résume cette approche.

4. le terme propriété regroupe ici aussi les notions d'attributs et de relations

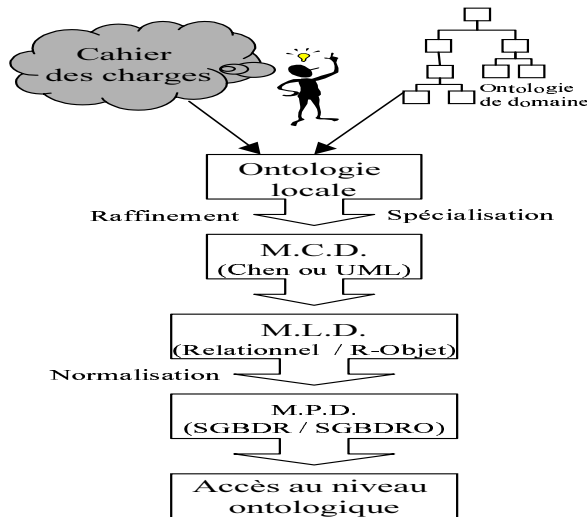


Figure 1. La méthode SISRO

4.1. Définition de l'ontologie locale

En se basant sur son cahier des charges, et éventuellement sur les interviews des utilisateurs finaux, le concepteur va identifier dans l'ontologie de domaine l'ensemble des concepts et propriétés qu'il souhaite intégrer dans son dictionnaire de données.

4.1.1. Définition des classes locales par subsomption

La relation de subsomption va permettre au concepteur de redéfinir entièrement la structure des classes en fonction de ses besoins. Afin d'assurer l'autonomie tant structurelle (super-classes, propriétés) que temporelle (évolution éventuelle de l'ontologie de domaine) de l'ontologie locale, celle-ci ne doit pas contenir directement des classes de l'ontologie partagée, mais seulement des classes spécifiques reliées à celles de l'ontologie partagée par des relations de subsomption. En exploitant cette relation, le concepteur va importer sélectivement dans le contexte de chaque classe les propriétés applicables de la classe subsumante qui s'avèrent pertinentes dans le contexte du système en cours de conception. Ceci permet de construire ainsi une ontologie locale (dictionnaire des données) conforme au cahier des charges et qui s'articule avec l'ontologie partagée. Ainsi, l'ontologie locale possède sa propre structure de généralisation / spécialisation qui n'est pas imposée par l'ontologie de domaine, bien qu'elle en respecte les relations de spécialisation. A ce stade, l'ontologie apparaît comme une spécialisation de l'ontologie partagée, avec importation partielle de ses propriétés. Dans l'exemple de la figure 2, le concept canonique *Personne*, et le concept non canonique *Mineur* (défini par restriction sur la propriété *age*), sont spécialisés en *Personne_Loc* et *Mineur_Loc* par la relation de subsomption. Seuls les attributs *nom*,

adresse, *date_nais* et *age* sont ensuite explicitement importés de l'ontologie de domaine.

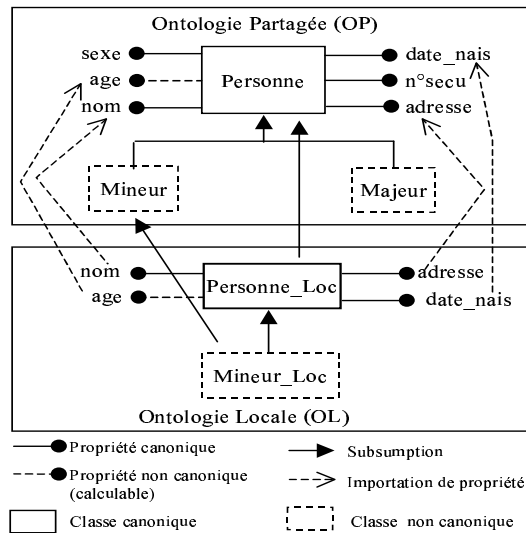


Figure 2. Extraction de l'ontologie locale via une relation de subsumption

4.1.2. Raffinement de l'ontologie locale

Le sous-ensemble de concepts (classes et propriétés) importé de l'ontologie de domaine par la relation de subsumption ne recouvre pas toujours intégralement le cahier des charges. Ce sous-ensemble nécessite en général d'être enrichi et spécialisé pour répondre aux objectifs applicatifs spécifiques. Ainsi, le concepteur peut si l'ontologie partagée ne recouvre pas de façon suffisamment détaillée le domaine de son application, étendre l'ontologie locale par ajout de nouvelles classes et/ou de nouvelles propriétés. Les concepts et propriétés pourront aussi être renommés pour mieux refléter le contexte applicatif. Notons que ce renommage est sans conséquence car chaque concept de chaque ontologie est associé à un unique identifiant et que les relations formelles sont des relations entre identifiants. Dans la figure 3, on peut voir la restructuration de l'ontologie locale où deux nouvelles classes (*Client_En_Compte* et *Client_Particulier*) ont été introduites dans la hiérarchie, et la classe *Personne_Loc* renommée en *Client*. L'attribut *n°secu* de la classe *Client_Particulier* est importé de l'ontologie de domaine par subsumption, les attributs *date_nais* et *age* sont descendus dans la classe *Client_Particulier* (on considère que, pour un client en compte, l'ensemble de ces informations figure dans une autre base de données).

Il est impératif qu'à la suite du raffinement / spécialisation, l'ontologie locale garde son articulation par rapport à l'ontologie de référence ; pour cela, elle doit respecter le *principe d'engagement sur une ontologie de référence* définie par les contraintes suivantes :

- Toute classe locale doit référencer, par la relation de subsomption, la plus petite classe subsumante existante dans la hiérarchie de référence si ce n'est pas la même que celle de sa propre super classe ;
- Toute propriété nécessaire à l'ontologie locale et existante dans l'ontologie de référence doit être importée en l'état au travers de la relation de subsomption.

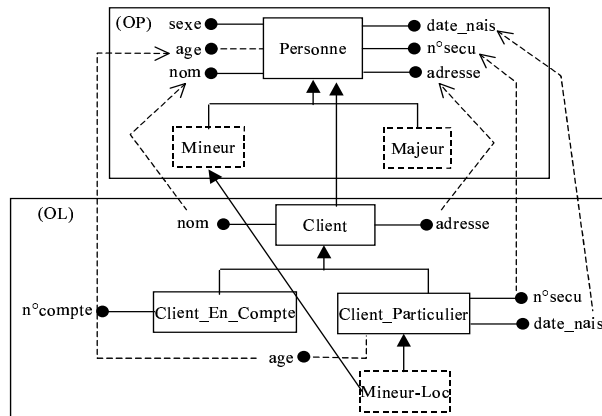


Figure 3. Raffinement / Spécialisation de l'ontologie locale

Notons que plusieurs ontologies de référence peuvent être utilisées dans la même ontologie locale.

4.2. Construction du modèle conceptuel

Une fois l'ontologie locale conçue, le concepteur peut alors définir son modèle conceptuel à partir du sous-ensemble canonique de celle-ci. L'adaptation aux besoins applicatifs ayant été pris en compte au niveau de l'ontologie locale, cette phase consiste essentiellement à choisir les classes et propriétés canoniques constituant le modèle conceptuel. Le MCD résultant est alors validé formellement par rapport au modèle de traitement. Il peut apparaître que certaines informations s'avèrent alors manquantes. Dans ce cas, l'ontologie locale est d'abord complétée, puis les éléments complémentaires nécessaires sont choisis pour faire partie de modèle conceptuel. Notons qu'il peut arriver que toutes les classes et propriétés de l'ontologie locale ne soient pas utilisées dans la construction du modèle conceptuel. Cela est lié au caractère descriptif de l'ontologie locale qui décrit le domaine couvert par la base mais peut également décrire le domaine couvert par d'autres bases de la même entreprise, et prévoir éventuellement leurs évolutions. Cela permet au concepteur d'intégrer dans sa modélisation des informations d'*urbanisation* qui permettront de garantir la stabilité et la flexibilité de la base de données en anticipant les besoins futurs de l'application (Wedemeijer 2001).

4.3. Représentation logique et physique des données

Selon le formalisme de modélisation choisie (UML, modèle de Chen), le MCD est traduit en un modèle logique normalisé selon les processus de normalisation usuels (Codd 1970, Fagin 1977). Ensuite, ceux des concepts non canoniques de l'ontologie locale que le concepteur souhaite rendre disponibles aux utilisateurs sont définis comme des vues sur le modèle logique précédent. Ainsi, la classe *Mineur_Loc* et l'attribut *age* seront seulement représentés dans des vues. Ce modèle logique est ensuite utilisé par le SGBD pour générer la représentation physique des données. Cette représentation n'intègre donc, pour respecter le principe du monde fermé, que les concepts primitifs utilisés dans la définition du MCD.

4.4. Fourniture d'un accès au niveau ontologique

Nous proposons deux méthodes différentes pour permettre l'accès complet au niveau ontologique, tant au niveau ontologie locale, qu'au niveau ontologie(s) partagée(s).

4.4.1. Accès par vues

En définissant explicitement des vues pour les concepts (canoniques ou non) de l'ontologie locale et, pour ceux de l'ontologie partagée faisant l'objet d'une relation de subsumption, les données stockées dans la base de données deviennent accessibles au niveau connaissance i.e., en termes ontologiques. La base de données peut donc être utilisée et exploitée dans un univers partagé par tout utilisateur ayant une connaissance de l'ontologie de ce domaine. Nous définissons la relation *Ext* qui à toute classe (canonique ou non) d'une ontologie associe une vue qui permet de déduire l'ensemble de ses instances comme suit :

- Pour un concept canonique : la vue correspondante sera la projection sur les propriétés applicables à ce concept de l'ensemble des instances appartenant à l'ensemble des classes subsumées par ce concept. Selon le résultat de l'application des règles de normalisation, et selon la représentation des relations d'héritage au sein de la base de données, les jointures adaptées seront évoquées. Par exemple, avec la représentation une table par classe concrete, les instances de la classe *Client* seront calculées par la vue : `SELECT nom, adresse FROM client_En_Compte UNION SELECT nom, adresse FROM client_Particulier.`

- Pour un concept non canonique : la vue correspondante sera exprimée par traduction de l'opérateur de non canonicité utilisé pour définir ce concept au niveau ontologique. Par exemple, elle pourra être une intersection ou une union sur la ou les relations correspondantes aux classes associées aux concepts canoniques dont elle dérive, ou encore une sélection associée à une projection suivie d'une jointure, etc. Les données des classes non canoniques seront donc stockées dans la base de données dans la ou les tables des classes canoniques dont elles dérivent. Ainsi, une vue sera implémentée pour calculer les instances de la classe *Mineur_Loc* à partir des instances

de la classe *Client_Particulier* : `SELECT * FROM Client_Particulier WHERE age < 18`. De même, les attributs non canoniques seront calculés à partir des attributs canoniques dont ils dérivent. Par exemple, l'attribut `age` sera dérivé de l'attribut `date_nais` par : `age = date_du_jour - date_nais`.

4.4.2. Accès par représentation explicite des ontologies

Il est également possible, et c'est la démarche que nous préconisons, de représenter explicitement dans la base de données l'ontologie locale et son articulation avec l'ontologie partagée. On peut alors exploiter les identifiants de concepts pour établir une relation bilatère entre toute donnée et le concept ontologique qui en définit le sens.

5. Mise en œuvre

L'ensemble de la méthode SISRO est associé à un ensemble d'outils qui permettent d'en mettre en œuvre les différentes étapes. Les ontologies actuellement utilisées sont des ontologies PLIB, normalisées à l'ISO sous le nom ISO13584 Part LIBrary⁵ (ISO 13584-501 2007). C'est le modèle d'ontologie le plus utilisé dans le domaine technique. Les outils comportent, un éditeur PLIBEditor, librement disponible sur le site de PLIB. Celui-ci permet de réaliser toutes les phases de création de l'ontologie locale et d'importer ou exporter ontologies et données. Cet outil a en particulier été utilisé à grande échelle par une entreprise du domaine automobile pour définir l'ontologie de l'ensemble des composants de production utilisés par elle (voir figure 6). Ils comportent également un système de gestion de base de données à base ontologique, OntoDB⁶ (Pierra *et al.* 2005, Dehainsala 2007), qui permet la génération de la base de données et son exploitation par un langage de requêtes définies au niveau ontologique (Jean *et al.* 2007). Cette base de données est actuellement utilisée, à l'état de prototype, dans plusieurs projets de recherche. Notons que l'outil PLIBEditor existe à la fois en version autonome (gestion des données en mémoire centrale avec sauvegarde sur fichier), et en tant qu'éditeur de la base de données OntoDB (les données étant gérées de façon persistante dans la base de données). Les deux grandes phases de la méthode SISRO (conception de l'ontologie locale et génération de la base de données) peuvent donc être réalisées soit dans deux environnements différents, soit, dans l'environnement de la base de données. La figure 4, ci-dessous, montre les différentes étapes et l'outil qui permet leur mise en œuvre.

Nous détaillons brièvement ci-dessous la mise en œuvre des deux grandes parties de la méthode SISRO.

5. site de référence : <http://www.plib.ensma.fr/>

6. <http://www.plib.ensma.fr/plib/demos/ontodb/index.html>

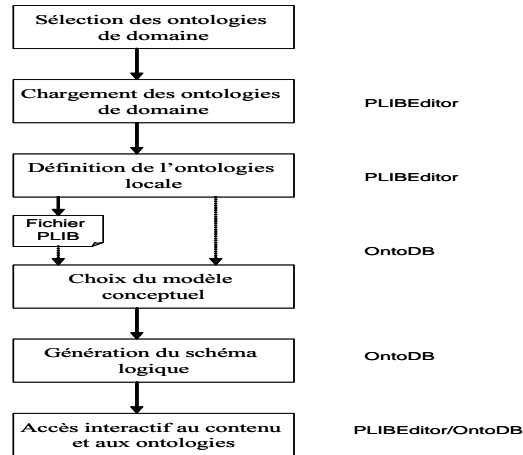


Figure 4. Mise en oeuvre de la méthode SISRO

5.1. Définition de l'ontologie locale

Les ontologies PLIB disposent de plusieurs syntaxes de représentation sous forme de fichiers : en particulier le format dit P21, conforme à la norme ISO 10303-21 :2002 (ISO 10303-21 2002), et le format OntoML (ISO/DIS 13584-32 2008), schéma XML en cours de normalisation. Le format le plus courant pour les ontologies disponibles étant le format P21, c'est celui-ci qui est utilisé. Les ontologies choisies sont donc d'abord lues sous forme P21 et chargées dans PLIBEditor afin d'être spécialisées.

Dans le modèle PLIB, l'opérateur de spécialisation d'une classe avec importation sélective de propriétés est représenté explicitement par un opérateur du langage appelé *OntoSub*. La caractéristique de cette relation est, d'une part, de préciser quelles sont les propriétés importées, et, d'autre part, d'être considérée comme une relation d'interfaçage entre ontologies différentes, et non comme une relation interne à une ontologie. La subsomption entre classes internes à la même ontologie avec l'héritage usuel est appelée *OOSub* (ou *is-a*). Notons qu'un mécanisme analogue peut être réalisé, en OWL en utilisant des espaces de noms différents. C'est cette notion de spécialisation avec importation sélective qui est utilisée dans PLIBEditor.

A partir des ontologies disponibles, l'ontologie locale est alors définie. Dans un premier temps les classes choisies sont spécialisées par la relation *OntoSub*. PLIBEditor crée alors une classe différente (identifiant différent) mais recopie, par défaut, tous les attributs⁷ de la classe subsumante. PLIBEditor propose alors la liste des propriétés applicables à cette classe (que celles-ci soient définies dans cette classe, qu'elles soient héritées de ses super classes par *OOSub*, ou qu'elles soient elles-mêmes importées par *OntoSub*) pour permettre de sélectionner celles qui doivent être importées. L'identifiant original de ces propriétés est alors enregistré dans le contexte de la nou-

7. le mot attribut correspond à ce qui est parfois appelé méta-attribut : nom, définition, ...

velle classe, ces propriétés deviennent applicables à cette classe et seront héritées par ses sous-classes *OOSub*.

Dans un deuxième temps, des classes spécifiques sont créées et rattachées à l'ontologie locale, puis des propriétés spécifiques sont ajoutées si besoin.

Si cette conception a été faite indépendamment de la base de données, l'ontologie locale est alors exportée sous forme de fichier. Celui-ci contient pour chaque classe ayant été définie par spécialisation *OntoSub* et importation sélective :

- l'identifiant absolu de la classe subsumante,
- l'identifiant des différentes propriétés importées sélectivement.

En cas d'édition isolée de l'ontologie locale, les attributs des propriétés importées ne sont pas disponibles. Ils peuvent le devenir en chargeant à nouveau les ontologies référencées, le système prenant alors en charge automatiquement la résolution des références.

5.2. Génération de la base de données

Dans un premier temps, l'ontologie locale, et, si besoin est, les ontologies référencées, sont chargées dans OntoDB (Dehainsala 2007). Le système demande alors quelles classes, et, pour chaque classe, quelles propriétés sont exclues du champ du modèle conceptuel. La génération de schéma de la base de données est alors automatique en se basant sur les cardinalités des propriétés qui représentent des associations. L'éditeur PLIBEditor peut alors être utilisé pour éditer simultanément les ontologies et le contenu de la base de données en fournissant un accès au niveau ontologique aux données de la base. La figure 5 présente un exemple de résultat de l'approche SISRO pour créer une ontologie locale à partir de deux ontologies normalisées.

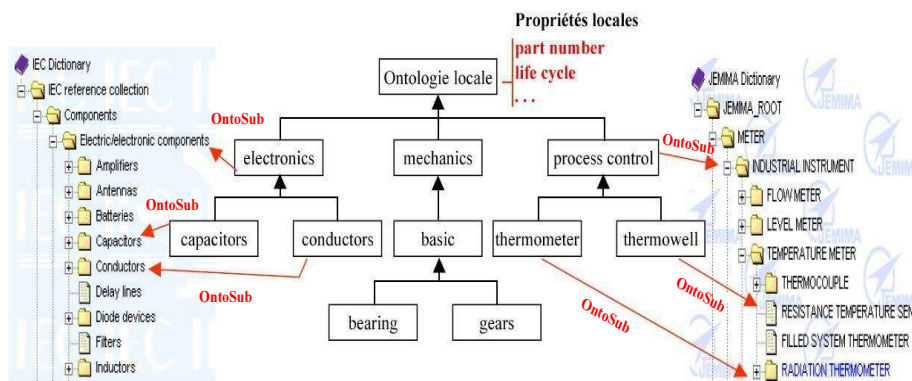


Figure 5. Référencement de plusieurs ontologies partagées

La figure 6 présente l'interface offerte par PLIBEditor pour accéder au contenu d'une base de données développée selon la méthode SISRO. Il s'agit de la base de données

de composants industriels développée par une grande entreprise du domaine automobile. Les données sont accessibles par classes, les différentes tables qui représentent la classe ayant éventuellement fait l'objet de jointure. La position de la classe dans l'ontologie est représentée dans la partie gauche. La partie droite comporte, dans l'onglet contenu, la table représentant la population de la classe, l'entête de chaque colonne étant le nom (dans le langage courant) de la propriété correspondante. Sélectionner l'entête d'une colonne (dans la figure 6 : " pouvoir de coupure") permet de faire apparaître dans la partie haute, la définition de la propriété. Si la propriété représente une association, la sélection d'une cellule correspondant à cette propriété fait apparaître, dans une nouvelle fenêtre, l'instance cible de l'association. L'interface ainsi offerte est générique puisqu'elle accède à l'ontologie représentée dans OntoDB.

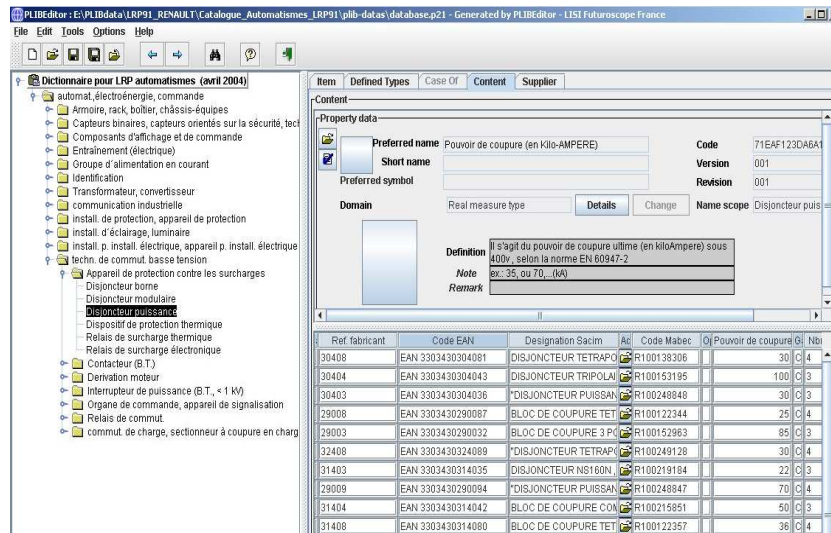


Figure 6. Accès au contenu d'une base de données au niveau ontologique avec PLIBEditor

6. Apport de la méthode SISRO

6.1. Simplifier la tâche du concepteur

En prenant comme point de départ le cahier des charges et l'ontologie de domaine, le temps et l'effort fournis par le concepteur afin d'aboutir à la maîtrise du domaine sont considérablement améliorés. La sémantique consensuelle du domaine fournie par l'ontologie chaque fois que celle-ci existe aide le concepteur dans la définition (choix des termes et relations) de son dictionnaire de données local.

6.2. Faciliter l'échange d'information

L'ontologie de domaine définit les nominations usuelles et les types des concepts qui caractérisent le domaine de l'application à développer. En spécialisant ces concepts et propriétés par la relation de subsomption, les MCDs définis par des concepteurs différents dans le même domaine d'application à partir d'un même cahier des charges et en respectant le principe d'engagement sur une même ontologie de domaine, ne présenteront plus de conflit pour tout ce qui est consensuel et figure déjà dans l'ontologie de domaine. Seules les spécificités de chaque application représentées par des extensions spécifiques de leurs ontologies locales, pourront, le cas échéant, nécessiter une intégration spécifique. Les MCDs pourront être différents, mais à tout le moins, les classes et propriétés communes référenceront les mêmes entrées de l'ontologie partagée ce qui évitera tous les problèmes d'hétérogénéité sémantique pour les mêmes informations.

6.3. Accès aux données au niveau ontologique

Les deux méthodes que nous avons proposées pour permettre un accès au niveau ontologique permettent toutes deux :

- d'interroger par requêtes la base en terme de l'ontologie partagée ; et donc
- d'intégrer automatiquement la base de données avec d'autres bases de données conçues selon la même approche et à partir de la même ontologie partagée soit selon une approche médiateur soit selon une approche entrepôt (XUAN 2006, Bellatreche *et al.* 2006).

Dans la représentation par vues, ceci est réalisé au niveau SQL. Lorsque les ontologies elles-mêmes sont représentées, la définition des vues n'est plus nécessaire. Nous avons en effet développé un langage de requête de niveau ontologique dont l'interpréteur est capable de calculer automatiquement le contenu d'une classe à partir des informations définies dans l'ontologie (Jean *et al.* 2006, Jean 2007). De plus, dans ce cas, l'éditeur PLIBEditor permet, par exploitation de l'ontologie, d'offrir directement un accès interactif du contenu de la base de données au niveau ontologique.

6.4. Autonomie dans la définition du MCD

Dans beaucoup de secteurs, et tout particulièrement dans le domaine de l'ingénierie dans lequel de nombreuses ontologies normalisées commencent à apparaître (Pierra 2008), les besoins d'une application particulière se sont pas couverts par une seule ontologie mais par plusieurs fragments (souvent de petite taille) de plusieurs ontologies différentes. Dans ce secteur, la complexité des ontologies normalisées les rendrait inutilisables si elles devaient être importées globalement dans le contexte d'une base de données. L'approche SISRO permet, par contre, de n'introduire dans la base de

données que les concepts ontologiques pertinents, tout en rendant celle-ci susceptible de répondre à toutes les requêtes correspondant à son contenu.

La figure 5, donne l'exemple d'une base de données d'ingénierie qui s'appuie sur deux ontologies normalisées (ISO 13584-501 2007, IEC 61360-4 1998) comportant plusieurs centaines de classes, mais qui n'en spécialise qu'un tout petit nombre.

6.5. Autonomie dans la gestion des évolutions de la base de données

Dans le contexte actuel, les ontologies partagées sont loin de recouvrir l'ensemble des domaines de connaissances, de nouvelles ontologies sont en développement dans de nombreuses communautés et de nouvelles ontologies peuvent prendre le dessus sur d'autres existant antérieurement, il est donc nécessaire que chaque organisation puisse décider de l'évolution de ses propres données. Une caractéristique essentielle de la méthode SISRO, et de la distinction ontologie partagée/ontologie locale, est précisément de laisser à l'administrateur de chaque base de données la maîtrise de l'évolution de sa base, indépendamment des évolutions éventuelles de la ou des ontologie(s) partagée(s) référencée(s). En effet, le MCD étant fondé sur l'ontologie locale, celle-ci peut évoluer indépendamment des ontologies partagées. Seules les correspondances entre ontologies partagées et ontologie locale, ou, dans le cas de l'accès par vues, seules les définitions de vues correspondantes aux classes de l'ontologie globale sont donc à modifier lorsque l'administrateur souhaite faire évoluer le MCD, ou lorsque, au contraire, l'ontologie partagée est modifiée et que l'administrateur ne souhaite pas modifier les données locales.

7. Origine de la méthode SISRO et perspective

Au cours des vingt dernières années, avec le développement de l'ingénierie numérique, est apparu le besoin de représenter sous forme informatique les composants industriels utilisés dans la production des produits. Ce besoin est apparu tant dans les entreprises utilisatrices, c'est la notion de "base de données composant" ou "base de données d'ingénierie", que dans les entreprises productrices de composants, c'est la notion de "catalogue informatisé". Une telle représentation nécessitant à la fois de modéliser la connaissance sur les composants (classes, propriété, comportement) et de modéliser leur caractérisation par des grandeurs caractéristiques, un modèle d'ontologie, appelé PLIB, a été développé dans ce but et normalisé simultanément à l'organisation internationale de normalisation (ISO) et à la commission électrotechnique internationale (IEC). Des ontologies conformes au modèle PLIB ont alors été développées pour différents domaines de l'ingénierie (par exemple : fixations mécaniques, composants électroniques, instruments de mesure, composants optiques, outils d'usinages, etc.), soit sous forme de normes internationales (par exemple les normes ISO 13399, ISO 13584-501, IEC 61360-4), soit sous forme de standards industriels développés par des consortia (par exemple : JENA au Japon, PROLIST en Allemagne, etc.). Le développement de telles ontologies partagées est toujours extrêmement long,

typiquement d'une durée de l'ordre de trois ans pour une ontologie de quelques centaines de classes. Ces ontologies sont donc, compte tenu de l'évolution permanente des produits, toujours incomplètes ou en retard sur les besoins de chaque organisation. Le problème s'est donc posé de savoir comment exploiter ces ontologies partielles pour permettre, d'une part, la création de catalogues informatisés pour les entreprises produisant des composants et, d'autre part, des bases de données d'ingénierie pour les entreprises utilisant des composants, de façon à permettre l'intégration relativement aisée des différents catalogues utilisés par une même entreprise. C'est dans ce contexte que nous avons développé la méthode SISRO. Celle-ci visait à la fois : (1) à tirer profit de la connaissance formalisée dans les ontologies partagées pertinentes pour le domaine d'activité de l'entreprise, (2) à faciliter l'intégration pour toutes les données correspondant à des concepts partagés, (3) à permettre à chaque entreprise de décrire, aussi finement qu'elle le souhaite, les spécificités de ses propres produits, et (4) à déconnecter l'évolution des bases de données locales, qui devraient rester sous le contrôle de l'entreprise, et l'évolution des ontologies partagées qui, étaient contrôlées par des consortia beaucoup plus importants. Cette méthode a, en particulier, été utilisée au cours des dernières années par une grande entreprise du secteur automobile pour fabriquer sa base de données pour les composants destinés à être utilisés dans ses lignes de production. Ce projet est actuellement en cours de déploiement. Notons que le développement actuel du Web sémantique, et de l'ingénierie ontologique dans des domaines très variés, devraient favoriser, comme cela est le cas en ingénierie, l'émergence d'ontologies partagées pour différents domaines et constituer ainsi un cadre favorable pour l'utilisation de la méthode SISRO.

Concernant les perspectives, nous travaillons actuellement dans trois directions. D'abord, et du point de vue méthodologique, nous utilisons actuellement comme relation d'articulation entre ontologies partagées et ontologie locale, la subsomption de classes primitives et l'importation de propriétés. Ontologie locale et partagée(s) devant choisir les mêmes classes comme classes primitives, ceci limite l'autonomie du concepteur à la fois sur le choix des classes et propriétés instanciables, et sur la granularité de sa base de données. Nous étudions actuellement l'extension de ces relations pour donner plus de flexibilité à la méthode. Ainsi, par exemple, des relations algébriques entre concepts de l'ontologie (opérateurs de classes de la logique de description, fonction de dérivation pour les valeurs de propriétés) pourront être exploitées lors de la définition du MCD. Ceci étendra les possibilités de dérivations du MCD à partir d'ontologies existantes. Cela facilitera également l'intégration de MCDs existants. Ensuite, au niveau support de notre méthode, l'ensemble de nos outils (PLIBEditor et OntoDB) est basé sur le modèle d'ontologie PLIB. Nous développons actuellement une nouvelle version de nos outils basée sur un modèle d'ontologie noyau (Jean *et al.* 2007) associé à des mécanismes d'extension. Ces outils devraient alors supporter la mise en œuvre de la méthode, ainsi que l'importation et l'exportation d'ontologie et des données, avec des ontologies basées sur des modèles différents, et, en particulier, RDFS (Brickley *et al.* 2002) et OWL (Antoniou *et al.* 2003). Enfin, au niveau implémentation, la méthodologie que nous proposons n'est pas liée à nos outils, même si ceux-ci en facilitent la mise en œuvre. Il s'agit d'une approche générale qui trouve

toute sa pertinence, du point de vue intégration, lorsque le système cible permet à la fois de représenter les ontologies, et des données sous une forme traditionnelle. Nous étudions donc la faisabilité d'utiliser Oracle, doté de son gestionnaire RDF/RDFS, comme une autre infrastructure d'implantation du résultat de la méthode SISRO.

8. Conclusion

Dans ce travail, nous avons proposé une nouvelle approche de conception de bases de données fondée sur les ontologies de domaine. Cette approche, appelée SISRO pour Spécialisation Importation Spécifique et Représentation des Ontologies, utilise une ontologie locale pour assurer une meilleure autonomie de la base de données par rapport à l'ontologie de domaine. Elle représente ontologie locale et ontologie(s) partagée(s) soit par un mécanisme de vues, soit par une représentation explicite des ontologies au sein de la base de données. Les classes de l'ontologie locale sont différentes des classes des ontologies partagées avec lesquelles elles sont articulées par des relations de subsomption. Elles importent à travers cette relation celles des propriétés applicables à leur classe subsumante qui s'avèrent pertinentes pour le cahier des charges à satisfaire. Notre méthodologie comporte quatre étapes principales. Dans l'étape 1, une ontologie locale est construite par identification des concepts et propriétés des ontologies partagées pertinents par rapport au cahier des charges. Le concepteur dispose d'une grande liberté dans la structuration des classes (qui représentent les concepts retenus) et des propriétés. Cette ontologie locale peut alors être spécialisée par les concepts nécessaires qui ne figuraient pas dans les ontologies partagées, puis étendue par ajout de nouvelles propriétés. Dans la seconde étape, le modèle conceptuel de l'application est défini à partir de l'ontologie locale. Le concepteur peut ne choisir qu'un sous ensemble de l'ontologie locale pour prévoir par exemple les extensions futures. Cette différence entre ontologie locale et MCD est justifiée par le caractère descriptif des ontologies et la nature prescriptive des modèles conceptuels. Dans la troisième étape, le modèle conceptuel est traduit en modèle logique, puis en modèle physique à partir d'un ensemble de règles de transformation suivant le formalisme utilisé. Enfin, la quatrième étape fournit un accès aux données au niveau ontologique afin de permettre l'intégration automatique. Dans cette étape, le concepteur a le choix entre deux possibilités : (1) utiliser des vues pour représenter les classes des ontologies concernées ; (2) représenter dans la même base de données à la fois les données, les concepts ontologiques qui en définissent le sens (ontologie locale, ontologie partagée), mais aussi le lien entre ontologie et données au travers des identifiants de concepts. Dans ce dernier cas, une interface graphique interactive et générique peut être développée pour permettre à un utilisateur de parcourir les concepts de la base au niveau connaissance, et un langage de requête fournissant l'accès aux données par le niveau ontologique peut être rendu disponible. La méthode SISRO est entièrement outillée par une suite d'outils qui permettent d'en réaliser toutes les étapes. Ces outils sont déjà mis en œuvre dans diverses applications d'ingénierie.

Notons que la méthode SISRO est particulièrement indiquée lorsqu'il existe dans le domaine à modéliser des ontologies existantes sur lesquelles le concepteur peut, au moins partiellement, baser son travail. Ceci est déjà le cas dans un certain nombre de domaines tels que l'ingénierie et le bio-médical. Les développements actuels dans le contexte du Web sémantique devraient contribuer à multiplier le nombre de domaines dans lesquels de telles ontologies consensuelles existent. En l'absence d'ontologie partagée préexistante, la mise en œuvre de la méthode SISRO nécessite en amont un effort supplémentaire. Cet effort nous paraît intéressant dans deux cas. D'une part, lorsque l'on souhaite préparer la base de données pour une intégration ultérieure. D'autre part, lorsque l'on dispose d'un SBGD à base ontologique qui comme notre système OntoDB, fournit automatiquement l'accès aux données de la base à partir de l'ontologie. L'effort supplémentaire de création de l'ontologie locale est alors compensé dans ce cas, d'une part par la génération automatique d'une API d'accès aux données au niveau ontologique, et, d'autre part, par l'existence d'une interface générique d'accès graphique interactif aux données à partir de l'ontologie qui évite l'écriture d'un applicatif spécifique.

9. Bibliographie

- Alexaki S., Christophides V., Karvounarakis G., Plexousakis D., Tolle K., « The ICS-FORTH RDFSuite : Managing Voluminous RDF Description Bases », *Proceedings of the 2nd International Workshop on the Semantic Web*, p. 1-13, 2001.
- Antoniou G., van Harmelen F., « Web ontology language : Owl », 2003.
- Barrasa J., Óscar Corcho, Gómez-Pérez A., « R2O, an Extensible and Semantically Based Database-to-ontology Mapping Language », *Second Workshop on Semantic Web and Databases (SWDB'2004)*, August, 2004.
- Bechhofer S., van Harmelen F., Hendler J., Horrocks I., McGuinness D., Patel-Schneider P., Stein L., « OWL Web Ontology Language Reference », *W3C*, <http://www.w3.org/TR/owl-ref/>, 2004.
- Bellatreche L., Dung N. X., Pierra G., dehainsala H., « Contribution of Ontology-based Data Modeling to Automatic Integration of Electronic Catalogues within Engineering Databases », *Computers in Industry*, vol. 57, p. 711-724, 2006.
- B.McBride, « Jena : Implementing the RDF model and syntax specification », *Proceedings of the 2nd International Workshop on the Semantic Web*, 2001.
- Bozsak E., Ehrig M., Handschuh S., Hotho A., Maedche A., Motik B., Oberle D., Schmitz C., Staab S., Stojanovic L., Stojanovic N., Studer R., Stumme G., Sure Y., Tane J., Volz R., Zacharias V., « KAON - Towards a Large Scale Semantic Web », *Proceedings of the 3rd International Conference on E-Commerce and Web Technologies (EC-WEB'02)*, Springer-Verlag, London, UK, p. 304-313, 2002.
- Brickley D., Guha R., « RDF Vocabulary Description Language 1.0 : RDF Schema », *W3C*, <http://www.w3.org/TR/rdf-schema/>, 2002.
- Broekstra J., Kampman A., van Harmelen F., « Sesame : A Generic Architecture for Storing and Querying RDF and RDF Schema », in I. Horrocks, J. Hendler (eds), *Proceedings of the 1st*

- International Semantic Web Conference (ISWC'02)*, n° 2342 in *Lecture Notes in Computer Science*, Springer Verlag, p. 54-68, July, 2002.
- Chawathe S. S., Garcia-Molina H., Hammer J., Ireland K., Papakonstantinou Y., Ullman J. D., Widom J., « The TSIMMIS Project : Integration of Heterogeneous Information Sources », *Proceedings of the 10th Meeting of the Information Processing Society of Japan*, vol. 26, p. 7-18, Marsh, 1994.
- Chen P., « The Entity-Relationship Model : Toward a Unified View of Data », *ACM Transactions on Database Systems (TODS)*, vol. 1, ACM, New York, NY, USA, p. 9-36, 1976.
- Chen Y., Ou J., Jiang Y., Meng X., « HStar - A Semantic Repository for Large Scale OWL Documents », *First Asian Semantic Web Conference (ASWC'06)*, vol. 4185, p. 415-428, 2006.
- Chong E. I., Das S., Eadon G., Srinivasan J., « An efficient SQL-based RDF querying scheme », *VLDB '05 : Proceedings of the 31st international conference on Very large data bases*, VLDB Endowment, p. 1216-1227, 2005.
- Codd E. F., « A relational model of data for large shared data banks », *Communications of the ACM*, vol. 13, n° 6, p. 377-387, 1970.
- Connolly D., van Harmelen F., Horrocks I., McGuinness D. L., Patel-Schneider P. F., Stein L. A., « DAML+OIL Reference Description », *W3C*, <http://www.w3.org/TR/daml+oil-reference>, March, 2001.
- Dehainsala H., « Explication de la sémantique dans les bases de données : base de données à base ontologique et le modèle OntoDB », *Thèse de Doctorat Université de Poitiers - <http://tel.archives-ouvertes.fr/docs/00/15/75/95/PDF/These.pdf>*, 2007.
- Dehainsala H., Pierra G., Bellatreche L., « OntoDB : An Ontology-Based Database for Data Intensive Applications », *Proceedings of the 12th International Conference on Database Systems for Advanced Applications (DASFAA'07)*, Lecture Notes in Computer Science, Springer, p. 497-508, 2007.
- Fagin R., « Multivalued dependencies and a new normal form for relational databases », *ACM Transactions on Database Systems (TODS)*, vol. 2, n° 3, p. 262-278, 1977.
- Goh C., Bressan S., Madnick E., Siegel M. D., « Context Interchange : New Features and Formalisms for the Intelligent Integration of Information », *ACM Transactions on Information Systems (TODS)*, vol. 17, n° 3, p. 270-293, 1999.
- Gruber T. R., *Formal ontology in conceptual analysis and knowledge representation. Chapter : Towards principles for the design of ontologies used for knowledge sharing.*, Kluwer Academic Publishers., 1993.
- Guarino N., Poli R., « Formal ontology in conceptual analysis and knowledge representation », *Special issue of the International Journal of Human and Computer Studies*, vol. 43, n° 5/6, p. 625-640, 1995.
- Haendel M. A., Neuhaus F., Osum-Sutherland D., Mabee P. M., Mungalland J. L. M. N. C. J., Smith B., « CARO - The Common Anatomy Reference Ontology », *Anatomy Ontologies for Bioinformatics Principles and Practice*, vol. 6, p. 327-349, 2007.
- Harth A., Decker S., « Optimized Index Structures for Querying RDF from the Web », *LA-WEB '05 : Proceedings of the Third Latin American Web Congress*, IEEE Computer Society, Washington, DC, USA, p. 71-80, 2005.
- IEC 61360-4, « Standard data element types with associated classification scheme for electric components – Part 4 : IEC reference collection of standard data element types », 1998.

- ISO 10303-21, « Systèmes d'automatisation industrielle et intégration – Représentation et échange de données de produits – Partie 21 : Méthodes de mise en application : Encodage en texte clair des fichiers d'échange », *International Organization for Standardization*, 2002.
- ISO 13584-501, « Systèmes d'automatisation industrielle et intégration – Bibliothèque de composants – Partie 501 : Dictionnaire de référence pour les instruments de mesure », *International Organization for Standardization*, 2007.
- ISO/DIS 13584-32, « Systèmes d'automatisation industrielle et intégration – Bibliothèque de composants – Partie 32 : Ressources d'implémentation : OntoML : Langage de marquage ontologique », *International Organization for Standardization / Draft International Standard*, 2008.
- Jean S., « OntoQL : un langage d'exploitation des bases de données à base ontologique », *Thèse de Doctorat Université de Poitiers - <http://www.lisi.ensma.fr/ftp/pub/documents/thesis/2007-thesis-jean.pdf>*, 2007.
- Jean S., Aït-Ameur Y., Pierra G., « Querying Ontology Based Database Using OntoQL (an Ontology Query Language) », *Proceedings of On the Move to Meaningful Internet Systems : CoopIS, DOA, GADA, and ODBASE, OTM Confederated International Conferences (ODBASE)*, Springer, p. 704-721, 2006.
- Jean S., Pierra G., Aït-Ameur Y., *Domain Ontologies : a Database-Oriented Analysis*, vol. 1 of *Lecture Notes in Business Information Processing*, Springer Berlin Heidelberg, p. 238-254, 2007.
- Kim W., Seo J., « Classifying Schematic and Data Heterogeneity in Multidatabase Systems », *IEEE Computer Society Press*, vol. 24, n° 12, p. 12-18, 1991.
- Kiryakov A., Ognyanov D., Manov D., « OWLIM : A Pragmatic Semantic Repository for OWL », p. 182-192, 2005.
- Lu J., Ma L., Zhang L., Brunner J.-S., Wang C., Pan Y., Yu Y., « SOR : a practical system for ontology storage, reasoning and search », *VLDB '07 : Proceedings of the 33rd international conference on Very large data bases*, VLDB Endowment, p. 1402-1405, 2007.
- Ma L., Wang C., Lu J., Cao F., Pan Y., Yu Y., « Effective and efficient semantic web data management over DB2 », *SIGMOD '08 : Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, ACM, New York, NY, USA, p. 1183-1194, 2008.
- Pan Z., Heflin J., « DLDB : Extending Relational Databases to Support Semantic Web Queries », *Proceedings of the 1st International Workshop on Practical and Scalable Semantic Systems (PSSS'03)*, p. 109-113, 2003.
- Park M. J., Lee J. H., Lee C. H., Lin J., Serres O., Chung C. W., « An Efficient and Scalable Management of Ontology », *Proceedings of the 12th International Conference on Database Systems for Advanced Applications (DASFAA'07)*, vol. 4443 of *Lecture Notes in Computer Science*, Springer, p. 975-980, 2007.
- Pierra G., « Context Representation in Domain Ontologies and its Use for Semantic Integration of Data », *Journal Of Data Semantics (JODS)*, vol. 4900, p. 173-210, 2008.
- Pierra G., Dehainsala H., Ait-Ameur Y., Bellatreche L., Chochon J., Mimoune M. E.-H., « Bases de données à base ontologique. Principe et mise en oeuvre », *Ingénierie des Systèmes d'Information (ISI'05)*, vol. 10, n° 2, p. 91-115, 2005.
- Roldan-Garcia, Navas-Delgado M., Aldana-Montes, Aldana-Montes J., « A design methodology for semantic Web database-based systems », *Third International Conference on Information Technology and Applications (ICITA'05)*, vol. 1, p. 233-237, 2005.

- Sugumaran V., Storey V. C., « The role of domain ontologies in database design : An ontology management and conceptual modeling environment », *ACM Trans. Database Syst.*, vol. 31, ACM Press, New York, NY, USA, p. 1064-1094, 2006.
- Wache H., Vögele T., Visser U., Stuckenschmidt H., Schuster G., Neumann H., Hübner S., « Ontology-based Integration of Information - A Survey of Existing Approaches », *Proceedings of the International Workshop on Ontologies and Information Sharing*, vol. , p. 108-117, August, 2001.
- Wedemeijer L., « Engineering for Conceptual Schema Flexibility », *RIDE '01 : Proceedings of the 11th International Workshop on research Issues in Data Engineering*, IEEE Computer Society, p. 85-92, 2001.
- Wiederhold G., « Mediators in the architecture of future information systems », *IEEE Computer*, vol. 25, n° 3, p. 38-49, March, 1992.
- XUAN D. N., « Intégration de base de données hétérogènes par articulation a priori d'ontologies : application aux catalogues de composants industriels. », *Thèse de Doctorat Université de Poitiers*, 2006.

Chimène Fankam termine son doctorat en informatique à l'ENSMA et à l'université de Poitiers. Ses travaux portent sur la conception d'une architecture de base de données à base ontologique (BDBO). Elle s'intéresse particulièrement l'intégration de mécanismes issus de différents modèles d'ontologies au sein d'une même BDBO par extension d'un modèle d'ontologie noyau.

Ladjet Bellatreche est maître de conférences en informatique à l'université de Poitiers. Ses travaux de recherche portent essentiellement sur la conception des bases de données à base ontologique, l'intégration des sources de données hétérogènes dirigée par des ontologies conceptuelles et l'optimisation physique des bases de données volumineuses.

Hondjack Dehainsala est Ingénieur en Recherche et Développement chez Mondeca, une entreprise spécialisée dans la gestion de thésaurus et de référentielle sémantique. Ses travaux de recherche portent sur le passage à l'échelle et le contrôle d'accès dans les bases de données à base ontologique, l'intégration de sources de données hétérogènes et le versionnement d'ontologies.

Yamine Ait Ameer est professeur d'informatique à l'ENSMA-Poitiers. Ses travaux portent sur la modélisation, la validation et la vérification formelles de données et de systèmes hétérogènes. En particulier, il aborde le problème de la réduction de l'hétérogénéité des modèles aux travers de l'explicitation de la sémantique pas la définition d'ontologies de domaine.

Guy Pierra est professeur d'informatique à l'ENSMA de Poitiers. Ses travaux actuels portent sur l'ingénierie des données, sur la conception d'ontologie de domaine et de bases de données à bases ontologiques, ainsi que sur la définition d'approches d'ingénierie dirigées par les modèles. Son équipe à déjà réalisée une série de normes ISO, connues sous le nom de PLIB, sur le domaine des ontologies appliquées aux commerces électroniques.

ANNEXE POUR LE SERVICE FABRICATION
A FOURNIR PAR LES AUTEURS AVEC UN EXEMPLAIRE PAPIER
DE LEUR ARTICLE ET LE COPYRIGHT SIGNÉ PAR COURRIER
LE FICHIER PDF CORRESPONDANT SERA ENVOYÉ PAR E-MAIL

1. ARTICLE POUR LA REVUE :
TSI Volume 28
2. AUTEURS :
Chimène Fankam — Ladjel Bellatreche* — Hondjack Dehainsala**
— Yamine Ait Aneur* — Guy Pierra**
3. TITRE DE L'ARTICLE :
*SISRO : Conception de bases de données à partir d'ontologies de do-
maine*
4. TITRE ABRÉGÉ POUR LE HAUT DE PAGE MOINS DE 40 SIGNES :
Conception de BD à partir d'ontologies
5. DATE DE CETTE VERSION :
6 janvier 2009
6. COORDONNÉES DES AUTEURS :
 - adresse postale :
 - * LISI/ ENSMA Université de Poitiers - 86961 Futuroscope Chasseneuil
Cedex
 - {fankamc, bellatreche, yamine, pierra}@ensma.fr
 - ** Mondeca (hondjack.dehainsala@mondeca.com)
 - téléphone : 05 49 49 80 63
 - télécopie : 05 49 49 80 64
 - e-mail : fankamc@ensma.fr
7. LOGICIEL UTILISÉ POUR LA PRÉPARATION DE CET ARTICLE :
L^AT_EX, avec le fichier de style `article-hermes.cls`,
version 1.2 du 03/03/2005.
8. FORMULAIRE DE COPYRIGHT :
Retourner le formulaire de copyright signé par les auteurs, téléchargé sur :
<http://www.revuesonline.com>

SERVICE ÉDITORIAL – HERMES-LAVOISIER
14 rue de Provigny, F-94236 Cachan cedex
Tél : 01-47-40-67-67
E-mail : revues@lavoisier.fr
Serveur web : <http://www.revuesonline.com>